Class 11: AlphaFold

Rachel Lamm (A18518313)

Table of contents

Background

We saw last day that the main respitory for biomolecular structure (the PDB database) only has ~250,000 entries

UniProtKB (the main protein sequence database) has over 200 million entries!

Alphafold

In this hands-on session we will utilize AlphaFold to predict protein structure from sequence (Jumper et al. 2021).

Without the aid of such approaches, it can take years of expensive laboratory work to determine the structure of just one protein. With AlphaFold we can now accurately compute a typical protein structure in as little as ten minutes.

The EBI AlphaFold databse

The EBI alphafold database contains lots of computed structure models increasingly likely that the structure you are interested in is already in this database at < https://alphafold.ebi.ac.uk

There are 3 major outputs from AlphaFold

1. A model of structure in **PDB** format
2. a **pLDDT score**: that tells us how confident the model is for a given residue in your protein (High values are good, above 70)
3. a **PAE score** that tells us about protein packing quality

If you can't find a matching entry for the sequence you are interested in AFDB you can run AlphaFold yourself...

Running AlphaFold

We will use ColabFold to run AlphaFold on our sequence

Figure from AlphaFold here!

Interpreting results

Custom analysis of resulting models

We can read all the AlphaFold results into R and do more quantitative analysis than just viewing the structures in Mol-star:

Read all the PDB MODELS:

```
class11_dir <- "hipvrdimer_94b5b/"
```

```
pdb_files <- list.files(path=class11_dir,
                        pattern="*.pdb",
                        full.names = TRUE)

basename(pdb_files)
```

```
[1] "hipvrdimer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model_4_seed_000.pdb"
[2] "hipvrdimer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model_5_seed_000.pdb"
[3] "hipvrdimer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model_3_seed_000.pdb"
[4] "hipvrdimer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.pdb"
[5] "hipvrdimer_94b5b_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:
hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model_4_seed_000.pdb
hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model_5_seed_000.pdb
hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model_3_seed_000.pdb
hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model_1_seed_000.pdb
hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb
.....

Extracting sequences

pdb/seq: 1   name: hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model
pdb/seq: 2   name: hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model
pdb/seq: 3   name: hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model
pdb/seq: 4   name: hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model
pdb/seq: 5   name: hipvrdimer_94b5b//hipvrdimer_94b5b_unrelaxed_rank_005_alphafold2_ptm_model
```

```
pdbs
```

```
                                 1        .         .         .         .         50
[Truncated_Name:1]hipvrdimer     PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]hipvrdimer     PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:3]hipvrdimer     PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:4]hipvrdimer     PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:5]hipvrdimer     PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                                 **************************************************
                                 1        .         .         .         .         50

                                51        .         .         .         .         99
[Truncated_Name:1]hipvrdimer     GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]hipvrdimer     GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]hipvrdimer     GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]hipvrdimer     GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]hipvrdimer     GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
                                 **************************************************
                                51        .         .         .         .         99
```

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

Class:
  pdbs, fasta

Alignment dimensions:
  5 sequence rows; 99 position columns (99 non-gap, 0 gap)

