

# Classlab 11: Bioinformatics (pt2. Focus on new AlphaFold2)

Nathaniel Nono (PID: A16782656)

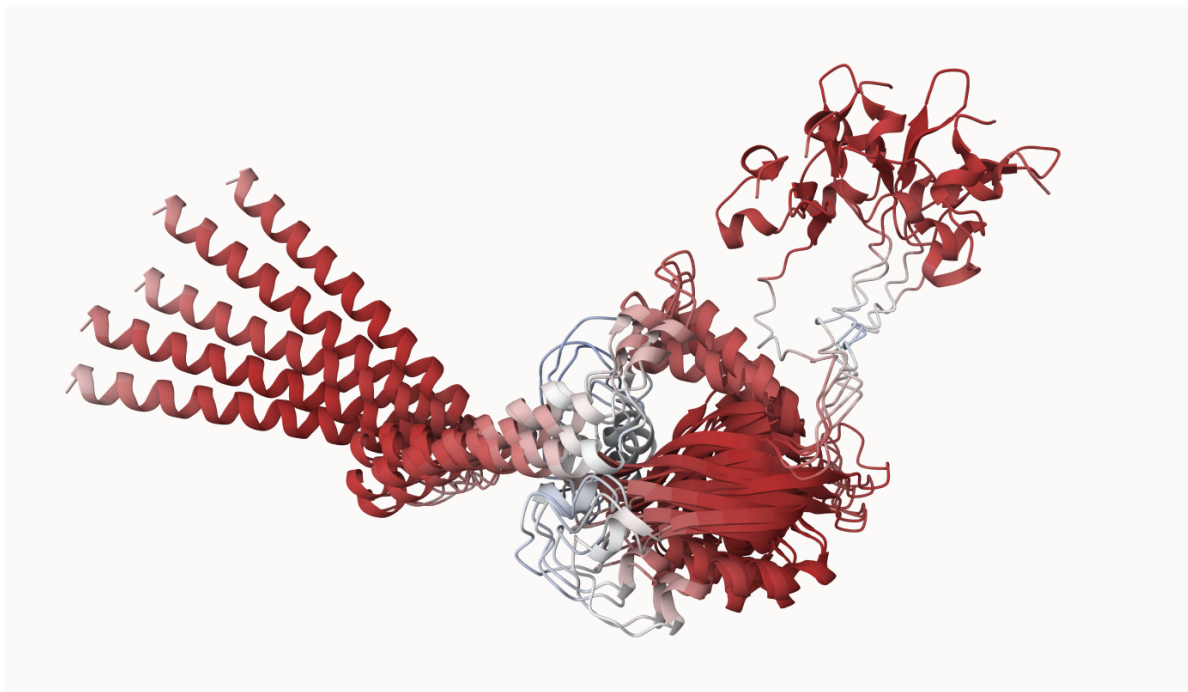


Figure 1: 'First superpose model'

## Custom analysis of resulting models

```
# Change this for YOUR results dir name  
results_dir <- "clock_novel_cDNA_89b2f/"
```

```
# File names for all PDB models
pdb_files <- list.files(path=results_dir,
                        pattern="*.pdb",
                        full.names = TRUE)

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

```
[1] "clock_novel_cDNA_89b2f_unrelaxed_rank_001_alphafold2_ptm_model_3_seed_000.pdb"
[2] "clock_novel_cDNA_89b2f_unrelaxed_rank_002_alphafold2_ptm_model_5_seed_000.pdb"
[3] "clock_novel_cDNA_89b2f_unrelaxed_rank_003_alphafold2_ptm_model_4_seed_000.pdb"
[4] "clock_novel_cDNA_89b2f_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.pdb"
[5] "clock_novel_cDNA_89b2f_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb"
```

```
library(bio3d)

# Read all data from Models
# and superpose/fit coords
pdbs <- pdbaln(pdb_files, fit=TRUE, exefile="msa")
```

Reading PDB files:

```
clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_001_alphafold2_ptm_model_3_seed_000.pdb
clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_002_alphafold2_ptm_model_5_seed_000.pdb
clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_003_alphafold2_ptm_model_4_seed_000.pdb
clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.pdb
clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb
.....
```

Extracting sequences

