Find a Gene Project: Alphafold

Yoonjin Lim (PID: A16850635)

Table of contents

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

```
library(bio3d)

pth <- "QuerySequence_93902/"
pdb.files <- list.files(path=pth, full.names= TRUE, pattern=".pdb")
```

Align and supperpose all these models.

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

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

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

```
Reading PDB files:
QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_001_alphafold2_ptm_model_5_seed_000.p
QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000.p
QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_003_alphafold2_ptm_model_3_seed_000.p
QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.p
QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.p
.....

Extracting sequences
```

```
pdb/seq: 1   name: QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_001_alphafold2_pt
pdb/seq: 2   name: QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_002_alphafold2_pt
pdb/seq: 3   name: QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_003_alphafold2_pt
pdb/seq: 4   name: QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_004_alphafold2_pt
pdb/seq: 5   name: QuerySequence_93902//QuerySequence_93902_unrelaxed_rank_005_alphafold2_pt
```

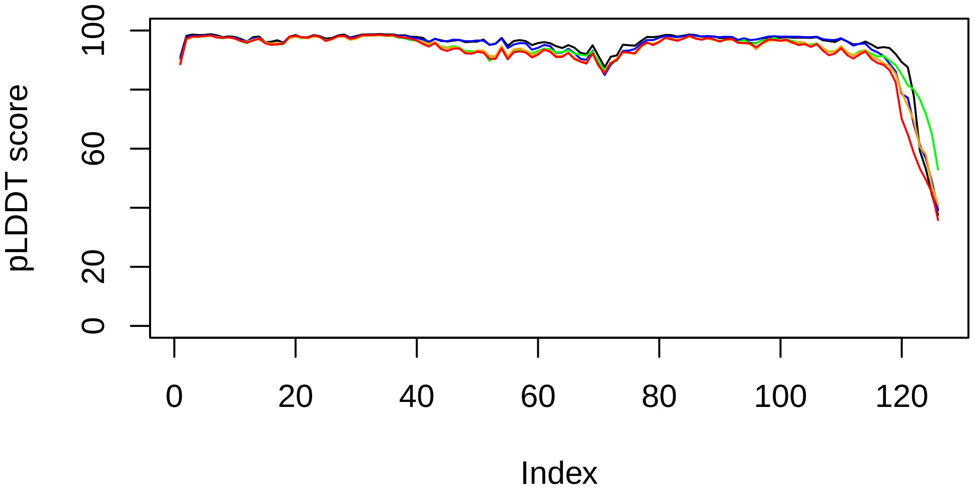
```r
pdbs
```

```
                               1        .         .         .         .         50
[Truncated_Name:1]QuerySeque   QVLFRFVTAHPEYQKKFSKFATVPQNELLGNGNFLAQAYTILAGLNVVVQ
[Truncated_Name:2]QuerySeque   QVLFRFVTAHPEYQKKFSKFATVPQNELLGNGNFLAQAYTILAGLNVVVQ
[Truncated_Name:3]QuerySeque   QVLFRFVTAHPEYQKKFSKFATVPQNELLGNGNFLAQAYTILAGLNVVVQ
[Truncated_Name:4]QuerySeque   QVLFRFVTAHPEYQKKFSKFATVPQNELLGNGNFLAQAYTILAGLNVVVQ
[Truncated_Name:5]QuerySeque   QVLFRFVTAHPEYQKKFSKFATVPQNELLGNGNFLAQAYTILAGLNVVVQ
                               **************************************************
                               1        .         .         .         .         50

                              51       .         .         .         .         100
[Truncated_Name:1]QuerySeque   SLSSQELLANQLNALGGAHQARGVTPIMFEQFGEILTGVLAEELGGAFNA
[Truncated_Name:2]QuerySeque   SLSSQELLANQLNALGGAHQARGVTPIMFEQFGEILTGVLAEELGGAFNA
[Truncated_Name:3]QuerySeque   SLSSQELLANQLNALGGAHQARGVTPIMFEQFGEILTGVLAEELGGAFNA
[Truncated_Name:4]QuerySeque   SLSSQELLANQLNALGGAHQARGVTPIMFEQFGEILTGVLAEELGGAFNA
[Truncated_Name:5]QuerySeque   SLSSQELLANQLNALGGAHQARGVTPIMFEQFGEILTGVLAEELGGAFNA
                               **************************************************
                              51       .         .         .         .         100

                             101        .         .     126
[Truncated_Name:1]QuerySeque   EAQSAWKSGLAALVAGVSKTLKIRGF
[Truncated_Name:2]QuerySeque   EAQSAWKSGLAALVAGVSKTLKIRGF
[Truncated_Name:3]QuerySeque   EAQSAWKSGLAALVAGVSKTLKIRGF
[Truncated_Name:4]QuerySeque   EAQSAWKSGLAALVAGVSKTLKIRGF
[Truncated_Name:5]QuerySeque   EAQSAWKSGLAALVAGVSKTLKIRGF
                               **************************
                             101        .         .     126

Call:
  pdbaln(files = pdb.files, fit = TRUE, exefile = "msa")

Class:
  pdbs, fasta
```

```
Alignment dimensions:
  5 sequence rows; 126 position columns (126 non-gap, 0 gap)

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

