Class 9 - PDB

Mirte Ciz Marieke Kuijpers

16/02/2022

Introduction

Analysis and investigation of PDB database.

After downloading data from PDB (https://www.rcsb.org/stats/summary), read it into R.

```
# Read .csv data
dat <- read.csv("Data Export Summary.csv", header = T, row.names = 1)
```

Question 1

```
# Calculate number of structures solved by X ray or EM
X.ray.and.EM <- sum(dat$X.ray) + sum(dat$EM)
X.ray.and.EM
```

```
## [1] 173240
```

```
# Find the percentage this makes up
perc <- (X.ray.and.EM/sum(dat$Total))*100

print(paste("In PDB, X ray and EM derived structures account for", signif(per
c, digits = 3), "% of the structures in the database (to 3 s.f.)."))
```

```
## [1] "In PDB, X ray and EM derived structures account for 92.6 % of the str
uctures in the database (to 3 s.f.)."
```

Another way to do this:

```
# Find the sum of each column (method)
n.type <- colSums(dat)

# Find the percentage of each method (column) of the total
percs <- signif((n.type/n.type["Total"] * 100), digits = 3)

percs
```

```
##            X.ray              NMR               EM Multiple.methods
##          87.2000           7.2800           5.3900           0.1060
##          Neutron            Other            Total
##           0.0385           0.0198         100.0000
```

The proportion or percentage of X-ray structures is 87.2% of the total structures in PDB (at the time of summary statistics download).

The proportion or percentage of EM structures is 5.39% of the total structures in PDB (at the time of summary statistics download).

Question 2

```
# Find number of protein only structures
p.only <- dat[1, "Total"]

# If one didn't know the row that was protein only, they could do the followi
ng
p.only.2 <- dat["Protein (only)", "Total"]
p.only.2

## [1] 163330

# Print answer
print(paste(p.only, "structures in the database are protein only structures,
this is a ratio of 1:", signif(sum(dat$Total)/p.only, digits = 3)))

## [1] "163330 structures in the database are protein only structures, this i
s a ratio of 1: 1.15"

# Or assign it to a variable that can be called in the text
q2 <- signif(p.only/sum(dat$Total), digits = 3)*100
```

The percentage of structures in PBD that are protein only is 87.3% (at time of download of summary statistics).

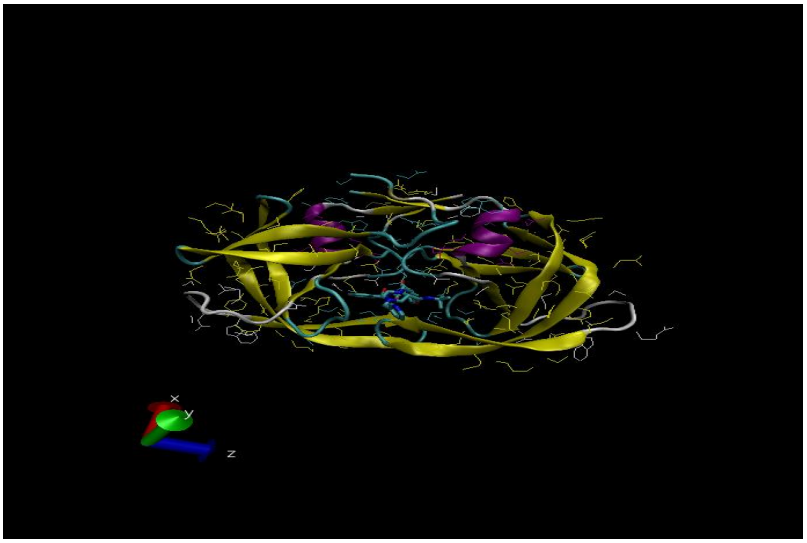
Question 3

Question 3 requires no code, just a PDB search. The search “HIV” and “protease” restricted to all HIV-1 variants under SCIENTIFIC NAME OF SOURCE ORGANISM gives 874 structures. However, searching with different terms gives different numbers and other structures seem to creep in. A better way to search is by sequence, this can be found in the NCBI

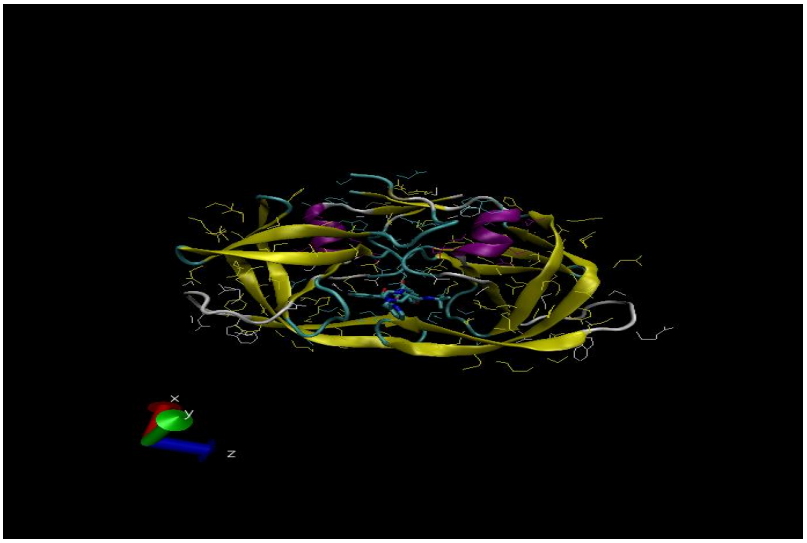
Using a parital HIv-1 protease found on NCBI (accession number: BAA88351.1), gives 860 structures. Using chain A, HIV-1 protease (PDB: 3DOX_A) gives 860 structures too.