```
pdb/seq: 1    name: clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_001_alphafold2_ptm_model_3_seed_000.pdb
pdb/seq: 2    name: clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_002_alphafold2_ptm_model_5_seed_000.pdb
pdb/seq: 3    name: clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_003_alphafold2_ptm_model_4_seed_000.pdb
pdb/seq: 4    name: clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.pdb
pdb/seq: 5    name: clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb
```

```
pdbs
```

```

                                1           .           .           .           .           50
[Truncated_Name:1]clock_nove  MEDEKDKAKRVSRNKSEKKRRDQFNVLIKELGTMLPGNTRKMDKSTILQ
```

```

[Truncated_Name:2] clock_nove MEDEKDKAKRVSRNKSEKKRRDQFNVLIKELGTMLPGNTRKMDKSTILQ
[Truncated_Name:3] clock_nove MEDEKDKAKRVSRNKSEKKRRDQFNVLIKELGTMLPGNTRKMDKSTILQ
[Truncated_Name:4] clock_nove MEDEKDKAKRVSRNKSEKKRRDQFNVLIKELGTMLPGNTRKMDKSTILQ
[Truncated_Name:5] clock_nove MEDEKDKAKRVSRNKSEKKRRDQFNVLIKELGTMLPGNTRKMDKSTILQ
*****
1 . . . . 50

51 . . . . 100
[Truncated_Name:1] clock_nove KSIDFLRKHKKEIAAQSESSEIRQDWKPTFLSNEEFTQLMLEALDGFFLAI
[Truncated_Name:2] clock_nove KSIDFLRKHKKEIAAQSESSEIRQDWKPTFLSNEEFTQLMLEALDGFFLAI
[Truncated_Name:3] clock_nove KSIDFLRKHKKEIAAQSESSEIRQDWKPTFLSNEEFTQLMLEALDGFFLAI
[Truncated_Name:4] clock_nove KSIDFLRKHKKEIAAQSESSEIRQDWKPTFLSNEEFTQLMLEALDGFFLAI
[Truncated_Name:5] clock_nove KSIDFLRKHKKEIAAQSESSEIRQDWKPTFLSNEEFTQLMLEALDGFFLAI
*****
51 . . . . 100

101 . . . . 150
[Truncated_Name:1] clock_nove MTDGNIIYVSESVTSLEHLPSDLVDQNLNFLPAGEHSDVYKALSSHVL
[Truncated_Name:2] clock_nove MTDGNIIYVSESVTSLEHLPSDLVDQNLNFLPAGEHSDVYKALSSHVL
[Truncated_Name:3] clock_nove MTDGNIIYVSESVTSLEHLPSDLVDQNLNFLPAGEHSDVYKALSSHVL
[Truncated_Name:4] clock_nove MTDGNIIYVSESVTSLEHLPSDLVDQNLNFLPAGEHSDVYKALSSHVL
[Truncated_Name:5] clock_nove MTDGNIIYVSESVTSLEHLPSDLVDQNLNFLPAGEHSDVYKALSSHVL
*****
101 . . . . 150

151 . . . . 200
[Truncated_Name:1] clock_nove EGEPLTPEYMKTKNQLEFCCHMLRGTINPKEPPVYEVVKFIGNFKSLNNV
[Truncated_Name:2] clock_nove EGEPLTPEYMKTKNQLEFCCHMLRGTINPKEPPVYEVVKFIGNFKSLNNV
[Truncated_Name:3] clock_nove EGEPLTPEYMKTKNQLEFCCHMLRGTINPKEPPVYEVVKFIGNFKSLNNV
[Truncated_Name:4] clock_nove EGEPLTPEYMKTKNQLEFCCHMLRGTINPKEPPVYEVVKFIGNFKSLNNV
[Truncated_Name:5] clock_nove EGEPLTPEYMKTKNQLEFCCHMLRGTINPKEPPVYEVVKFIGNFKSLNNV
*****
151 . . . . 200