```
#view.pdbs(pdbs)
```

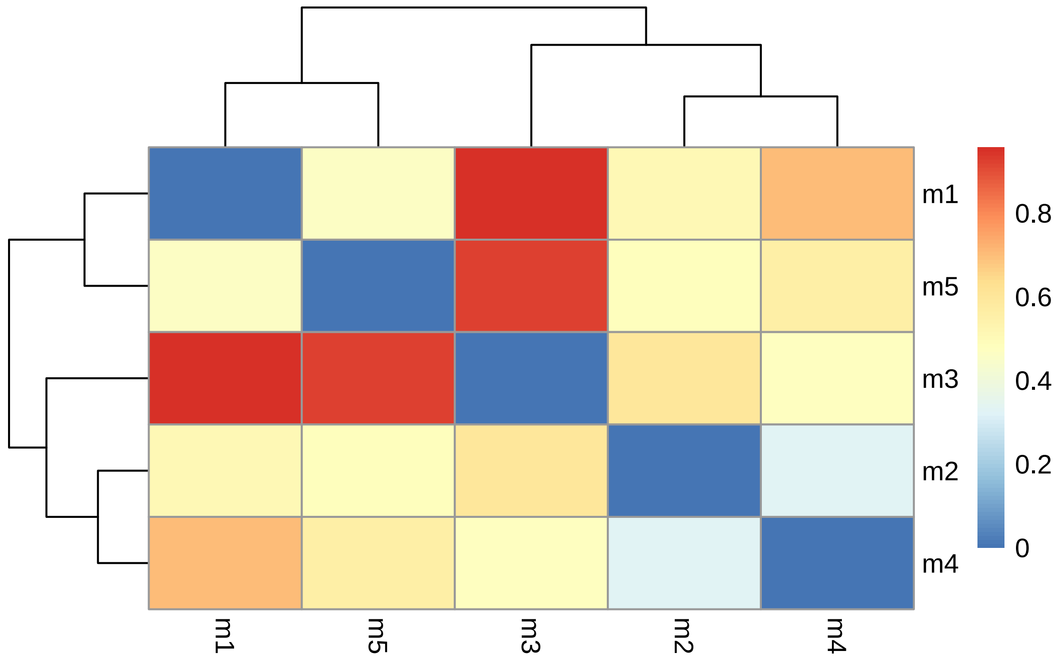
```
plot(pdbs$b[1,], typ ="l", ylim=c(0,100), ylab="pLDDT score")
lines(pdbs$b[2,], typ = "l", col="blue")
lines(pdbs$b[3,], typ = "l", col="green")
lines(pdbs$b[4,], typ = "l", col="orange")
lines(pdbs$b[5,], typ = "l", col="red")
```



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

```
Warning in rmsd(pdbs): No indices provided, using the 126 non NA positions
```

