

Class 11: AlphaFold

Michael McClellan (PID: A169692395)

Table of contents

Exploration of my sequence for the Find a Gene Project: 13

Here we read the results from AlphaFold and try to interpret all the models and quality score metrics:

```
library(bio3d)

pth <- "dimer_23119/"
pdb.files <- list.files(path = pth, full.names = TRUE, pattern = ".pdb")
```

Align and superpose all these models

```
file.exists(pdb.files)
```

```
[1] TRUE TRUE TRUE TRUE TRUE
```

```
pdb<= pdbaln(pdb.files, fit=TRUE, exefile="msa")
```

Reading PDB files:

```
dimer_23119//dimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_2_seed_000.pdb
dimer_23119//dimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb
dimer_23119//dimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb
dimer_23119//dimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_1_seed_000.pdb
dimer_23119//dimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb
.....
```

Extracting sequences

```
pdb/seq: 1    name: dimer_23119//dimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_2
pdb/seq: 2    name: dimer_23119//dimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_8
pdb/seq: 3    name: dimer_23119//dimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4
pdb/seq: 4    name: dimer_23119//dimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_
pdb/seq: 5    name: dimer_23119//dimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3
```

```
library(bio3dview)
view.pdbs(pdbs)
```

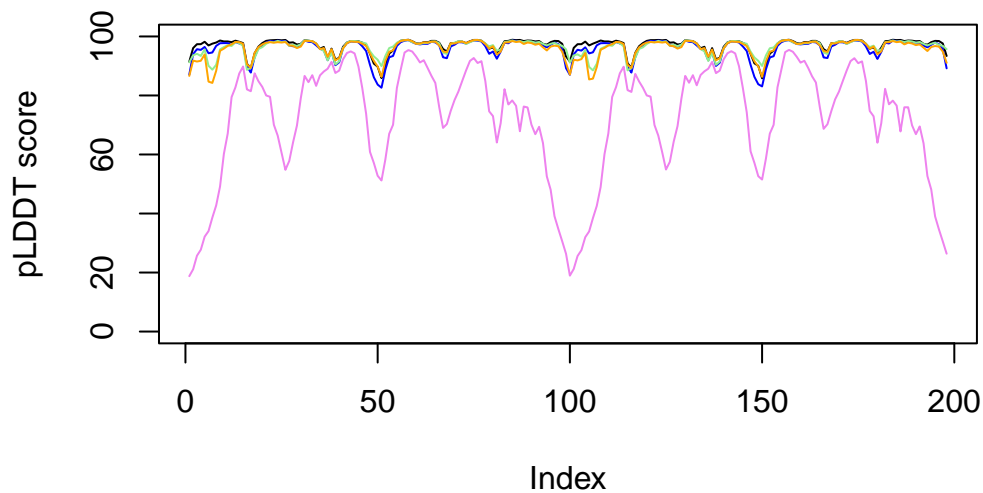
```
file:///private/var/folders/pr/k92jzqt11mq3td1wn21ttgvh0000gn/T/RtmpnrFUE2/file1e8a161552e9,
```



```

plot(pdbb$b[1,], typ="l", ylim=c(0,100), ylab = "pLDDT score")
lines(pdbb$b[2,], typ="l", col="blue")
lines(pdbb$b[3,], typ="l", col="lightgreen")
lines(pdbb$b[4,], typ="l", col="orange")
lines(pdbb$b[5,], typ="l", col="violet")

```



```

library(pheatmap)

rd <- rmsd(pdbb, fit=T)

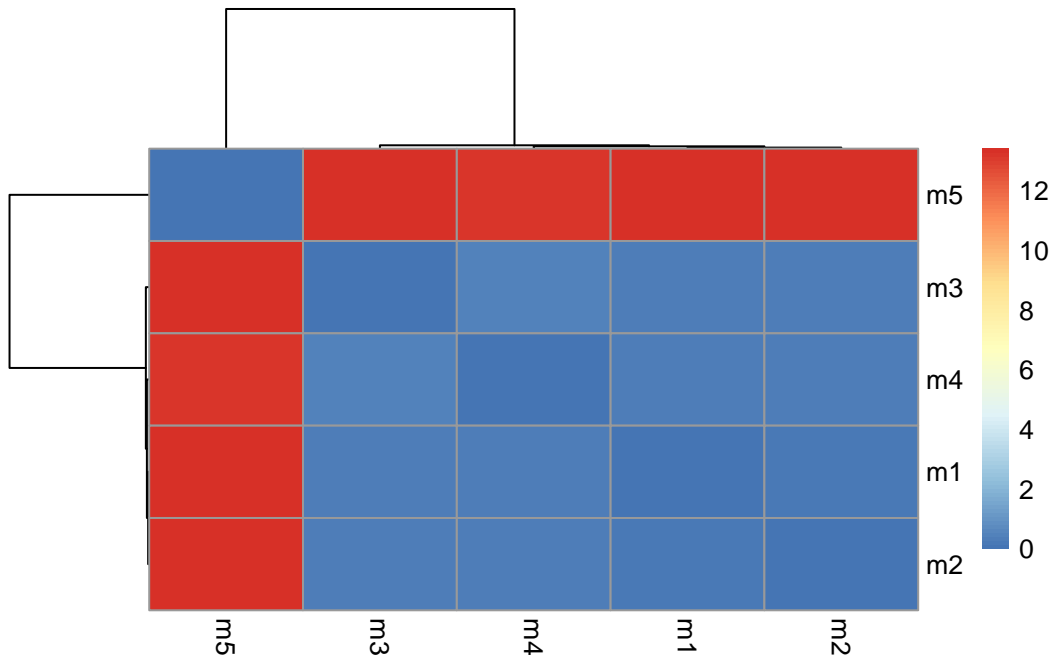
```