201 . . . . 250
[Truncated_Name:1] clock_nove PNSTLNGFDGVIQRSLHSSFGRVCFIATVRLAKPQFIKEMCTVDEPNEE
[Truncated_Name:2] clock_nove PNSTLNGFDGVIQRSLHSSFGRVCFIATVRLAKPQFIKEMCTVDEPNEE
[Truncated_Name:3] clock_nove PNSTLNGFDGVIQRSLHSSFGRVCFIATVRLAKPQFIKEMCTVDEPNEE
[Truncated_Name:4] clock_nove PNSTLNGFDGVIQRSLHSSFGRVCFIATVRLAKPQFIKEMCTVDEPNEE
[Truncated_Name:5] clock_nove PNSTLNGFDGVIQRSLHSSFGRVCFIATVRLAKPQFIKEMCTVDEPNEE
*****
201 . . . . 250

```

```

                251          .          .          .      283
[Truncated_Name:1]clock_nove  FTSRHSLEWKFLFLGHRAPPIIGYLPFEVLGTS
[Truncated_Name:2]clock_nove  FTSRHSLEWKFLFLGHRAPPIIGYLPFEVLGTS
[Truncated_Name:3]clock_nove  FTSRHSLEWKFLFLGHRAPPIIGYLPFEVLGTS
[Truncated_Name:4]clock_nove  FTSRHSLEWKFLFLGHRAPPIIGYLPFEVLGTS
[Truncated_Name:5]clock_nove  FTSRHSLEWKFLFLGHRAPPIIGYLPFEVLGTS
                *****
                251          .          .          .      283

```

Call:

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

Class:

```
pdb, fasta
```

Alignment dimensions:

```
5 sequence rows; 283 position columns (283 non-gap, 0 gap)
```

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

RMSD is a standard measure of structural distance between coordinate sets. We can use the `rmsd()` function to calculate the RMSD between all pairs models.

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

Warning in `rmsd(pdb, fit = T)`: No indices provided, using the 283 non NA positions

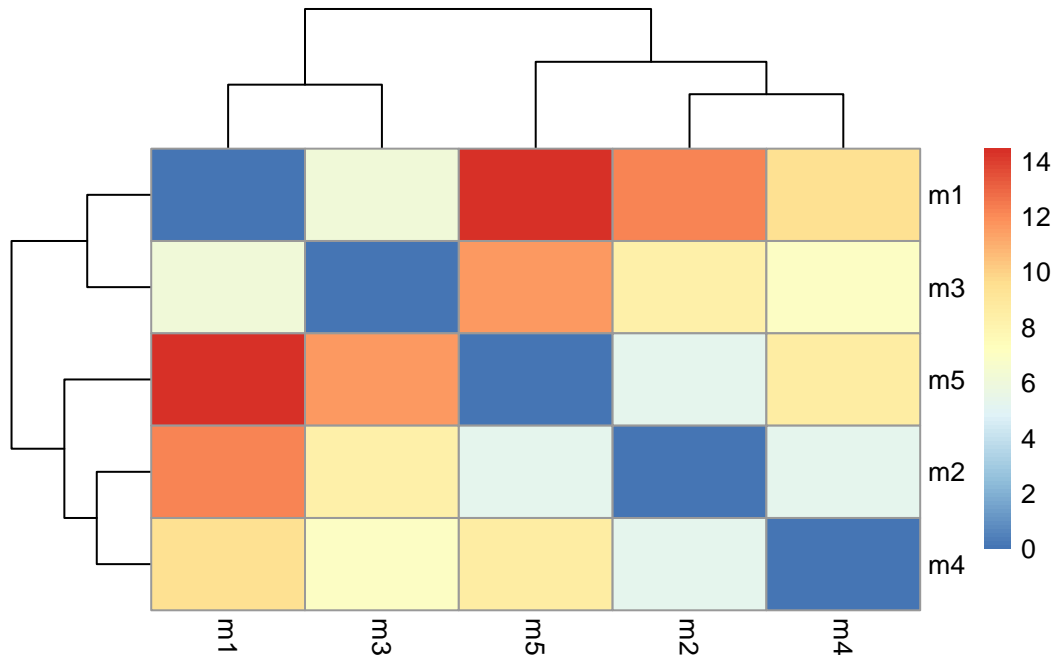
```
range(rd)
```

```
[1] 0.000 14.452
```

Draw a heatmap of these RMSD matrix values

```
library(pheatmap)