```
library(pheatmap)

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



Predicted Alignment Error for domains

```r
library(jsonlite)

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

```
[1] "QuerySequence_93902//QuerySequence_93902_scores_rank_001_alphafold2_ptm_model_5_seed_00
[2] "QuerySequence_93902//QuerySequence_93902_scores_rank_002_alphafold2_ptm_model_4_seed_00
[3] "QuerySequence_93902//QuerySequence_93902_scores_rank_003_alphafold2_ptm_model_3_seed_00
[4] "QuerySequence_93902//QuerySequence_93902_scores_rank_004_alphafold2_ptm_model_1_seed_00
[5] "QuerySequence_93902//QuerySequence_93902_scores_rank_005_alphafold2_ptm_model_2_seed_00
```

```r
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] 91.19 98.19 98.62 98.44 98.50 98.75
```

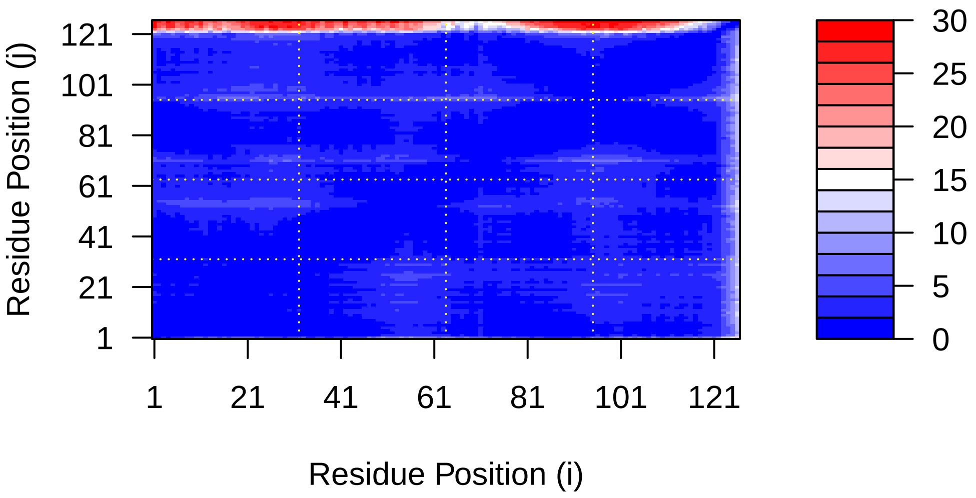
```
pae1$max_pae
```

```
[1] 29.35938
```

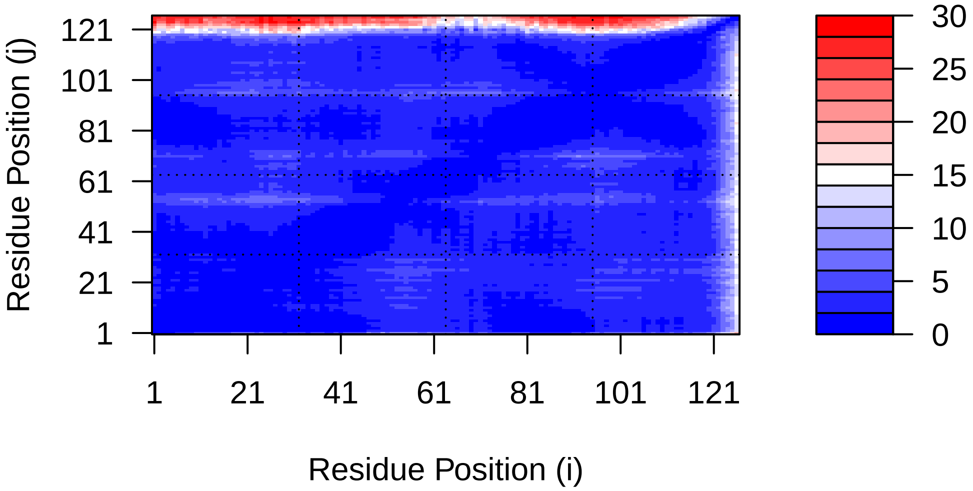
```
pae5$max_pae
```