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

```

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

```



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

```
core size 197 of 198 vol = 32.323
core size 196 of 198 vol = 28.916
core size 195 of 198 vol = 27.276
core size 194 of 198 vol = 25.733
core size 193 of 198 vol = 24.724
core size 192 of 198 vol = 23.805
core size 191 of 198 vol = 23.128
core size 190 of 198 vol = 22.502
core size 189 of 198 vol = 21.867
core size 188 of 198 vol = 21.293
core size 187 of 198 vol = 20.774
core size 186 of 198 vol = 20.305
core size 185 of 198 vol = 19.783
core size 184 of 198 vol = 19.353
core size 183 of 198 vol = 18.94
core size 182 of 198 vol = 18.539
core size 181 of 198 vol = 18.097
core size 180 of 198 vol = 17.694
core size 179 of 198 vol = 17.257
core size 178 of 198 vol = 16.867
```

core size 177 of 198	vol = 16.519
core size 176 of 198	vol = 16.237
core size 175 of 198	vol = 15.978
core size 174 of 198	vol = 15.693
core size 173 of 198	vol = 15.412
core size 172 of 198	vol = 15.174
core size 171 of 198	vol = 14.957
core size 170 of 198	vol = 14.733
core size 169 of 198	vol = 14.532
core size 168 of 198	vol = 14.363
core size 167 of 198	vol = 14.222
core size 166 of 198	vol = 13.981
core size 165 of 198	vol = 13.885
core size 164 of 198	vol = 13.822
core size 163 of 198	vol = 13.736
core size 162 of 198	vol = 13.646
core size 161 of 198	vol = 13.58
core size 160 of 198	vol = 13.46
core size 159 of 198	vol = 13.261
core size 158 of 198	vol = 13.076
core size 157 of 198	vol = 12.91
core size 156 of 198	vol = 12.971
core size 155 of 198	vol = 12.926
core size 154 of 198	vol = 12.892
core size 153 of 198	vol = 12.769
core size 152 of 198	vol = 12.648
core size 151 of 198	vol = 12.53
core size 150 of 198	vol = 12.326
core size 149 of 198	vol = 12.104
core size 148 of 198	vol = 11.905
core size 147 of 198	vol = 11.473
core size 146 of 198	vol = 11.155
core size 145 of 198	vol = 10.956
core size 144 of 198	vol = 10.755
core size 143 of 198	vol = 10.546
core size 142 of 198	vol = 10.276
core size 141 of 198	vol = 10.066
core size 140 of 198	vol = 9.835
core size 139 of 198	vol = 9.619
core size 138 of 198	vol = 9.405
core size 137 of 198	vol = 9.142
core size 136 of 198	vol = 8.863
core size 135 of 198	vol = 8.526

core size 134 of 198	vol = 8.229
core size 133 of 198	vol = 7.998
core size 132 of 198	vol = 7.809
core size 131 of 198	vol = 7.509
core size 130 of 198	vol = 7.288
core size 129 of 198	vol = 7.084
core size 128 of 198	vol = 6.88
core size 127 of 198	vol = 6.59
core size 126 of 198	vol = 6.38
core size 125 of 198	vol = 6.197
core size 124 of 198	vol = 5.976
core size 123 of 198	vol = 5.764
core size 122 of 198	vol = 5.568
core size 121 of 198	vol = 5.312
core size 120 of 198	vol = 5.021
core size 119 of 198	vol = 4.758
core size 118 of 198	vol = 4.501
core size 117 of 198	vol = 4.218
core size 116 of 198	vol = 4.031
core size 115 of 198	vol = 3.801
core size 114 of 198	vol = 3.604
core size 113 of 198	vol = 3.379
core size 112 of 198	vol = 3.183
core size 111 of 198	vol = 3.002
core size 110 of 198	vol = 2.79
core size 109 of 198	vol = 2.603
core size 108 of 198	vol = 2.508
core size 107 of 198	vol = 2.421
core size 106 of 198	vol = 2.24
core size 105 of 198	vol = 2.084
core size 104 of 198	vol = 1.945
core size 103 of 198	vol = 1.832
core size 102 of 198	vol = 1.659
core size 101 of 198	vol = 1.582
core size 100 of 198	vol = 1.483
core size 99 of 198	vol = 1.382
core size 98 of 198	vol = 1.331
core size 97 of 198	vol = 1.264
core size 96 of 198	vol = 1.137
core size 95 of 198	vol = 1.043
core size 94 of 198	vol = 0.957
core size 93 of 198	vol = 0.885
core size 92 of 198	vol = 0.803

```
core size 91 of 198  vol = 0.73
core size 90 of 198  vol = 0.637
core size 89 of 198  vol = 0.56
core size 88 of 198  vol = 0.489
FINISHED: Min vol ( 0.5 ) reached
```