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



```
attributes(pdb)
```

```
$names
[1] "xyz" "resno" "b" "chain" "id" "ali" "resid" "sse" "call"

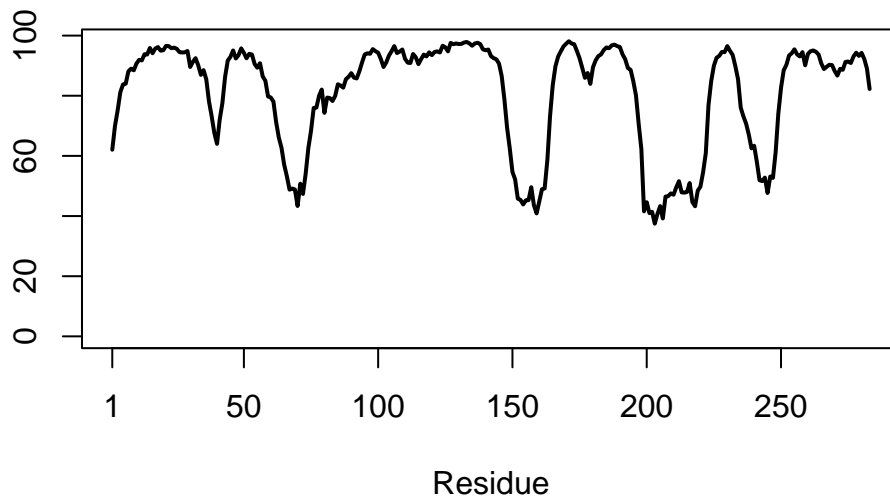
$class
[1] "pdb" "fasta"
```

```
# Read a reference PDB structure
pdb <- read.pdb("4dj3")
```

Note: Accessing on-line PDB file

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

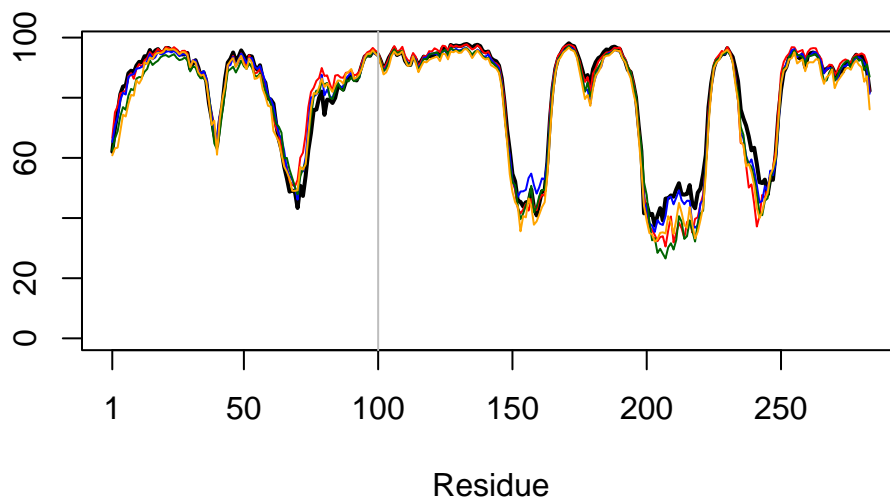
Warning in plotb3(pdb\$b[1, ], typ = "l", lwd = 2, sse = pdb): Length of input 'sse' does not equal the length of input 'x'; Ignoring 'sse'



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

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

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



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

```
core size 282 of 283  vol = 20544.9
core size 281 of 283  vol = 18816.35
core size 280 of 283  vol = 17565.9
core size 279 of 283  vol = 16298.31
core size 278 of 283  vol = 15433.82
core size 277 of 283  vol = 14564.22
core size 276 of 283  vol = 13600.94
core size 275 of 283  vol = 12955.81
core size 274 of 283  vol = 12359.64
core size 273 of 283  vol = 11830.95
core size 272 of 283  vol = 11348.58
core size 271 of 283  vol = 10936.55
core size 270 of 283  vol = 10490.37
core size 269 of 283  vol = 10154.68
core size 268 of 283  vol = 9782.75
core size 267 of 283  vol = 9483.431
core size 266 of 283  vol = 9195.352
core size 265 of 283  vol = 8870.478
core size 264 of 283  vol = 8581.094
core size 263 of 283  vol = 8278.86
```