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

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

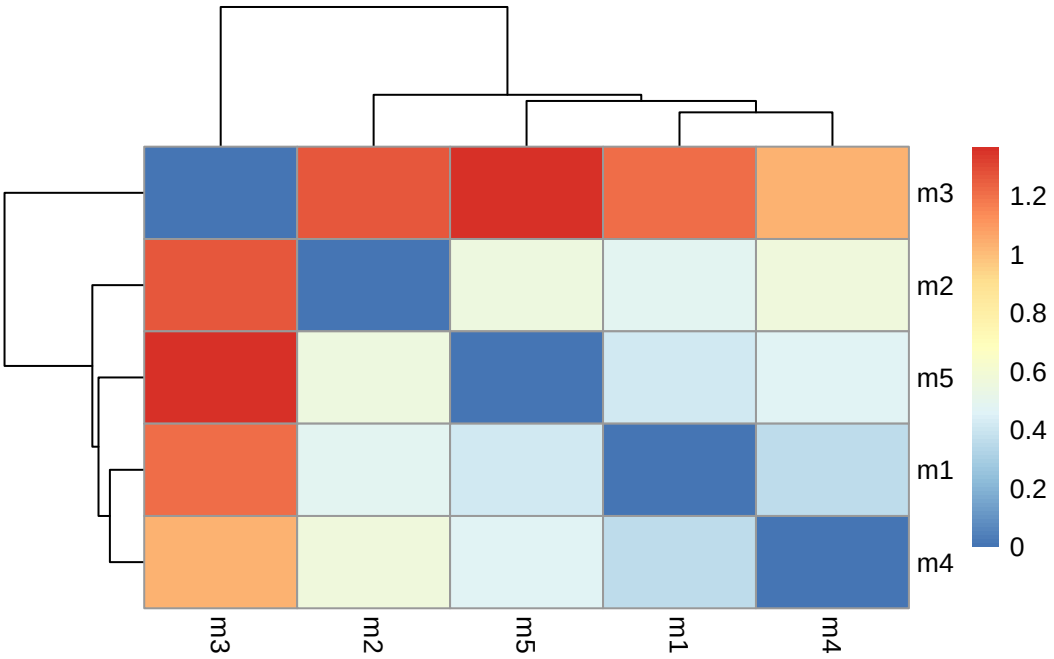
```
Warning in rmsd(pdbs, fit = T): No indices provided, using the 99 non NA positions
```

```
range(rd)
```

```
[1] 0.000 1.367
```

```
library(pheatmap)

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



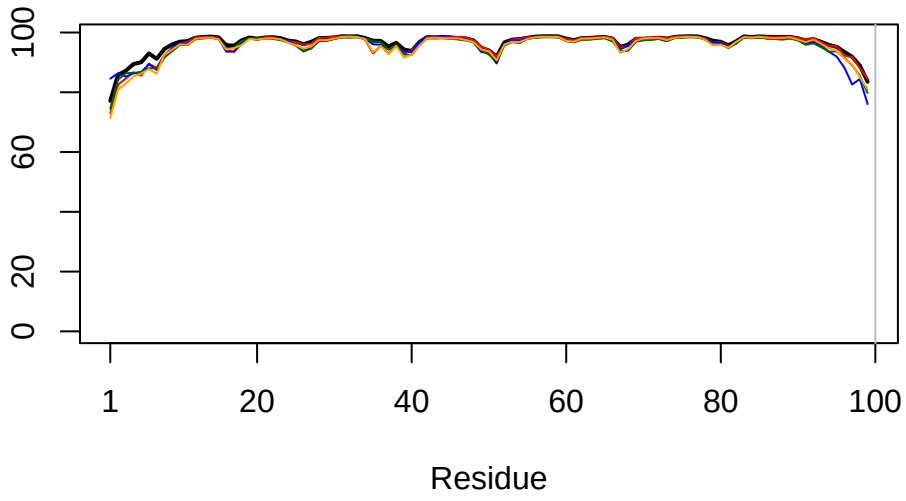

```
pdb <- read.pdb("1hsg")
```

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

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

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

```
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 98 of 99  vol = 4.608
 core size 97 of 99  vol = 3.729
 core size 96 of 99  vol = 2.981
 core size 95 of 99  vol = 2.312
 core size 94 of 99  vol = 1.831
 core size 93 of 99  vol = 1.359
 core size 92 of 99  vol = 0.992
 core size 91 of 99  vol = 0.606
 core size 90 of 99  vol = 0.38
 FINISHED: Min vol ( 0.5 ) reached
```

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