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

```
# 89 positions (cumulative volume <= 0.5 Angstrom^3)
```

	start	end	length
1	10	42	33
2	44	50	7
3	52	66	15
4	69	77	9
5	80	98	19

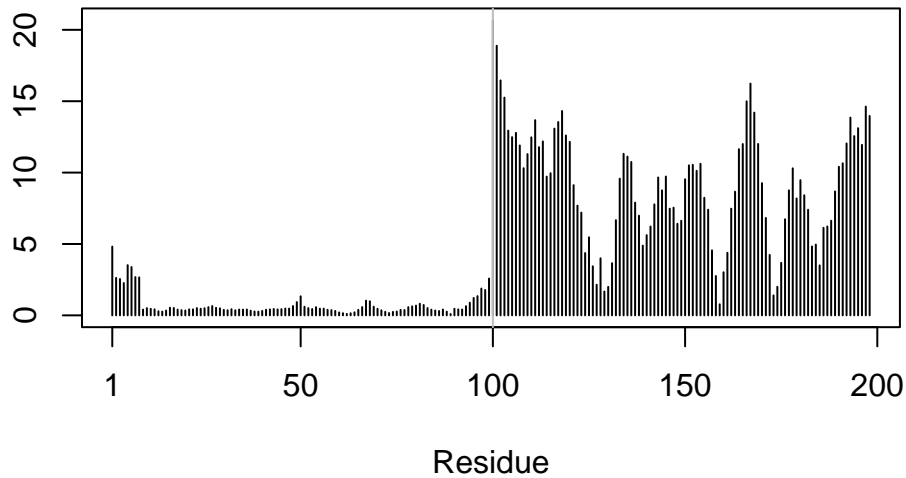
```
xyz <- pdbfit(pdb, core.inds, outpath="corefit_structures")
```

```
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")
```

```
library(jsonlite)

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

[1] "dimer_23119//dimer_23119_scores_rank_001_alphafold2_multimer_v3_model_2_seed_000.json"
[2] "dimer_23119//dimer_23119_scores_rank_002_alphafold2_multimer_v3_model_5_seed_000.json"
[3] "dimer_23119//dimer_23119_scores_rank_003_alphafold2_multimer_v3_model_4_seed_000.json"
[4] "dimer_23119//dimer_23119_scores_rank_004_alphafold2_multimer_v3_model_1_seed_000.json"
[5] "dimer_23119//dimer_23119_scores_rank_005_alphafold2_multimer_v3_model_3_seed_000.json"

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] 91.44 96.06 97.38 97.38 98.19 96.94
```

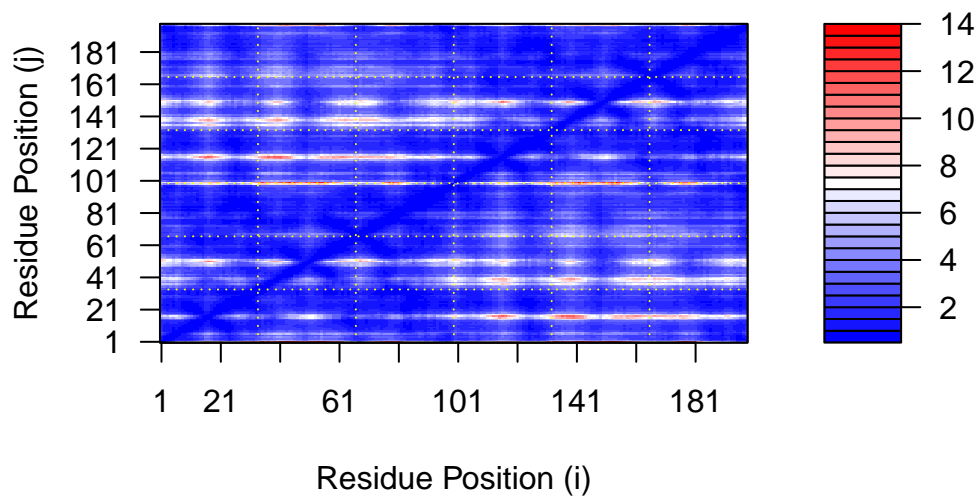
```
pae1$max_pae
```

```
[1] 13.57812
```

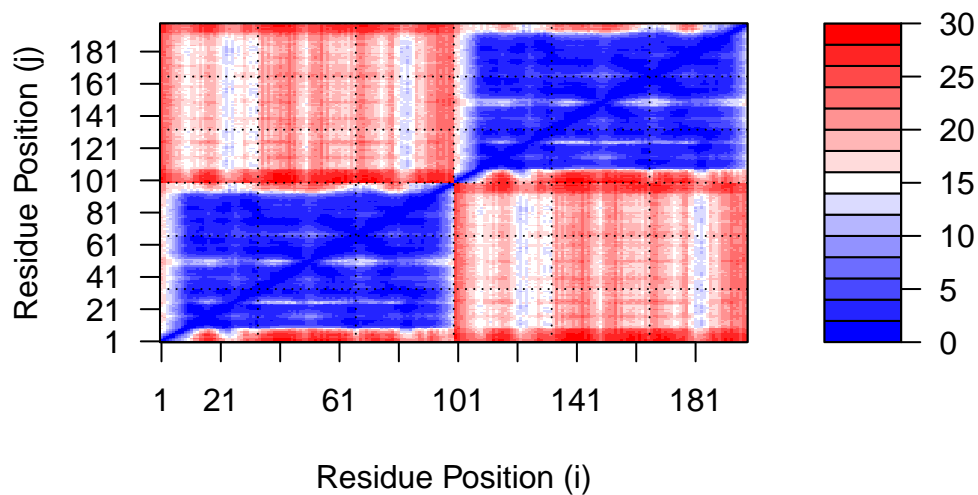
```
pae5$max_pae
```

```
[1] 29.85938
```

```
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))
```