core size	262 of 283	vol = 7919.665
core size	261 of 283	vol = 7609.741
core size	260 of 283	vol = 7293.702
core size	259 of 283	vol = 6924.674
core size	258 of 283	vol = 6529.204
core size	257 of 283	vol = 6157.332
core size	256 of 283	vol = 5810.81
core size	255 of 283	vol = 5492.234
core size	254 of 283	vol = 5153.4
core size	253 of 283	vol = 4799.889
core size	252 of 283	vol = 4433.97
core size	251 of 283	vol = 4107.033
core size	250 of 283	vol = 3777.283
core size	249 of 283	vol = 3441.592
core size	248 of 283	vol = 3143.493
core size	247 of 283	vol = 2890.459
core size	246 of 283	vol = 2698.582
core size	245 of 283	vol = 2500.211
core size	244 of 283	vol = 2307.258
core size	243 of 283	vol = 2147.26
core size	242 of 283	vol = 1956.565
core size	241 of 283	vol = 1771.225
core size	240 of 283	vol = 1643.302
core size	239 of 283	vol = 1513.55
core size	238 of 283	vol = 1423.619
core size	237 of 283	vol = 1338.604
core size	236 of 283	vol = 1267.642
core size	235 of 283	vol = 1198.471
core size	234 of 283	vol = 1128.144
core size	233 of 283	vol = 1082.093
core size	232 of 283	vol = 1040.031
core size	231 of 283	vol = 1007.592
core size	230 of 283	vol = 975.659
core size	229 of 283	vol = 958.153
core size	228 of 283	vol = 938.183
core size	227 of 283	vol = 914.287
core size	226 of 283	vol = 893.076
core size	225 of 283	vol = 878.785
core size	224 of 283	vol = 855.499
core size	223 of 283	vol = 828.196
core size	222 of 283	vol = 812.091
core size	221 of 283	vol = 790.87
core size	220 of 283	vol = 771.835



core size 219 of 283 vol = 748.049  
core size 218 of 283 vol = 730.98  
core size 217 of 283 vol = 714.853  
core size 216 of 283 vol = 694.06  
core size 215 of 283 vol = 680.213  
core size 214 of 283 vol = 662.753  
core size 213 of 283 vol = 645.933  
core size 212 of 283 vol = 630.122  
core size 211 of 283 vol = 616.921  
core size 210 of 283 vol = 600.73  
core size 209 of 283 vol = 588.179  
core size 208 of 283 vol = 576.459  
core size 207 of 283 vol = 563.093  
core size 206 of 283 vol = 550.355  
core size 205 of 283 vol = 539.188  
core size 204 of 283 vol = 529.126  
core size 203 of 283 vol = 518.694  
core size 202 of 283 vol = 509.441  
core size 201 of 283 vol = 500.585  
core size 200 of 283 vol = 489.269  
core size 199 of 283 vol = 478.172  
core size 198 of 283 vol = 465.454  
core size 197 of 283 vol = 456.349  
core size 196 of 283 vol = 446.106  
core size 195 of 283 vol = 436.315  
core size 194 of 283 vol = 426.952  
core size 193 of 283 vol = 414.768  
core size 192 of 283 vol = 405.54  
core size 191 of 283 vol = 396.063  
core size 190 of 283 vol = 387.531  
core size 189 of 283 vol = 378.423  
core size 188 of 283 vol = 366.239  
core size 187 of 283 vol = 353.248  
core size 186 of 283 vol = 344.257  
core size 185 of 283 vol = 335.91  
core size 184 of 283 vol = 326.663  
core size 183 of 283 vol = 318.061  
core size 182 of 283 vol = 309.413  
core size 181 of 283 vol = 299.87  
core size 180 of 283 vol = 290.826  
core size 179 of 283 vol = 281.199  
core size 178 of 283 vol = 270.943  
core size 177 of 283 vol = 260.26