```
# 91 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1     3   3      1
2     7  96     90
```

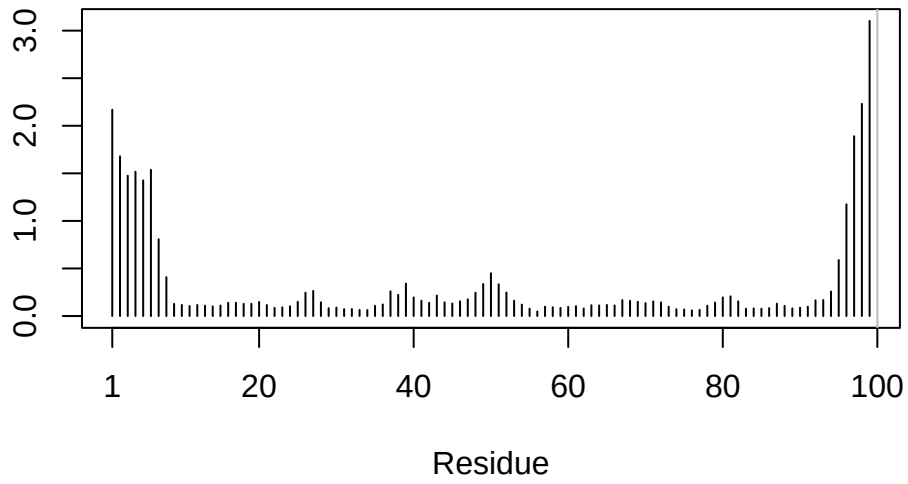
```
xyz <- pdbfit(pdbs, core.inds, outpath="corefit_structures")
```

```
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")
```



Predicted Alignment Error for domains

```
library(jsonlite)

# Listing of all PAE JSON files
pae_files <- list.files(path=class11_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"
```

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

```
[1] 77.12 85.81 87.19 89.44 90.06 93.00
```

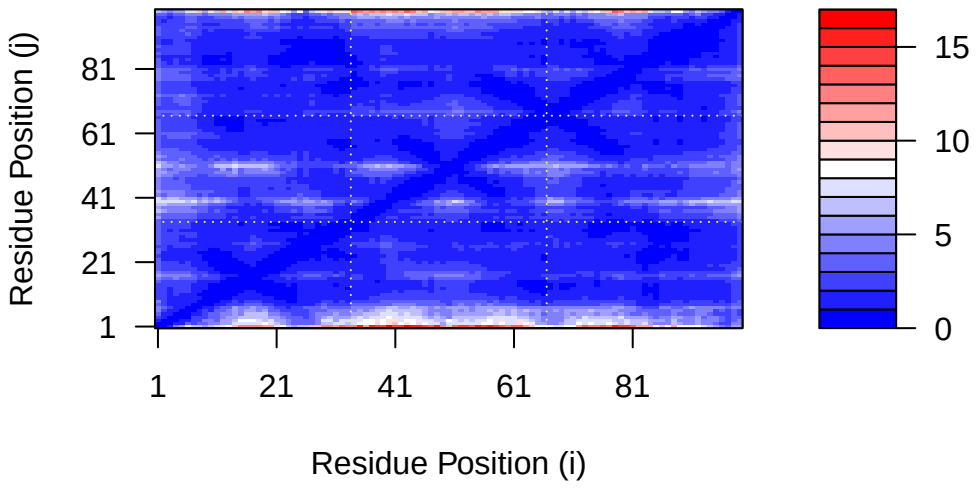
```
pae1$max_pae
```

```
[1] 16.96875
```

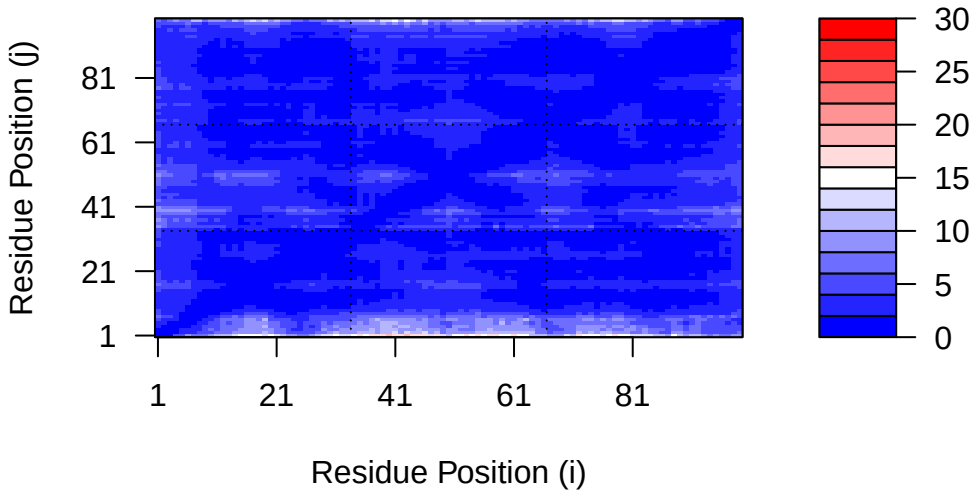
```
pae5$max_pae
```

