class11

Eli Sobel A69027989

Today, before delving into structure prediction with AlphaFold we will finish off our previous lab10 "comparative structure analysis" section.

```r
library(bio3d)

id <- "1ake_A"

aa <- get.seq(id)
```

```
Warning in get.seq(id): Removing existing file: seqs.fasta
```

```
Fetching... Please wait. Done.
```

```r
aa
```

```
             1        .         .         .         .         .          60
pdb|1AKE|A    MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
             1        .         .         .         .         .          60

             61       .         .         .         .         .         120
pdb|1AKE|A    DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
             61       .         .         .         .         .         120

             121      .         .         .         .         .         180
pdb|1AKE|A    VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
             121      .         .         .         .         .         180

             181      .         .         .   214
pdb|1AKE|A    YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG
             181      .         .         .   214
```

```
Call:
  read.fasta(file = outfile)

Class:
  fasta

Alignment dimensions:
  1 sequence rows; 214 position columns (214 non-gap, 0 gap)

+ attr: id, ali, call
```

```
b <- blast.pdb(aa)
```

```
 Searching ... please wait (updates every 5 seconds) RID = JS1FFSC6016
 ...
 Reporting 85 hits
```

```
attributes(b)
```

```
$names
[1] "hit.tbl" "raw"     "url"

$class
[1] "blast"
```

```
head(b$hit.tbl)
```

```
        queryid subjectids identity alignmentlength mismatches gapopens q.start
1 Query_1967435     1AKE_A  100.000             214          0        0       1
2 Query_1967435     8BQF_A   99.533             214          1        0       1
3 Query_1967435     4X8M_A   99.533             214          1        0       1
4 Query_1967435     6S36_A   99.533             214          1        0       1
5 Query_1967435     8Q2B_A   99.533             214          1        0       1
6 Query_1967435     8RJ9_A   99.533             214          1        0       1
  q.end s.start s.end    evalue bitscore positives mlog.evalue pdb.id    acc
1   214       1   214 1.58e-156      432    100.00    358.7458 1AKE_A 1AKE_A
2   214      21   234 2.58e-156      433    100.00    358.2555 8BQF_A 8BQF_A
3   214       1   214 2.82e-156      432    100.00    358.1665 4X8M_A 4X8M_A
4   214       1   214 4.14e-156      432    100.00    357.7826 6S36_A 6S36_A
5   214       1   214 1.10e-155      431     99.53    356.8054 8Q2B_A 8Q2B_A
6   214       1   214 1.10e-155      431     99.53    356.8054 8RJ9_A 8RJ9_A
```

```
hits <- plot(b)
```

```
* Possible cutoff values:    197 11
         Yielding Nhits:    19 85

* Chosen cutoff value of:    197
         Yielding Nhits:    19
```
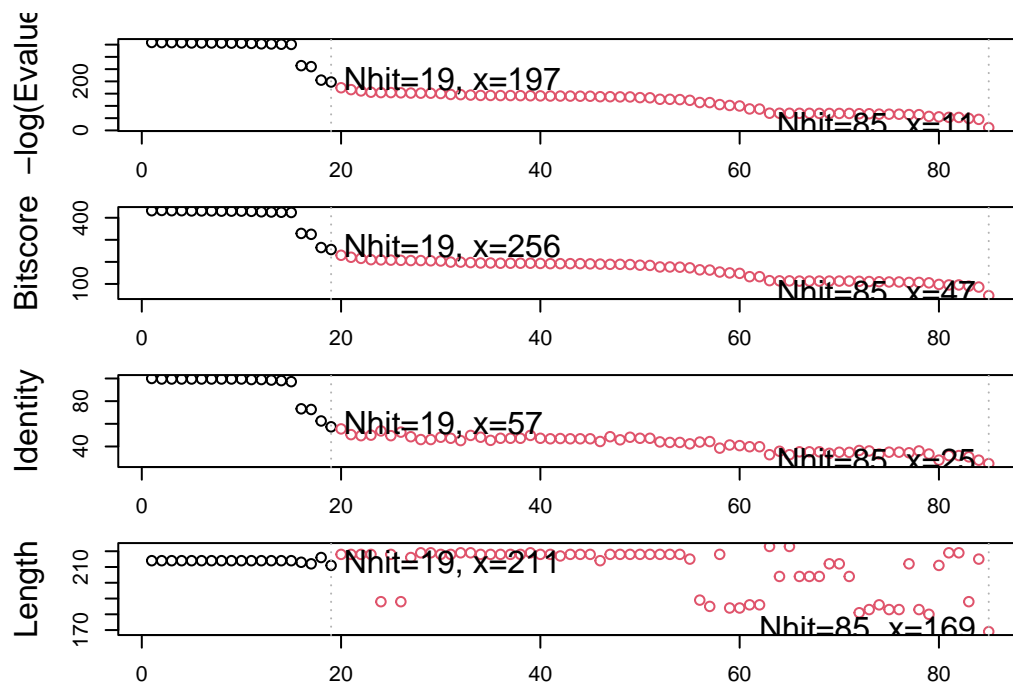


```
attributes(hits)
```

```
$names
[1] "hits"   "pdb.id" "acc"    "inds"

$class
[1] "blast"
```

