class11: AlphaFold2 Analysis

Jiachen Fan (A17662703)

Here we post process and inspect our modeling results from AlphaFold2.

```r
results_dir <- "hivprdimer_23119/"

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

```r
library(bio3d)

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

```
Reading PDB files:
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
.....

Extracting sequences

pdb/seq: 1   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_
pdb/seq: 2   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_
pdb/seq: 3   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_
pdb/seq: 4   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_
pdb/seq: 5   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_
```

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

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

```r
rd
```

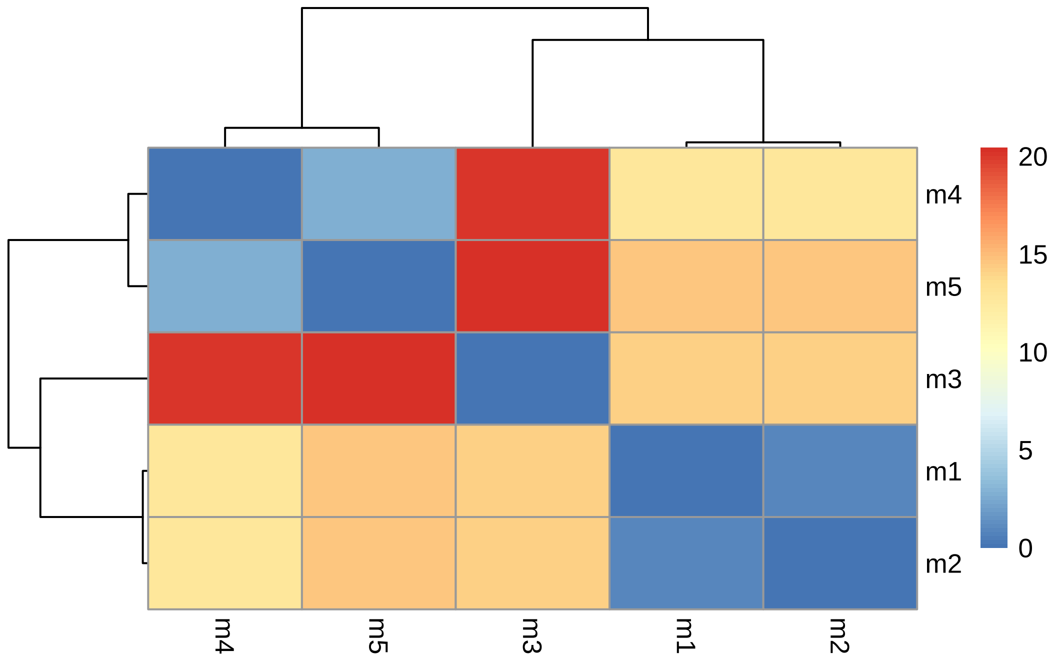
```
                                                                             hivprdimer_23119_
hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                             hivprdimer_23119_
hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                             hivprdimer_23119_
hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                             hivprdimer_23119_
hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                             hivprdimer_23119_
hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
```

```r
library(pheatmap)
```

```
Warning: package 'pheatmap' was built under R version 4.3.2
```

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

```
pheatmap(rd)
```



```
xyz <- pdbfit(pdbs, outfile="fitted")
```

A full atom based fitting or supperposition did not work very well because we have multiple chains that are in different conformations.