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

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

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

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

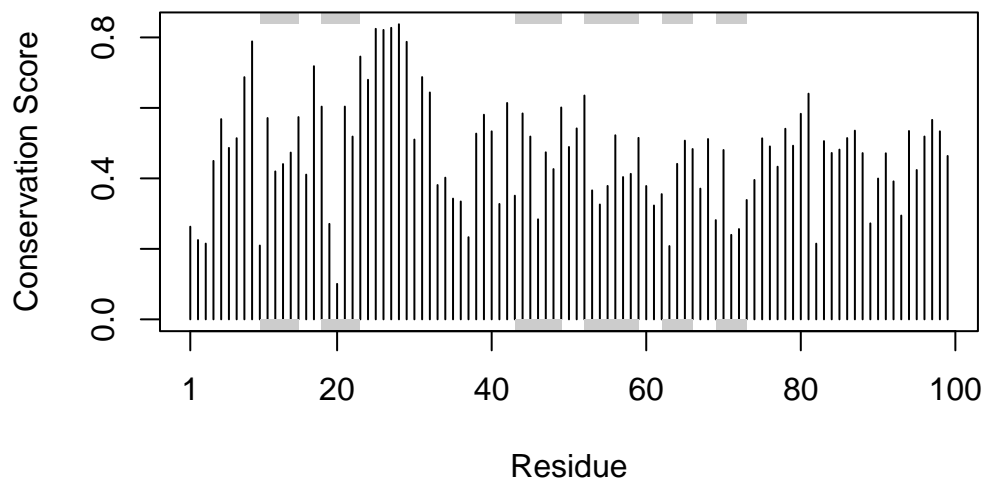
```
[1] 5378 132
```

```
pdb2 <- read.pdb("1hsg")
```

Note: Accessing on-line PDB file

```
sim <- conserv(aln)

plotb3(sim[1:99], sse=trim.pdb(pdb2, 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")
```

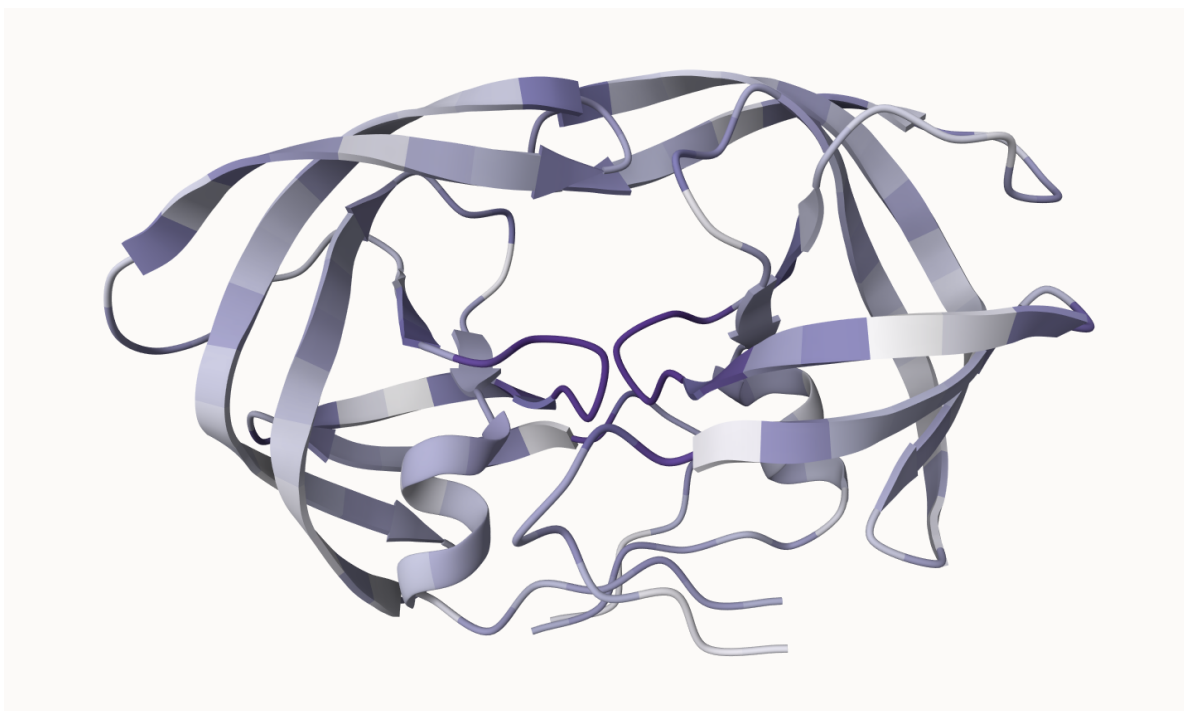


Figure 1: M1_Conserv

Exploration of my sequence for the Find a Gene Project:

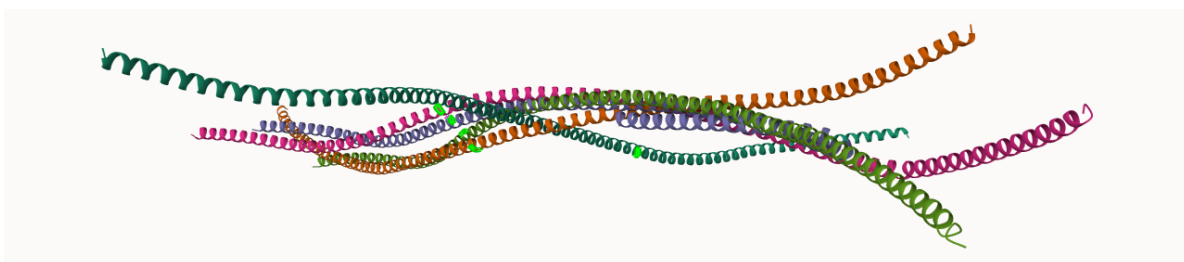


Figure 2: My_Gene

```
results_dir <- "test_2d266/"  
  
pdb_files_2 <- list.files(path=results_dir,  
                           pattern="*.pdb",  
                           full.names = TRUE)
```

```
basename(pdb_files_2)
```

```
[1] "test_2d266_unrelaxed_rank_001_alphafold2_ptm_model_3_seed_000.pdb"
[2] "test_2d266_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000.pdb"
[3] "test_2d266_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000.pdb"
[4] "test_2d266_unrelaxed_rank_004_alphafold2_ptm_model_5_seed_000.pdb"
[5] "test_2d266_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb"
```

```
pdb3_3 <- pdbaln(pdb_files_2, fit=TRUE, exefile="msa")
```