```
[1] 19.3125
```

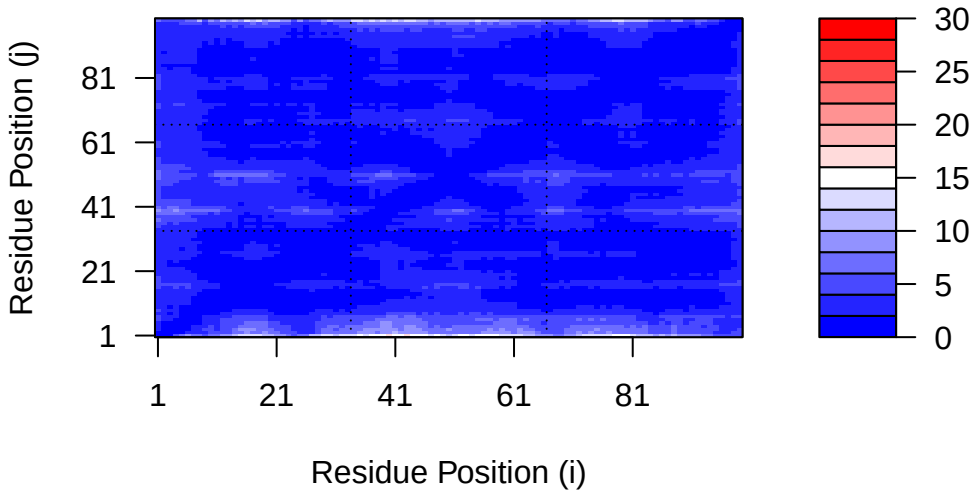
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```




```
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=class11_dir,
                       pattern=".a3m$",
                        full.names = TRUE)
aln_file
```

```
[1] "hipvrdimer_94b5b//hipvrdimer_94b5b.a3m"
```

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

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

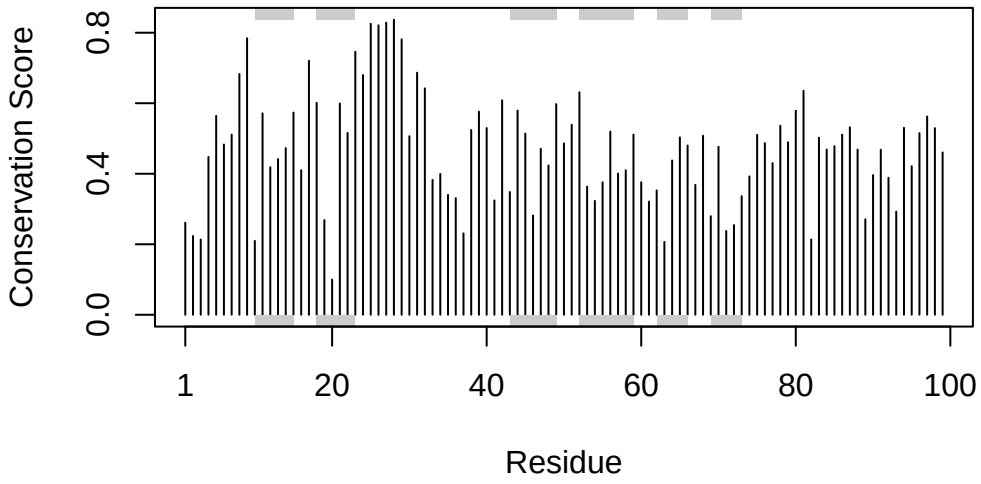
How many sequences are in this alignment

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

```
[1] 5397  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] "-" "-" "-" "-" "-" "-"
```