```
[1] 29.10938
```

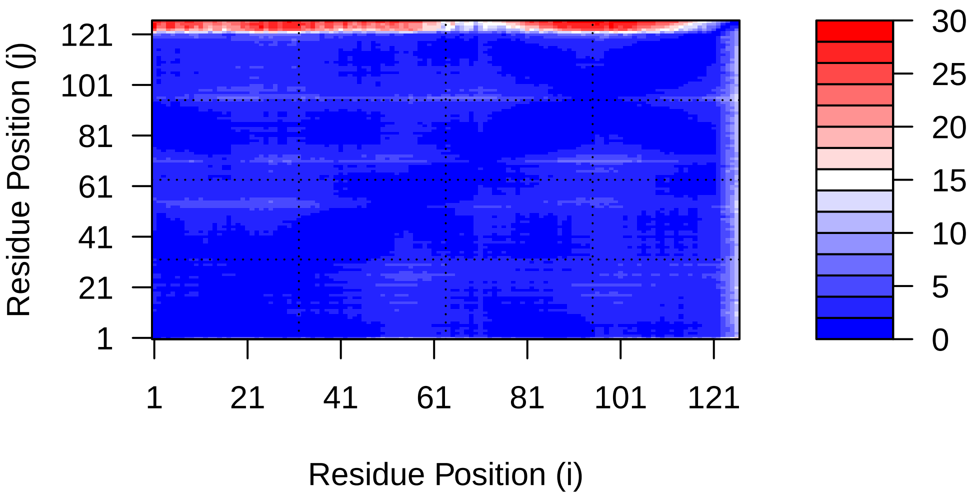
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```



```
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))
```



Score Residue conservation from alignment file

AlphaFold returns its large alignment file used for analysis. Here we read this file and score conservation per position.

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

```
[1] "QuerySequence_93902//QuerySequence_93902.a3m"
```

Read the alignment file.

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

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

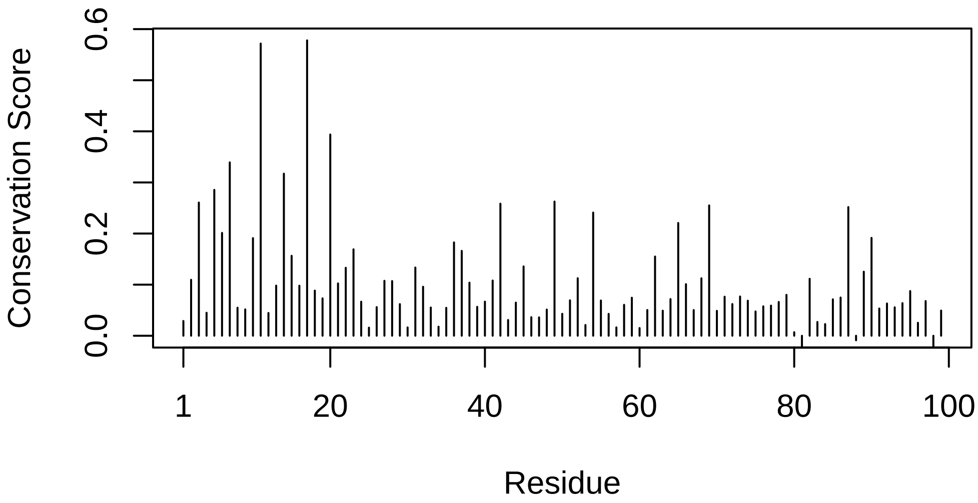
```
dim(aln$ali)
```

```
[1] 1992  177
```

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

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

```r
plotb3(sim[1:99],
       ylab="Conservation Score")
```



Find the consensus sequence at a very high cut-off to find invariant residues.

```r
con <- consensus(aln, cutoff = 0.7)
con$seq
```

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

```
[145] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[163] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
```

Notes: P11 and F17 seems like to be a conserved residue.