Structural Bioinformatics (pt 2)

Melissa Guereca PID: A16511023

AlphaFold has changed the game for protein structure prediction and allows anyone with sufficient bioinformatics skills to predict the structure of virtually any protein.

We ran AlphaFold via GoogleColan at: https://github.com/sokrypton/ColabFold

In particular we used their AlphaFold2_mmseqs version that uses mmseqs2 rather than HMMMer for sequence search.

The main outputs include a set of **PDB structure files** along with matching **JSON format files** that tell us how good the resulting models might be.

Let's start by loading the PDB structures up in Mol*

```
# Change this for YOUR results dir name
results_dir <- "HIVPrDimer_23119/"
```

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

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

```
[1] "HIVPrDimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
[2] "HIVPrDimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "HIVPrDimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "HIVPrDimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
[5] "HIVPrDimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb"
```

```
library(bio3d)
```

```
# Read all data from Models
#  and superpose/fit coords
pdbs <- pdbaln(pdb_files[1:2], fit=TRUE, exefile="msa" )
```

```
Reading PDB files:
HIVPrDimer_23119//HIVPrDimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_00
HIVPrDimer_23119//HIVPrDimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_00
..

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
```

```
pdbs
```

```
                              1        .         .         .         .         50
[Truncated_Name:1]HIVPrDimer  PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]HIVPrDimer  PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                              **************************************************
                              1        .         .         .         .         50

                             51        .         .         .         .         100
[Truncated_Name:1]HIVPrDimer  GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]HIVPrDimer  GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                              **************************************************
                             51        .         .         .         .         100

                            101        .         .         .         .         150
[Truncated_Name:1]HIVPrDimer  QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:2]HIVPrDimer  QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
                              **************************************************
                            101        .         .         .         .         150

                            151        .         .         .         .       198
[Truncated_Name:1]HIVPrDimer  GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]HIVPrDimer  GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
                              *************************************************
                            151        .         .         .         .       198
```

```
Call:
  pdbaln(files = pdb_files[1:2], fit = TRUE, exefile = "msa")

Class:
  pdbs, fasta

Alignment dimensions:
  2 sequence rows; 198 position columns (198 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(pdbs, fit=T)
```

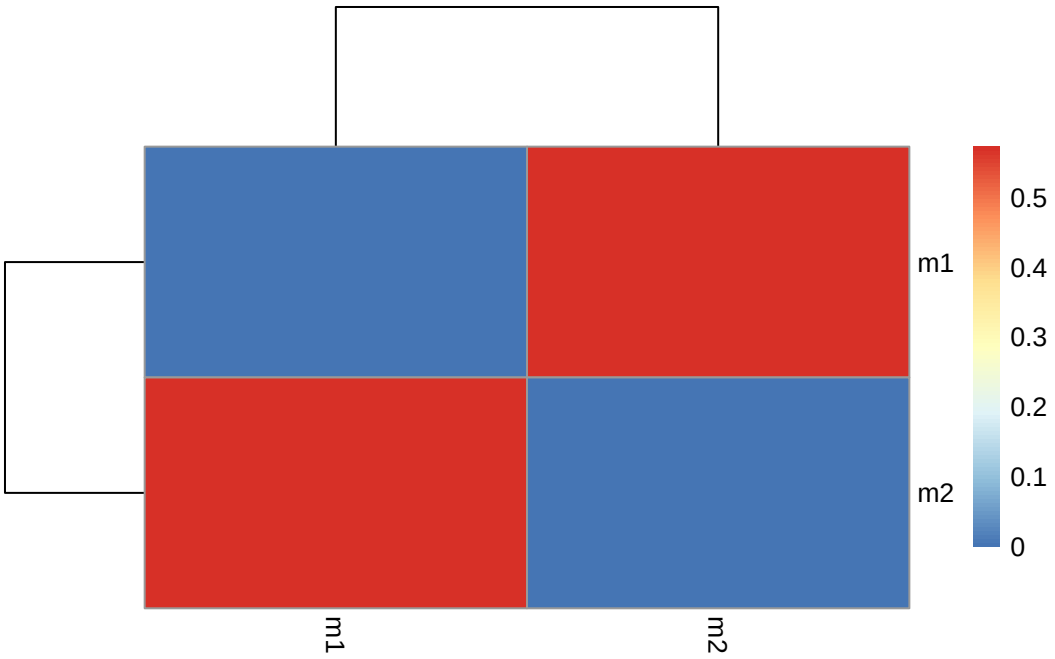
```
Warning in rmsd(pdbs, fit = T): No indices provided, using the 198 non NA positions
```

```
range(rd)
```

```
[1] 0.000 0.572
```