Reading PDB files:

```
test_2d266//test_2d266_unrelaxed_rank_001_alphafold2_ptm_model_3_seed_000.pdb
test_2d266//test_2d266_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000.pdb
test_2d266//test_2d266_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000.pdb
test_2d266//test_2d266_unrelaxed_rank_004_alphafold2_ptm_model_5_seed_000.pdb
test_2d266//test_2d266_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb
.....
```

Extracting sequences

```
pdb/seq: 1   name: test_2d266//test_2d266_unrelaxed_rank_001_alphafold2_ptm_model_3_seed_000
pdb/seq: 2   name: test_2d266//test_2d266_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000
pdb/seq: 3   name: test_2d266//test_2d266_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000
pdb/seq: 4   name: test_2d266//test_2d266_unrelaxed_rank_004_alphafold2_ptm_model_5_seed_000
pdb/seq: 5   name: test_2d266//test_2d266_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000
```

```
rd_2 <- rmsd(pdb3_3, fit=T)
```

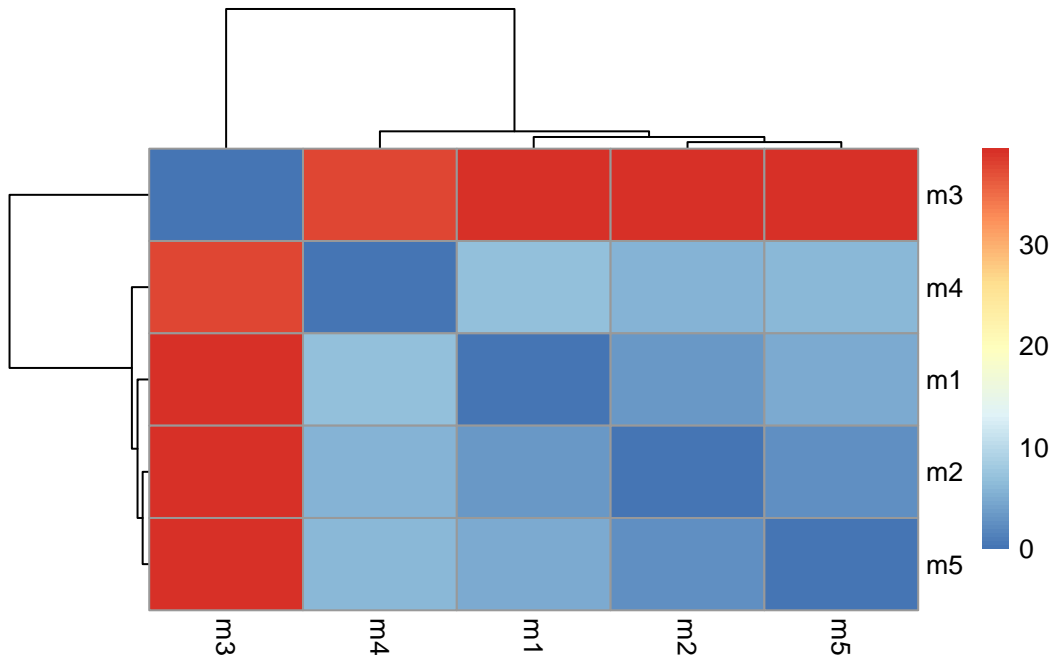
Warning in rmsd(pdb3_3, fit = T): No indices provided, using the 260 non NA positions

```
range(rd_2)
```

```
[1] 0.000 39.448
```

```
library(pheatmap)
```

```
colnames(rd_2) <- paste0("m",1:5)
rownames(rd_2 ) <- paste0("m",1:5)
pheatmap(rd_2)
```



```
pdb_my_gene <- read.pdb("3K5E")
```

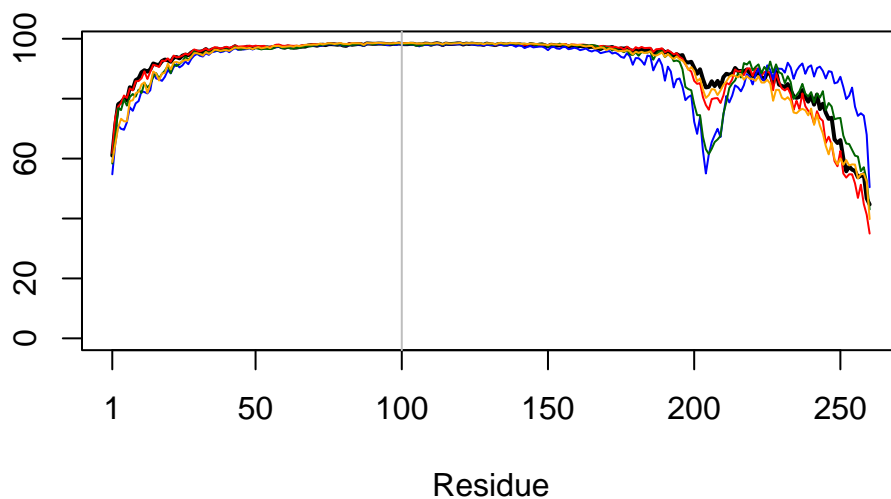
Note: Accessing on-line PDB file

PDB has ALT records, taking A only, rm.alt=TRUE

```
plotb3(pdb3_3$b[1,], typ="l", lwd=2, sse=pdb_my_gene)
```