I want to focus our supperpostion on the ost invariant part

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

```
core size 197 of 198  vol = 6154.839
core size 196 of 198  vol = 5399.676
core size 195 of 198  vol = 5074.795
core size 194 of 198  vol = 4802.518
core size 193 of 198  vol = 4520.256
core size 192 of 198  vol = 4305.362
core size 191 of 198  vol = 4089.792
core size 190 of 198  vol = 3886.145
core size 189 of 198  vol = 3758.321
core size 188 of 198  vol = 3620.18
core size 187 of 198  vol = 3496.698
core size 186 of 198  vol = 3389.985
core size 185 of 198  vol = 3320.114
core size 184 of 198  vol = 3258.683
core size 183 of 198  vol = 3208.591
core size 182 of 198  vol = 3156.736
core size 181 of 198  vol = 3141.668
core size 180 of 198  vol = 3136.574
core size 179 of 198  vol = 3155.52
core size 178 of 198  vol = 3185.362
core size 177 of 198  vol = 3204.487
core size 176 of 198  vol = 3211.978
core size 175 of 198  vol = 3234.993
core size 174 of 198  vol = 3244.062
core size 173 of 198  vol = 3237.845
core size 172 of 198  vol = 3218.77
core size 171 of 198  vol = 3180.743
core size 170 of 198  vol = 3130.369
core size 169 of 198  vol = 3067.881
core size 168 of 198  vol = 2989.546
core size 167 of 198  vol = 2928.272
core size 166 of 198  vol = 2851.193
core size 165 of 198  vol = 2780.877
core size 164 of 198  vol = 2708.433
core size 163 of 198  vol = 2636.516
core size 162 of 198  vol = 2563.25
core size 161 of 198  vol = 2478.024
core size 160 of 198  vol = 2404.793
core size 159 of 198  vol = 2330.997
core size 158 of 198  vol = 2250.477
core size 157 of 198  vol = 2159.432
core size 156 of 198  vol = 2070.759
core size 155 of 198  vol = 1983.579
core size 154 of 198  vol = 1917.913
core size 153 of 198  vol = 1842.556
core size 152 of 198  vol = 1775.398
core size 151 of 198  vol = 1695.133
core size 150 of 198  vol = 1632.173
core size 149 of 198  vol = 1570.391
core size 148 of 198  vol = 1497.238
core size 147 of 198  vol = 1434.802
core size 146 of 198  vol = 1367.706
core size 145 of 198  vol = 1302.596
core size 144 of 198  vol = 1251.985
core size 143 of 198  vol = 1207.976
core size 142 of 198  vol = 1167.112
core size 141 of 198  vol = 1118.27
core size 140 of 198  vol = 1081.664
core size 139 of 198  vol = 1029.75
core size 138 of 198  vol = 981.766
core size 137 of 198  vol = 944.446
core size 136 of 198  vol = 899.224
core size 135 of 198  vol = 859.402
core size 134 of 198  vol = 814.694
core size 133 of 198  vol = 771.862
core size 132 of 198  vol = 733.807
core size 131 of 198  vol = 702.053
core size 130 of 198  vol = 658.757
core size 129 of 198  vol = 622.574
core size 128 of 198  vol = 578.29
core size 127 of 198  vol = 543.07
core size 126 of 198  vol = 510.934
core size 125 of 198  vol = 481.595
core size 124 of 198  vol = 464.672
core size 123 of 198  vol = 451.721
core size 122 of 198  vol = 430.417
core size 121 of 198  vol = 409.141
core size 120 of 198  vol = 378.942
core size 119 of 198  vol = 348.325
core size 118 of 198  vol = 324.738
core size 117 of 198  vol = 312.394
core size 116 of 198  vol = 300.89
core size 115 of 198  vol = 279.976
core size 114 of 198  vol = 263.434
core size 113 of 198  vol = 250.263
core size 112 of 198  vol = 229.592
core size 111 of 198  vol = 209.929
core size 110 of 198  vol = 196.379
core size 109 of 198  vol = 180.628
core size 108 of 198  vol = 167.088
core size 107 of 198  vol = 155.875
core size 106 of 198  vol = 142.595
core size 105 of 198  vol = 128.924
core size 104 of 198  vol = 114.054
core size 103 of 198  vol = 100.936
core size 102 of 198  vol = 90.431
core size 101 of 198  vol = 81.972
core size 100 of 198  vol = 74.017
core size 99 of 198  vol = 66.855
core size 98 of 198  vol = 59.525
core size 97 of 198  vol = 52.263
core size 96 of 198  vol = 43.699
core size 95 of 198  vol = 35.813
core size 94 of 198  vol = 28.888
core size 93 of 198  vol = 20.692
core size 92 of 198  vol = 14.975
core size 91 of 198  vol = 9.146
core size 90 of 198  vol = 5.232
core size 89 of 198  vol = 3.53
core size 88 of 198  vol = 2.657
core size 87 of 198  vol = 1.998
core size 86 of 198  vol = 1.333
core size 85 of 198  vol = 1.141
core size 84 of 198  vol = 1.012
core size 83 of 198  vol = 0.891
core size 82 of 198  vol = 0.749
core size 81 of 198  vol = 0.618
core size 80 of 198  vol = 0.538
core size 79 of 198  vol = 0.479
FINISHED: Min vol ( 0.5 ) reached
```

```r
core.inds <- core
```