core size 176 of 283	vol = 249.699
core size 175 of 283	vol = 237.916
core size 174 of 283	vol = 224.319
core size 173 of 283	vol = 208.43
core size 172 of 283	vol = 197.494
core size 171 of 283	vol = 185.646
core size 170 of 283	vol = 171.291
core size 169 of 283	vol = 156.519
core size 168 of 283	vol = 141.918
core size 167 of 283	vol = 126.992
core size 166 of 283	vol = 114.382
core size 165 of 283	vol = 101.197
core size 164 of 283	vol = 89.189
core size 163 of 283	vol = 79.647
core size 162 of 283	vol = 69.353
core size 161 of 283	vol = 60.74
core size 160 of 283	vol = 54.783
core size 159 of 283	vol = 49.946
core size 158 of 283	vol = 45.506
core size 157 of 283	vol = 41.624
core size 156 of 283	vol = 38.016
core size 155 of 283	vol = 34.799
core size 154 of 283	vol = 30.81
core size 153 of 283	vol = 28.021
core size 152 of 283	vol = 25.488
core size 151 of 283	vol = 22.826
core size 150 of 283	vol = 20.874
core size 149 of 283	vol = 18.903
core size 148 of 283	vol = 17.431
core size 147 of 283	vol = 15.708
core size 146 of 283	vol = 13.809
core size 145 of 283	vol = 12.395
core size 144 of 283	vol = 10.863
core size 143 of 283	vol = 9.72
core size 142 of 283	vol = 8.561
core size 141 of 283	vol = 7.881
core size 140 of 283	vol = 7.171
core size 139 of 283	vol = 6.408
core size 138 of 283	vol = 5.721
core size 137 of 283	vol = 5.034
core size 136 of 283	vol = 4.452
core size 135 of 283	vol = 3.81
core size 134 of 283	vol = 3.227

```

core size 133 of 283 vol = 2.871
core size 132 of 283 vol = 2.38
core size 131 of 283 vol = 1.966
core size 130 of 283 vol = 1.72
core size 129 of 283 vol = 1.447
core size 128 of 283 vol = 1.267
core size 127 of 283 vol = 1.124
core size 126 of 283 vol = 1
core size 125 of 283 vol = 0.882
core size 124 of 283 vol = 0.809
core size 123 of 283 vol = 0.722
core size 122 of 283 vol = 0.668
core size 121 of 283 vol = 0.609
core size 120 of 283 vol = 0.526
core size 119 of 283 vol = 0.467
FINISHED: Min vol ( 0.5 ) reached

```

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

```

# 120 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    77  77      1
2    79  80      2
3    85 150     66
4   163 198     36
5   223 237     15