Top hits that we like from our blast results:

```
hits$pdb.id
```

```
 [1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A" "6RZE_A" "4X8H_A"
 [9] "3HPR_A" "1E4V_A" "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "4K46_A"
[17] "4NP6_A" "3GMT_A" "4PZL_A"
```

find and download all related structures

```
# donwload related PDB files
files <- get.pdb(hits$pdb.id, path="pdbs", split=TRUE, gzip=TRUE)
```

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1AKE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8BQF.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8M.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6S36.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8Q2B.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8RJ9.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6RZE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8H.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3HPR.pdb exists. Skipping download

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4V.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/5EJE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4Y.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3X2S.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAP.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAM.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4K46.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4NP6.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3GMT.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4PZL.pdb exists. Skipping download


  |
  |                                                                      |   0%
  |
  |====                                                                  |   5%
  |
  |=======                                                               |  11%
  |
  |==========                                                            |  16%
```
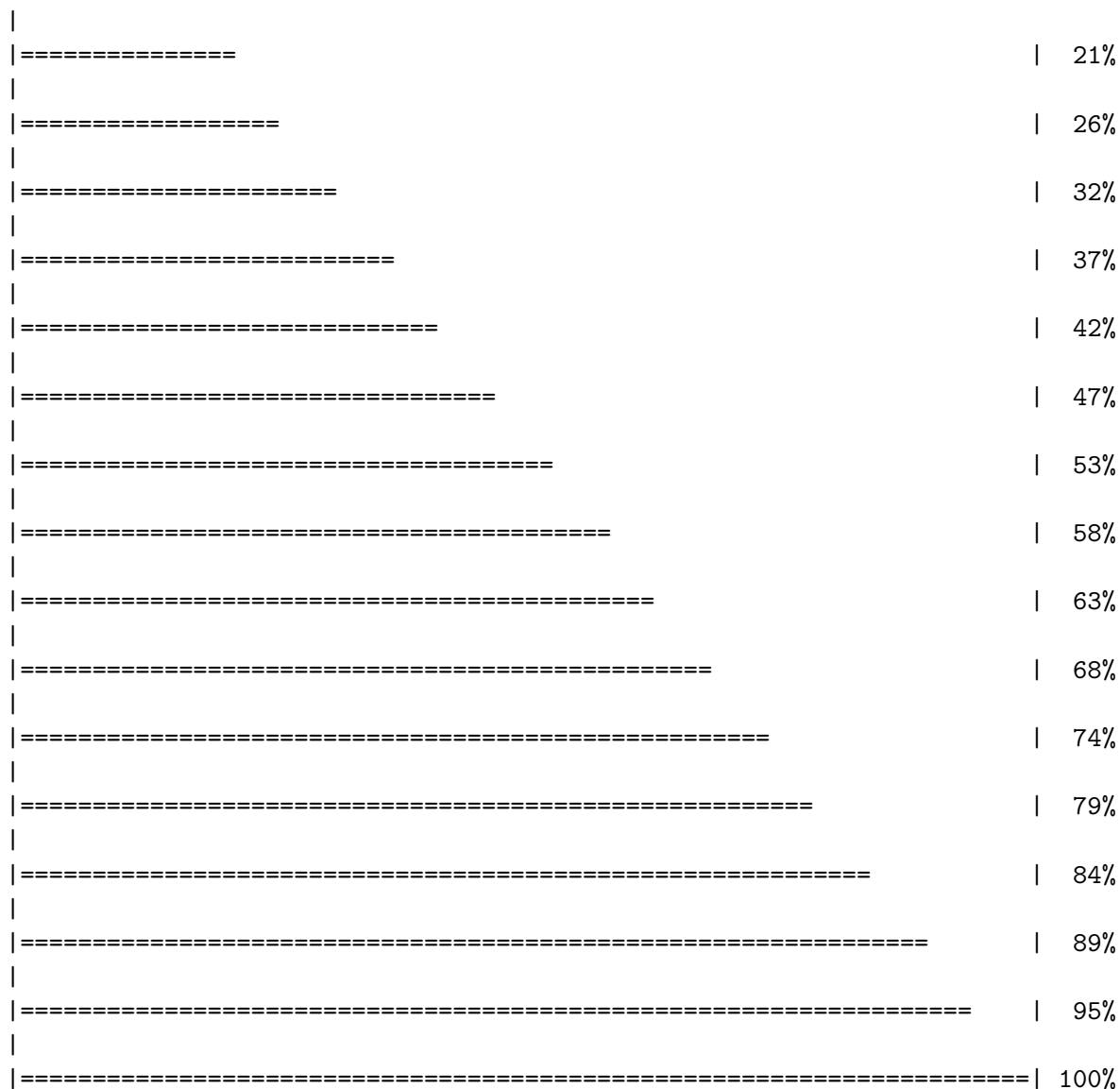
```
  |
  |===============                                                       |   21%
  |
  |=================                                                     |   26%
  |
  |====================                                                  |   32%
  |