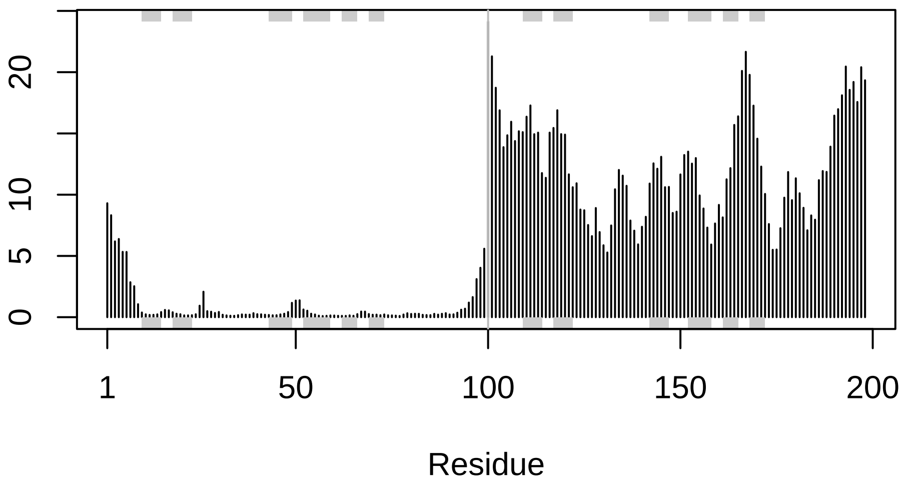
```r
xyz <- pdbfit(pdbs, inds = core.inds, outpath = 'core_fitted')
```

```r
# Read a reference PDB structure
pdb <- read.pdb("1hsg")
```

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

```r
rf <- rmsf(xyz)

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



To evaluate how good multi-chain or multi-domain models are wee need to look at the PAE scores (predicted aligned error)

These are output as JSON format files.

```r
pae_files <- list.files(results_dir, pattern ="0.json", full.names = T)
pae_files
```

```
[1] "hivprdimer_23119/hivprdimer_23119_scores_rank_001_alphafold2_multimer_v3_model_1_seed_0
[2] "hivprdimer_23119/hivprdimer_23119_scores_rank_002_alphafold2_multimer_v3_model_5_seed_0
[3] "hivprdimer_23119/hivprdimer_23119_scores_rank_003_alphafold2_multimer_v3_model_4_seed_0
[4] "hivprdimer_23119/hivprdimer_23119_scores_rank_004_alphafold2_multimer_v3_model_2_seed_0
[5] "hivprdimer_23119/hivprdimer_23119_scores_rank_005_alphafold2_multimer_v3_model_3_seed_0
```

```r
library(jsonlite)

pae1 <- read_json(pae_files[1], simplifyVector = TRUE)
pae5 <- read_json(pae_files[5], simplifyVector = TRUE)
```