Warning in plotb3(pdb3_3\$b[1,], typ = "l", lwd = 2, sse = pdb_my_gene): Length of input 'sse' does not equal the length of input 'x'; Ignoring 'sse'

```
points(pdb3_3$b[2,], typ="l", col="red")
points(pdb3_3$b[3,], typ="l", col="blue")
points(pdb3_3$b[4,], typ="l", col="darkgreen")
points(pdb3_3$b[5,], typ="l", col="orange")
abline(v=100, col="gray")
```



```
core_2 <- core.find(pdb3_3)
```

```
core size 259 of 260  vol = 49718.24
core size 258 of 260  vol = 45982.01
core size 257 of 260  vol = 43081.15
core size 256 of 260  vol = 40173.56
core size 255 of 260  vol = 37102.53
core size 254 of 260  vol = 34546.24
core size 253 of 260  vol = 32570.54
core size 252 of 260  vol = 30352.69
core size 251 of 260  vol = 28289.87
core size 250 of 260  vol = 26447.75
core size 249 of 260  vol = 24835.52
core size 248 of 260  vol = 23215.76
core size 247 of 260  vol = 21542.09
core size 246 of 260  vol = 20250.29
core size 245 of 260  vol = 18956.64
core size 244 of 260  vol = 17630.52
core size 243 of 260  vol = 16532.81
core size 242 of 260  vol = 15400.35
core size 241 of 260  vol = 14737
core size 240 of 260  vol = 14055.4
```


core size 239 of 260 vol = 13223.35
core size 238 of 260 vol = 12340.46
core size 237 of 260 vol = 11508.32
core size 236 of 260 vol = 10794.94
core size 235 of 260 vol = 10352.5
core size 234 of 260 vol = 9843.522
core size 233 of 260 vol = 9301.555
core size 232 of 260 vol = 8671.782
core size 231 of 260 vol = 8032.037
core size 230 of 260 vol = 7472.601
core size 229 of 260 vol = 6861.517
core size 228 of 260 vol = 6396.072
core size 227 of 260 vol = 5897.367
core size 226 of 260 vol = 5466.552
core size 225 of 260 vol = 4987.3
core size 224 of 260 vol = 4476.321
core size 223 of 260 vol = 3950.685
core size 222 of 260 vol = 3568.507
core size 221 of 260 vol = 3153.677
core size 220 of 260 vol = 2822.584
core size 219 of 260 vol = 2525.494
core size 218 of 260 vol = 2210.306
core size 217 of 260 vol = 1948.481
core size 216 of 260 vol = 1691.163
core size 215 of 260 vol = 1463.35
core size 214 of 260 vol = 1253.553
core size 213 of 260 vol = 1069.611
core size 212 of 260 vol = 942.042
core size 211 of 260 vol = 816.649
core size 210 of 260 vol = 682.85
core size 209 of 260 vol = 583.21
core size 208 of 260 vol = 533.69
core size 207 of 260 vol = 488.196
core size 206 of 260 vol = 459.062
core size 205 of 260 vol = 433.548
core size 204 of 260 vol = 410.607
core size 203 of 260 vol = 388.121
core size 202 of 260 vol = 342.493
core size 201 of 260 vol = 320.517
core size 200 of 260 vol = 299.888
core size 199 of 260 vol = 282.924
core size 198 of 260 vol = 266.009
core size 197 of 260 vol = 251.475

core size 196 of 260	vol = 227.753
core size 195 of 260	vol = 206.032
core size 194 of 260	vol = 193.747
core size 193 of 260	vol = 180.7
core size 192 of 260	vol = 168.557
core size 191 of 260	vol = 159.055
core size 190 of 260	vol = 148.96
core size 189 of 260	vol = 140.484
core size 188 of 260	vol = 130.829
core size 187 of 260	vol = 123.932
core size 186 of 260	vol = 117.726
core size 185 of 260	vol = 110.678
core size 184 of 260	vol = 104.02
core size 183 of 260	vol = 97.825
core size 182 of 260	vol = 92.557
core size 181 of 260	vol = 86.72
core size 180 of 260	vol = 82.005
core size 179 of 260	vol = 76.824
core size 178 of 260	vol = 71.847
core size 177 of 260	vol = 66.932
core size 176 of 260	vol = 62.637
core size 175 of 260	vol = 58.394
core size 174 of 260	vol = 54.557
core size 173 of 260	vol = 51.363
core size 172 of 260	vol = 48.135
core size 171 of 260	vol = 45.12
core size 170 of 260	vol = 42.271
core size 169 of 260	vol = 39.26
core size 168 of 260	vol = 36.353
core size 167 of 260	vol = 33.998
core size 166 of 260	vol = 31.802
core size 165 of 260	vol = 29.686
core size 164 of 260	vol = 27.975
core size 163 of 260	vol = 26.303
core size 162 of 260	vol = 24.456
core size 161 of 260	vol = 22.909
core size 160 of 260	vol = 21.726
core size 159 of 260	vol = 20.498
core size 158 of 260	vol = 18.78
core size 157 of 260	vol = 17.758
core size 156 of 260	vol = 16.78
core size 155 of 260	vol = 15.677
core size 154 of 260	vol = 14.69