  |=======================                                               |   37%
  |
  |==========================                                            |   42%
  |
  |==============================                                        |   47%
  |
  |=================================                                     |   53%
  |
  |====================================                                  |   58%
  |
  |========================================                              |   63%
  |
  |===========================================                           |   68%
  |
  |===============================================                       |   74%
  |
  |==================================================                    |   79%
  |
  |======================================================                |   84%
  |
  |=========================================================             |   89%
  |
  |=============================================================         |   95%
  |
  |======================================================================|  100%
```

I have now found and downloaded all ADK structures in the PDB database but viewing them is difficult as they need to be aligned and superposed.

I am going to install BiocManager package from CRAN. Then I can use "BiocManager::install("msa")".

```r
# Align releated PDBs
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
Reading PDB files:
```

```
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/8BQF_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/8Q2B_A.pdb
pdbs/split_chain/8RJ9_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/4X8H_A.pdb
pdbs/split_chain/3HPR_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/5EJE_A.pdb
pdbs/split_chain/1E4Y_A.pdb
pdbs/split_chain/3X2S_A.pdb
pdbs/split_chain/6HAP_A.pdb
pdbs/split_chain/6HAM_A.pdb
pdbs/split_chain/4K46_A.pdb
pdbs/split_chain/4NP6_A.pdb
pdbs/split_chain/3GMT_A.pdb
pdbs/split_chain/4PZL_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..    PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..    PDB has ALT records, taking A only, rm.alt=TRUE
..    PDB has ALT records, taking A only, rm.alt=TRUE
....    PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
....