```r
attributes(pae1)
```

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

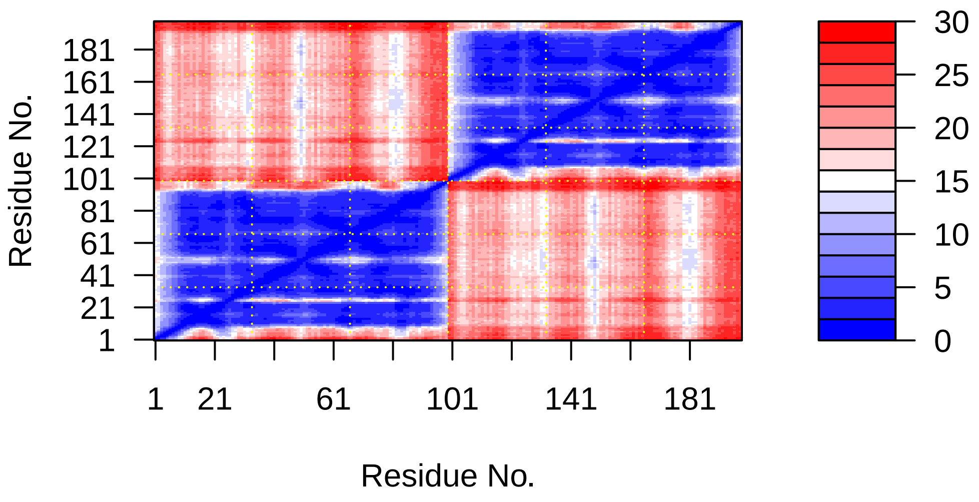
```
pae1$max_pae
```

```
[1] 15.54688
```

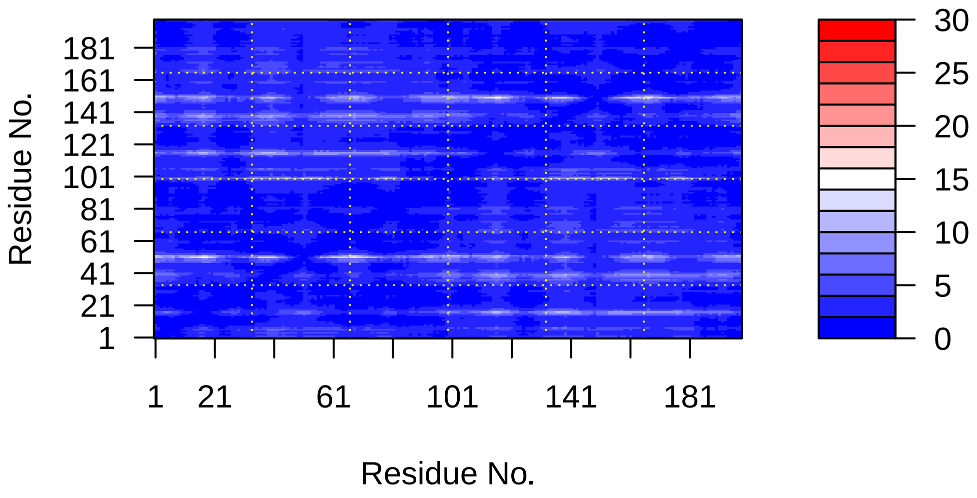
```
pae5$max_pae
```

```
[1] 29.29688
```

```
plot.dmat(pae5$pae,
          xlab = 'Residue No.',
          ylab = 'Residue No.',
          zlim =c(0, 30))
```



```
plot.dmat(pae1$pae,
          xlab = 'Residue No.',
          ylab = 'Residue No.',
          zlim =c(0, 30))
```



Main points

We can run AlphaFold on colab We can read these results into R and process to help make sense of these models and their PAE and pLDDT scores.

Residue conservation from alignment file

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