core size 153 of 260	vol = 13.864
core size 152 of 260	vol = 12.986
core size 151 of 260	vol = 12.151
core size 150 of 260	vol = 11.54
core size 149 of 260	vol = 11.016
core size 148 of 260	vol = 10.601
core size 147 of 260	vol = 10.083
core size 146 of 260	vol = 9.689
core size 145 of 260	vol = 9.187
core size 144 of 260	vol = 8.775
core size 143 of 260	vol = 8.435
core size 142 of 260	vol = 8.068
core size 141 of 260	vol = 7.68
core size 140 of 260	vol = 7.412
core size 139 of 260	vol = 7.175
core size 138 of 260	vol = 6.954
core size 137 of 260	vol = 6.725
core size 136 of 260	vol = 6.436
core size 135 of 260	vol = 6.209
core size 134 of 260	vol = 5.988
core size 133 of 260	vol = 5.756
core size 132 of 260	vol = 5.52
core size 131 of 260	vol = 5.3
core size 130 of 260	vol = 5.067
core size 129 of 260	vol = 4.821
core size 128 of 260	vol = 4.608
core size 127 of 260	vol = 4.413
core size 126 of 260	vol = 4.21
core size 125 of 260	vol = 3.998
core size 124 of 260	vol = 3.82
core size 123 of 260	vol = 3.672
core size 122 of 260	vol = 3.535
core size 121 of 260	vol = 3.392
core size 120 of 260	vol = 3.23
core size 119 of 260	vol = 3.049
core size 118 of 260	vol = 2.875
core size 117 of 260	vol = 2.729
core size 116 of 260	vol = 2.626
core size 115 of 260	vol = 2.501
core size 114 of 260	vol = 2.41
core size 113 of 260	vol = 2.287
core size 112 of 260	vol = 2.21
core size 111 of 260	vol = 2.133

```

core size 110 of 260  vol = 2.043
core size 109 of 260  vol = 1.971
core size 108 of 260  vol = 1.9
core size 107 of 260  vol = 1.809
core size 106 of 260  vol = 1.738
core size 105 of 260  vol = 1.668
core size 104 of 260  vol = 1.594
core size 103 of 260  vol = 1.509
core size 102 of 260  vol = 1.437
core size 101 of 260  vol = 1.369
core size 100 of 260  vol = 1.295
core size 99 of 260   vol = 1.221
core size 98 of 260   vol = 1.14
core size 97 of 260   vol = 1.064
core size 96 of 260   vol = 1.032
core size 95 of 260   vol = 0.969
core size 94 of 260   vol = 0.904
core size 93 of 260   vol = 0.843
core size 92 of 260   vol = 0.789
core size 91 of 260   vol = 0.725
core size 90 of 260   vol = 0.686
core size 89 of 260   vol = 0.662
core size 88 of 260   vol = 0.639
core size 87 of 260   vol = 0.619
core size 86 of 260   vol = 0.597
core size 85 of 260   vol = 0.58
core size 84 of 260   vol = 0.555
core size 83 of 260   vol = 0.537
core size 82 of 260   vol = 0.507
core size 81 of 260   vol = 0.483
FINISHED: Min vol ( 0.5 ) reached

```

```
core.inds_2 <- print(core, vol=0.5)
```

```

# 89 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    10  42     33
2    44  50      7
3    52  66     15
4    69  77      9
5    80  98     19

```

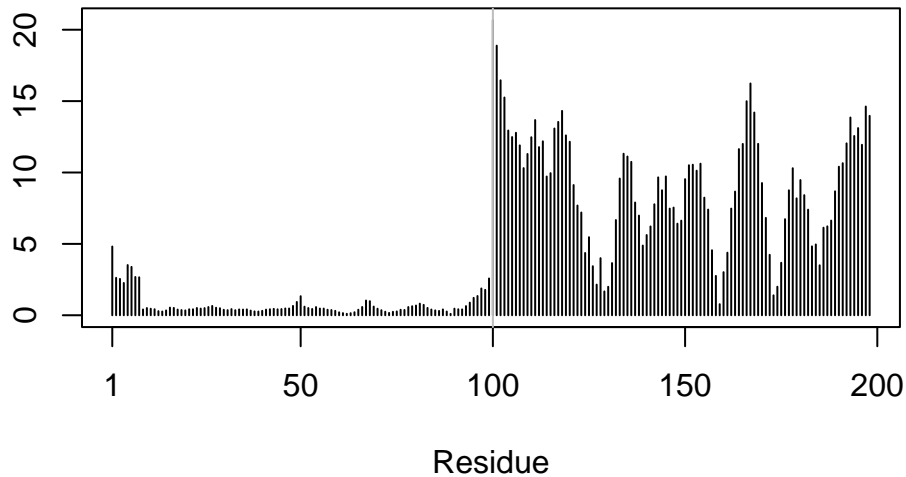
```
xyz_2 <- pdbfit(pdb, core.inds, outpath="corefit_structures")
```

```
rf_2 <- rmsf(xyz_2)
```

```
plotb3(rf_2, sse=pdb_my_gene)
```

Warning in plotb3(rf_2, sse = pdb_my_gene): Length of input 'sse' does not equal the length of input 'x'; Ignoring 'sse'

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



```
pae_files_2 <- list.files(path=results_dir,  
                           pattern=".*model.*\\.json",  
                           full.names = TRUE)
```

```
pae1_2 <- read_json(pae_files_2[1],simplifyVector = TRUE)  
pae5_2 <- read_json(pae_files_2[5],simplifyVector = TRUE)
```

```
attributes(pae1_2)
```

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

```
head(pae1_2$plddt)
```

```
[1] 61.00 71.00 78.12 78.81 79.88 79.44
```