Heatmap of these RMSD matrix values

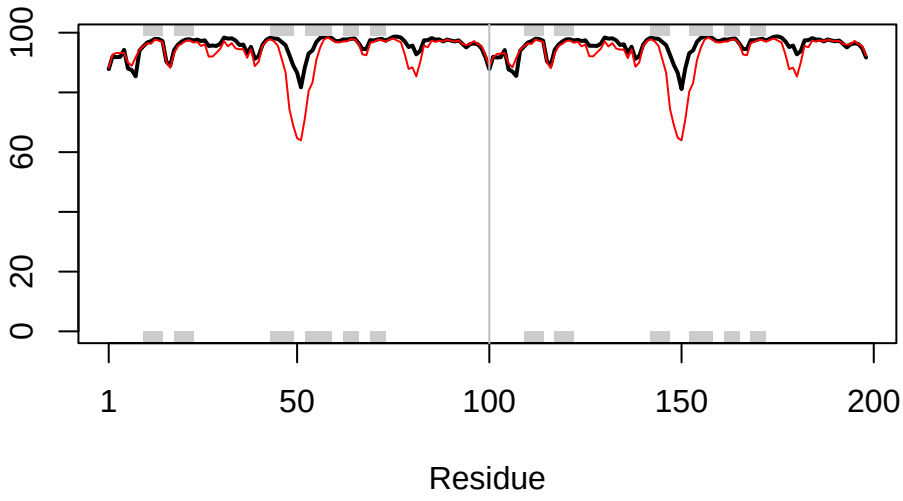
```
library(pheatmap)

colnames(rd) <- paste0("m",1:2)
rownames(rd) <- paste0("m",1:2)
pheatmap(rd)
```




```
# Read 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")
```




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

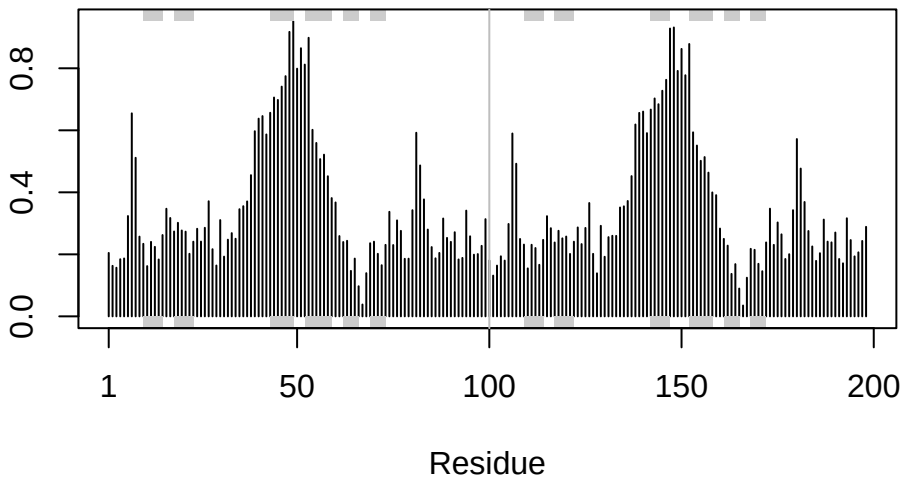
```
 core size 197 of 198  vol = 0
 FINISHED: Min vol ( 0.5 ) reached
```

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

```
# 198 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1     1  99     99
```

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

```
rf <- rmsf(xyz)

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



If the predicted model has more than one domain, each domain may have high confidence, yet the relative positions of the domains may not, The estimated reliability of relative domain positions is in graphs of predicted aligned error (PAE)…

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"     "iptm"
```

```
# Per-residue pLDDT scores
#  same as B-factor of PDB..
head(pae1$plddt)
```

```
[1] 87.81 92.00 91.81 91.88 94.25 88.00
```

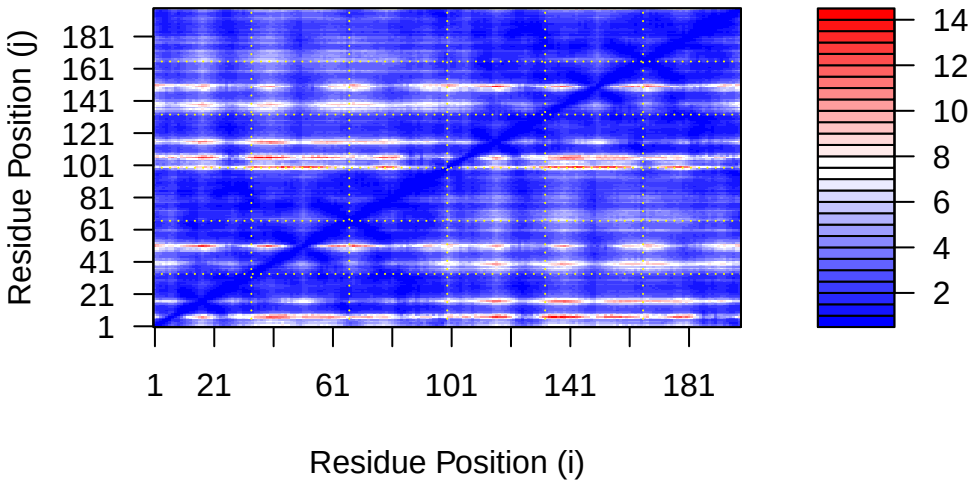
```
pae1$max_pae
```

```
[1] 14.09375
```