```
[1] "hivprdimer_23119/hivprdimer_23119.a3m"
```

```r
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 this alignment

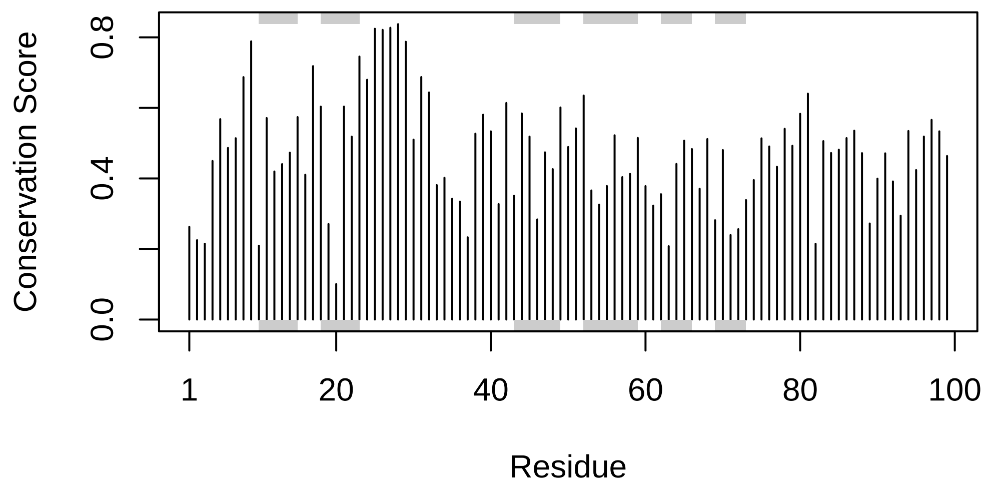
```r
dim(aln$ali)
```

```
[1] 5378  132
```

We can score residue conservation in the alignment with the conserv() function

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

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



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

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

For a final visualization of these functionally important sites we can map this conservation score to the Occupancy column of a PDB file for viewing in molecular viewer programs such as Mol*, PyMol, VMD, chimera etc.

```r
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")
```

```r
sessionInfo()
```

```
R version 4.3.1 (2023-06-16 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 11 x64 (build 22621)

Matrix products: default


locale:
[1] LC_COLLATE=Chinese (Simplified)_China.utf8
[2] LC_CTYPE=Chinese (Simplified)_China.utf8
[3] LC_MONETARY=Chinese (Simplified)_China.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_China.utf8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
```

```
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] jsonlite_1.8.7  pheatmap_1.0.12 bio3d_2.4-4

loaded via a namespace (and not attached):
 [1] crayon_1.5.2            cli_3.6.1               knitr_1.45
 [4] rlang_1.1.2             xfun_0.40               glue_1.6.2
 [7] S4Vectors_0.40.1        colorspace_2.1-0        RCurl_1.98-1.13
[10] Biostrings_2.70.1       htmltools_0.5.7         stats4_4.3.1
[13] scales_1.2.1            rmarkdown_2.25          grid_4.3.1
[16] munsell_0.5.0           evaluate_0.23           bitops_1.0-7
[19] fastmap_1.1.1           lifecycle_1.0.4         yaml_2.3.7
[22] IRanges_2.36.0          GenomeInfoDb_1.38.0     compiler_4.3.1
[25] RColorBrewer_1.1-3      Rcpp_1.0.11             XVector_0.42.0
[28] rstudioapi_0.15.0       digest_0.6.33           R6_2.5.1
[31] parallel_4.3.1          GenomeInfoDbData_1.2.11 gtable_0.3.4
[34] tools_4.3.1             zlibbioc_1.48.0         msa_1.34.0
[37] BiocGenerics_0.48.1
```

References

Baek, Minkyung, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, et al. 2021. "Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network." Science 373 (6557): 871–76. https://doi.org/10.1126/science.abj8754. Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, et al. 2021. "Highly Accurate Protein Structure Prediction with AlphaFold." Nature 596 (7873): 583–89. https://doi.org/10.1038/s41586-021-03819-2. Mirdita, Milot, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. 2022. "ColabFold: Making Protein Folding Accessible to All." Nature Methods 19 (6): 679–82. https://doi.org/10.1038/s41592-022-01488-1.