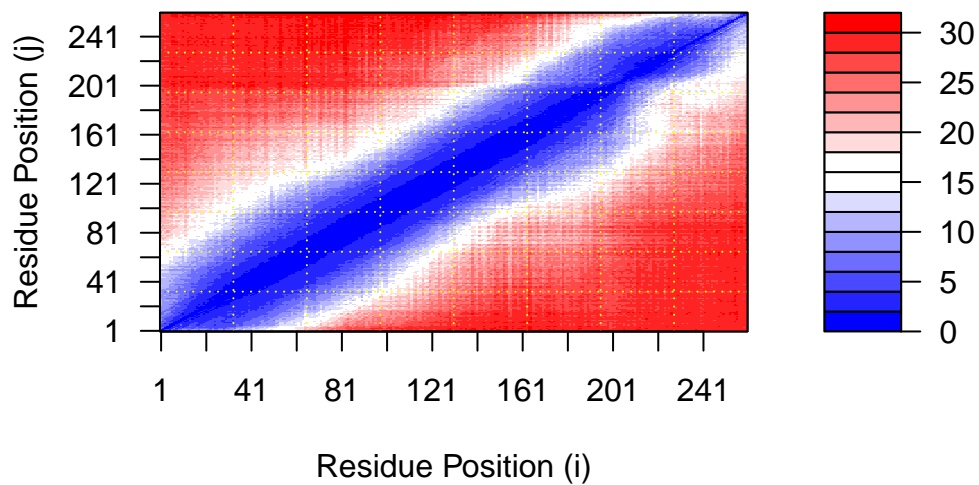
```
pae1_2$max_pae
```

```
[1] 31.14062
```

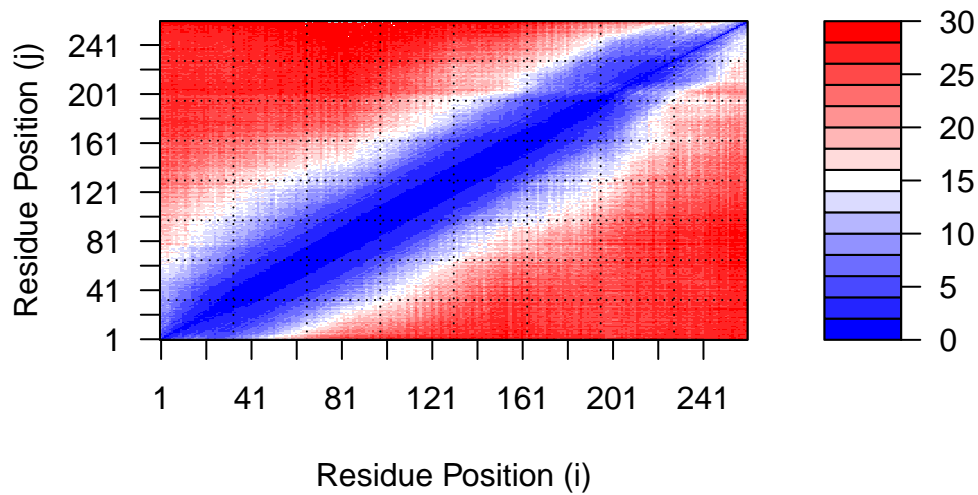
```
pae5_2$max_pae
```

```
[1] 30.8125
```

```
plot.dmat(pae1_2$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```



```
plot.dmat(pae5_2$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```



```
aln_file_2 <- list.files(path=results_dir,
                        pattern=".a3m$",
                        full.names = TRUE)
aln_file_2
```

```
[1] "test_2d266//test_2d266.a3m"
```

```
aln_2 <- read.fasta(aln_file_2[1], to.upper = TRUE)
```

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

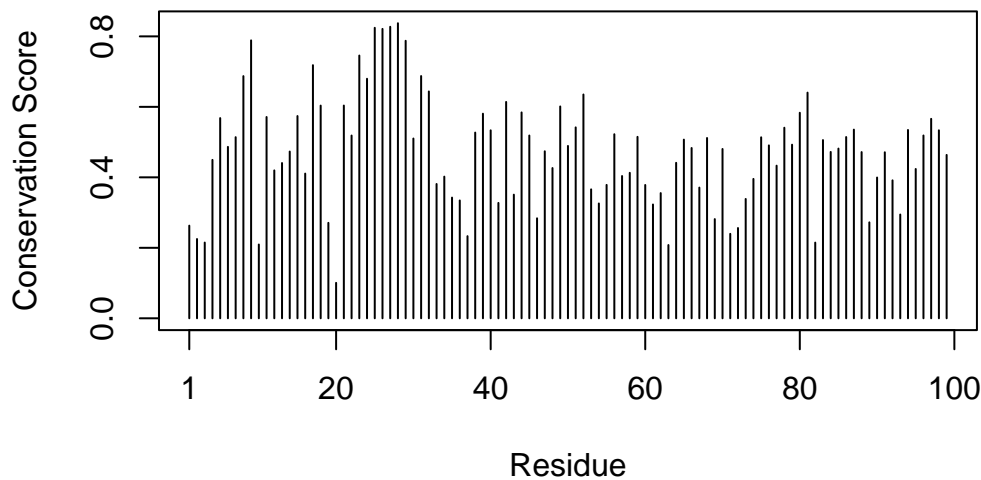
```
dim(aln_2$ali)
```

```
[1] 929 465
```

```
sim_2 <- conserv(aln_2)
```

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

Warning in plotb3(sim[1:99], sse = trim.pdb(pdb_my_gene, chain = "A"), ylab = "Conservation Score"): Length of input 'sse' does not equal the length of input 'x'; Ignoring 'sse'



```
con_2 <- consensus(aln, cutoff = 0.9)
con_2$seq
```

```
[1] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[19] "-" "-" "-" "-" "-" "-" "D" "T" "G" "A" "-" "-" "-" "-" "-" "-" "-" "-"
[37] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[55] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[73] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[91] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[109] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[127] "-" "-" "-" "-" "-" "-"
```