Extracting sequences

pdb/seq: 1   name: pdbs/split_chain/1AKE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 2   name: pdbs/split_chain/8BQF_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 3   name: pdbs/split_chain/4X8M_A.pdb
pdb/seq: 4   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/8Q2B_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 6   name: pdbs/split_chain/8RJ9_A.pdb
```

```
    PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 7   name: pdbs/split_chain/6RZE_A.pdb
    PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 8   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 9   name: pdbs/split_chain/3HPR_A.pdb
    PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 10   name: pdbs/split_chain/1E4V_A.pdb
pdb/seq: 11   name: pdbs/split_chain/5EJE_A.pdb
    PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 12   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 13   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 14   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 15   name: pdbs/split_chain/6HAM_A.pdb
    PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 16   name: pdbs/split_chain/4K46_A.pdb
    PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 17   name: pdbs/split_chain/4NP6_A.pdb
pdb/seq: 18   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: 19   name: pdbs/split_chain/4PZL_A.pdb
```

Principal Component analysis

```
pc <- pca(pdbs)
plot(pc)
```
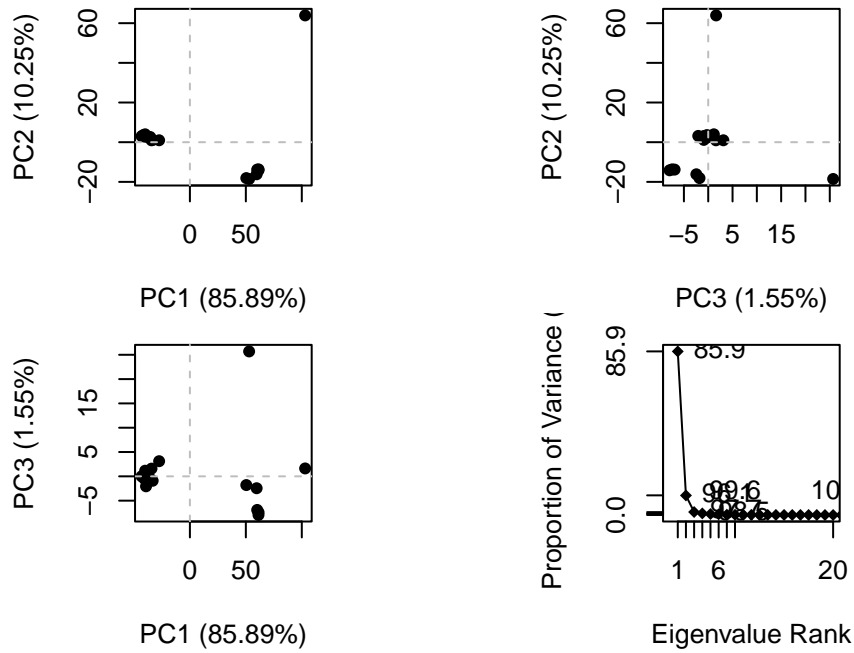
```
plot(pc, pc.axes = c(1:2))
```
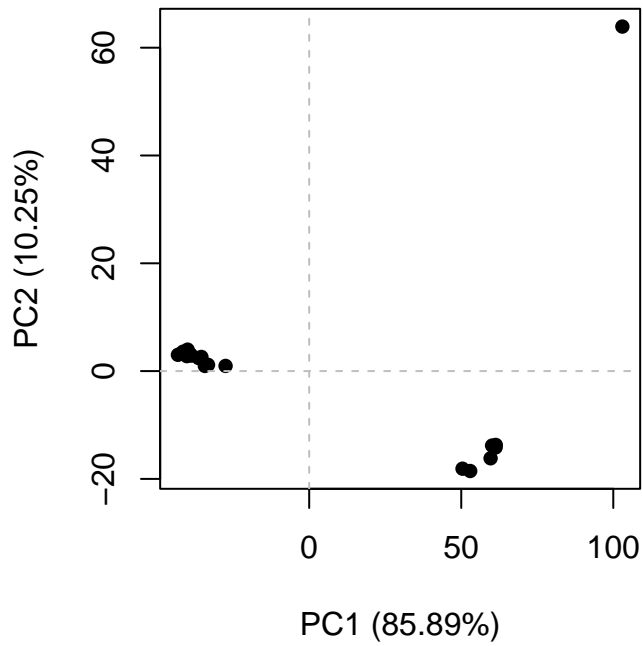

To examine in more detail what PC1 (or any PC) is capturing here we an plot the loadings or make a movie (trajectory) of moving along PC1

9

```
mktrj(pc, pc=1, file="pc1.pdb")
```