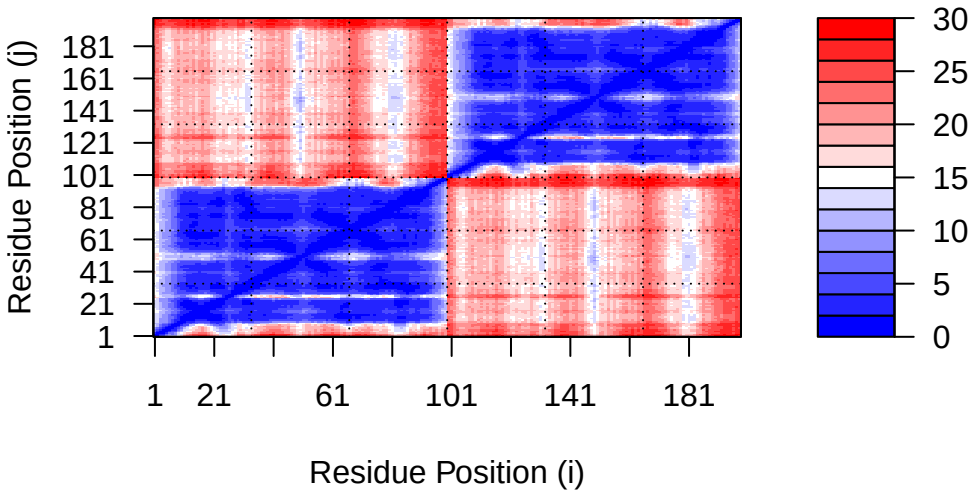
```
pae5$max_pae
```

```
[1] 29.29688
```

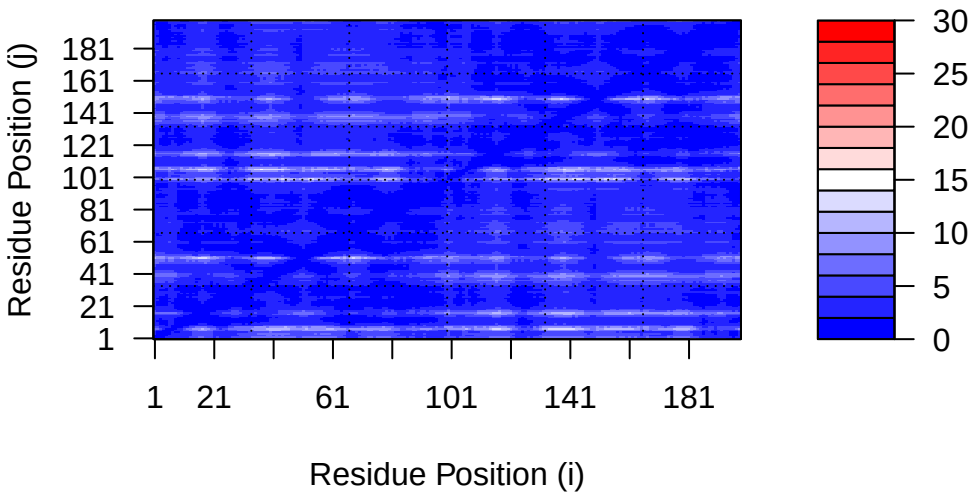
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```



```
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] "HIVPrDimer_23119//HIVPrDimer_23119.a3m"
```

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

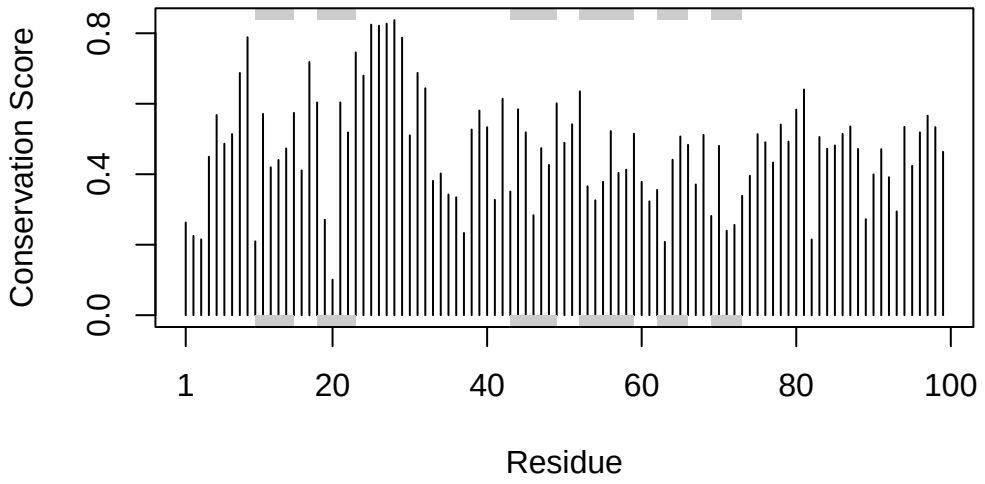
```
[1] " ** Duplicated sequence id's: 101 **"
[2] " ** Duplicated sequence id's: 101 **"
```

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

```
[1] 5378  132
```

```
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] "-" "-" "-" "-" "-" "-"
```