```

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

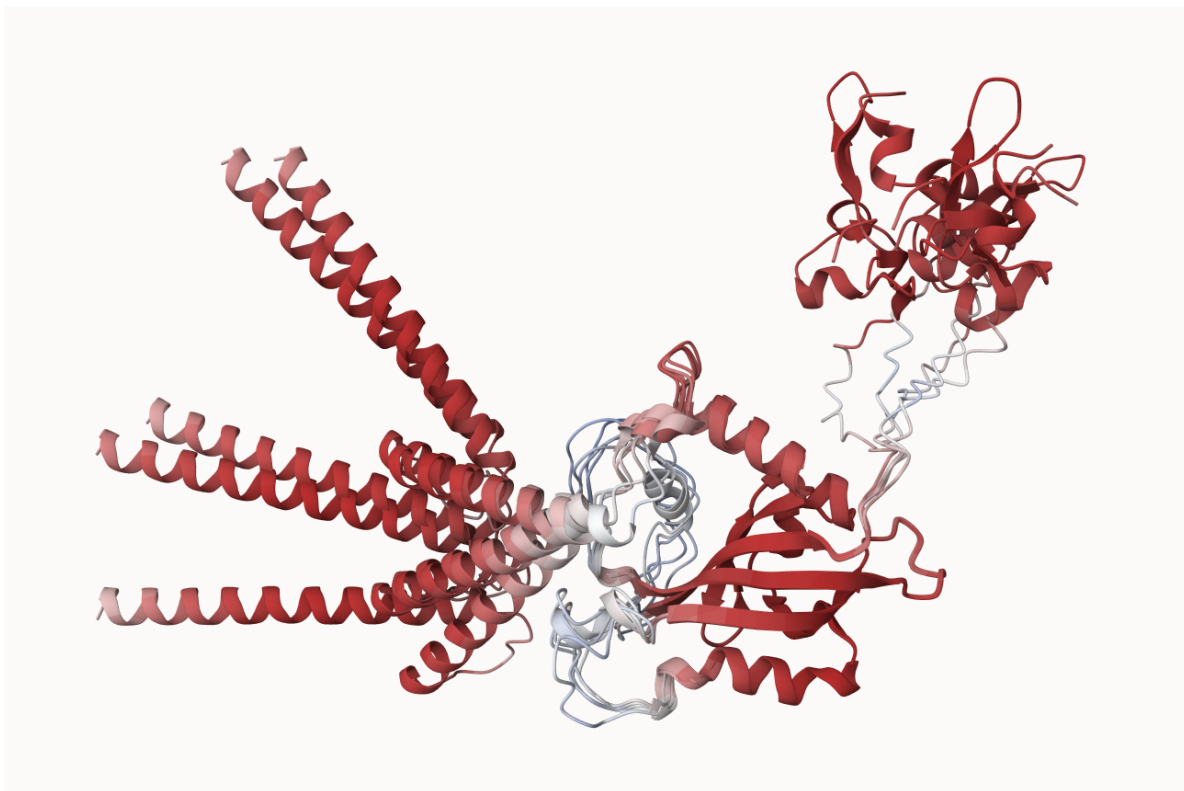
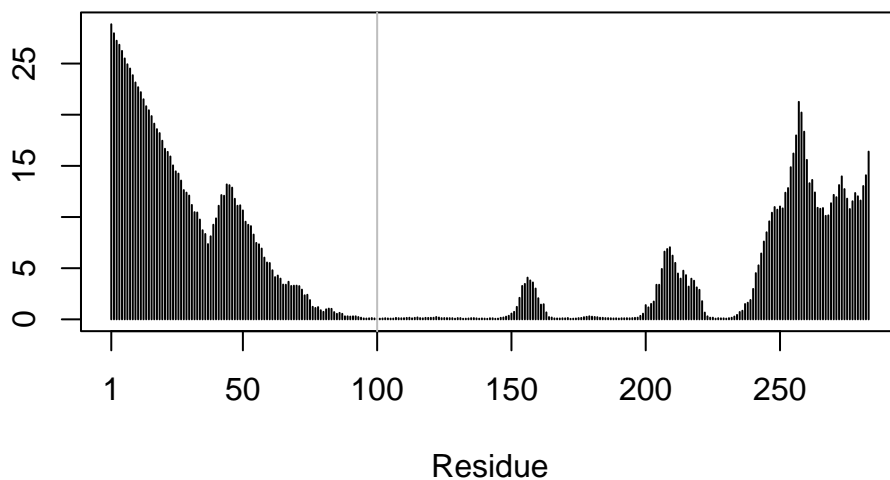


Figure 2: 'Core superpose structure model'

```
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")
```



### Predicted Alignment Error for domains

```
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"
```

```
# Per-residue pLDDT scores
```

```
# same as B-factor of PDB..
```

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

```
[1] 62.06 69.94 75.12 81.19 83.75 84.00
```

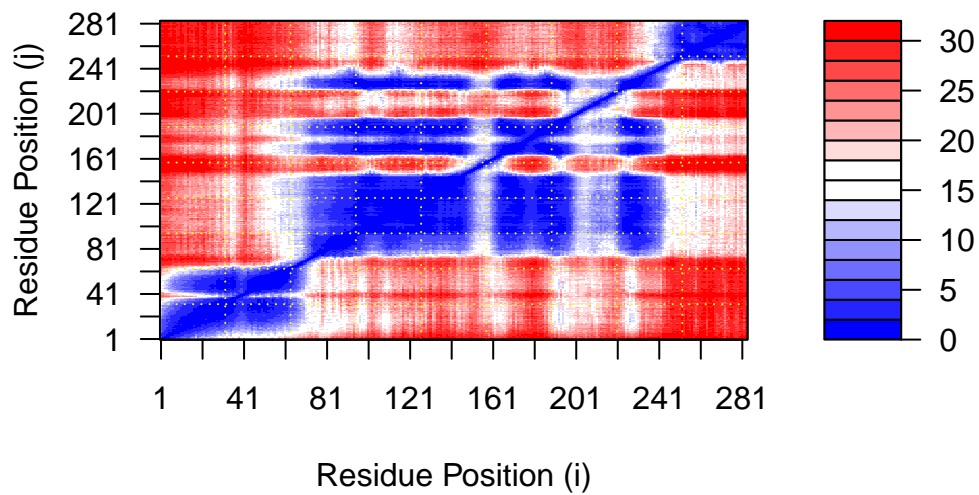
```
pae1$max_pae
```

```
[1] 31.17188
```

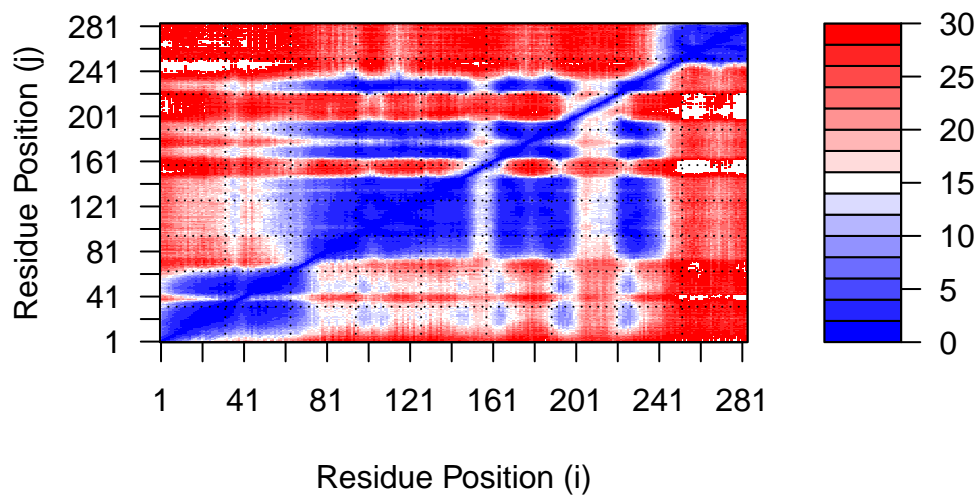
```
pae5$max_pae
```

```
[1] 31.1875
```

```
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))
```



### Residue conservation from alignment file

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

```
[1] "clock_novel_cDNA_89b2f//clock_novel_cDNA_89b2f.a3m"
```

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

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

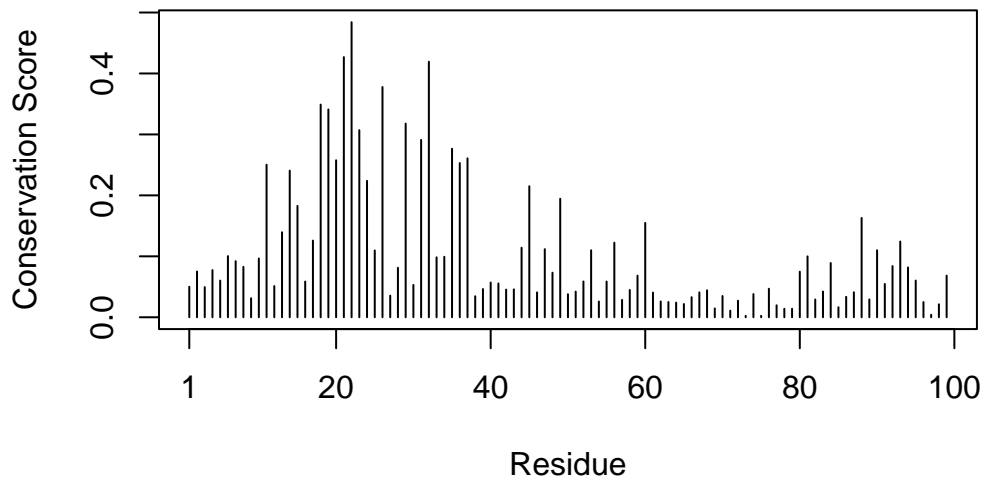
```
dim(aln$ali)
```

```
[1] 4153  552
```

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

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

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



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

```
[1] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[19] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[37] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[55] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[73] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[91] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[109] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[127] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[145] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
```



```

[163] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[181] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[199] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[217] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[235] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[253] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[271] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[289] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[307] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[325] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[343] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[361] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[379] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[397] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[415] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[433] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[451] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[469] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[487] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[505] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[523] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[541] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"

```

```

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")

```