VMD image insertion

After creating an image using VMD we can use code of the following format “exclamation-square-brackets-brackets with image name within them” to insert it into the markdown document.



Or you can insert using the visual editor, using no code.



Question 4

While the water was not shown in the image I inserted in my report, in VMD one can display the water. When this is done, only the oxygen molecule is shown. This is because in many imaging techniques, hydrogen, the smallest atom, cannot be visualized.

Question 5

HOH308 found using “within 5 of resname MK1” with MK1 as liquorice and the protein in cartoonNew form.

Question 6

There is a Beta-sheet composed of two beta-chains in one direction from monomer with a third beta-chain in the opposite direction from the other monomer in between them. This would be unlikely to exist if the two monomers were separated. Alpha helices are self-contained and so likely to exist in separate monomers. The remainder of the beta-sheets appear to only rely on H-bonding within the same polypeptide chain and so are probably stable as well. The remainder of the contacts between the monomers in the dimer appear to be in flexible loop regions, which, where they contact, may be hydrophobic and this might skew monomer structure but as the open acitve site is also at the contact surface of the two monomers this should hopefully not be too serious.

Bio3D for strucutral bioinformatics

```
# Load the bio3d package
library(bio3d)

# Use the function to read PDB files from this package, I am using the file I downloaded, but I could also have used the protein tag and the function would have pulled the data from online
pdb <- read.pdb("1hsg.pdb")
pdb

##
##  Call:  read.pdb(file = "1hsg.pdb")
##
##    Total Models#: 1
##      Total Atoms#: 1686,  XYZs#: 5058  Chains#: 2  (values: A B)
##
##      Protein Atoms#: 1514  (residues/Calpha atoms#: 198)
##      Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
##
##      Non-protein/nucleic Atoms#: 172  (residues: 128)
##      Non-protein/nucleic resid values: [ HOH (127), MK1 (1) ]
##
##    Protein sequence:
##       PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYD
##       QILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFPQITLWQRPLVTIKIGGQLKE
##       ALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTP
##       VNIIGRNLLTQIGCTLNF
##
## + attr: atom, xyz, seqres, helix, sheet,
##         calpha, remark, call
```

In the information above MK1 stands for Merk1, a small molecule that was developed to target this protease.

```
# To find the one letter code of three letter amino acids one can use aa321,
as shown below
aa321("GLN")
```

```
## [1] "Q"
```

```
# Within the read in object atom contains the data for each atom in the struc
ture (as seen in the raw data file after scrolling down in a text editor)
head(pdb$atom)
```

```
##   type eleno elety  alt resid chain resno insert      x      y     z o
b
## 1 ATOM     1     N <NA>   PRO     A     1   <NA> 29.361 39.686 5.862 1 38.
10
## 2 ATOM     2    CA <NA>   PRO     A     1   <NA> 30.307 38.663 5.319 1 40.
62
## 3 ATOM     3     C <NA>   PRO     A     1   <NA> 29.760 38.071 4.022 1 42.
64
## 4 ATOM     4     O <NA>   PRO     A     1   <NA> 28.600 38.302 3.676 1 43.
40
## 5 ATOM     5    CB <NA>   PRO     A     1   <NA> 30.508 37.541 6.342 1 37.
87
## 6 ATOM     6    CG <NA>   PRO     A     1   <NA> 29.296 37.591 7.162 1 38.
40
##   segid elesy charge
## 1  <NA>     N   <NA>
## 2  <NA>     C   <NA>
## 3  <NA>     C   <NA>
## 4  <NA>     O   <NA>
## 5  <NA>     C   <NA>
## 6  <NA>     C   <NA>
```

Questions 7 - 9

Q7: There are 198 amino acid residues in this pdb object?

Q8: There are two non-protein residues, water (HOH) and MK1 (a small molecule).

Q9: There are two protein chains in this structure.

Compartive analysis of protein structures

```
# Load necessary packages (after installing in console as necessary)
library("bio3d")
library("ggplot2")
library("ggrepel")
library("devtools")
library("BiocManager")
library("msa")
library("bio3d.view") # N.B. instructions were to use devtools::install_bitbu
cket("Grantlab/bio3d-view"), however, when attempting to use library("Grantla
```

```
b/bio3d-view")
# a second attempt of devtools::install_bitbucket("Grantlab/bio3d-view") retu
rned a warning message that bio3d.view' had already been installed and so thi
s nomenclature was used for loading
```

Questions 10 -12

Q10: The `msa` package is not found in CRAN, only BioConductor, as it has to be installed using `BiocManager::install`.

Q11: The `Grantlab/bio3d-view` is found on neither the CRAN nor the BioConducter database.

Q12: TRUE, functions from the devtools package can be used to install packages from GitHub and BitBucket (using devtools::install_github() and devtools::install_bitbucket() respectively).

Compartive analysis continued

Read a single ADK structure from teh database

```
# Read in the sequence of the POI
adk <- get.seq("1ake_A")

## Warning in get.seq("1ake_A"): Removing existing file: seqs.fasta

## Fetching... Please wait. Done.

# Observe data
adk

##              1        .         .         .         .         .         60
## pdb|1AKE|A   MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
##              1        .         .         .         .         .         60
##
##             61        .         .         .         .         .         12
0
## pdb|1AKE|A   DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
##             61        .         .         .         .         .         12
0
##
##            121        .         .         .         .         .         18
0
## pdb|1AKE|A   VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
##            121        .         .         .         .         .         18
0
##
##            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
```

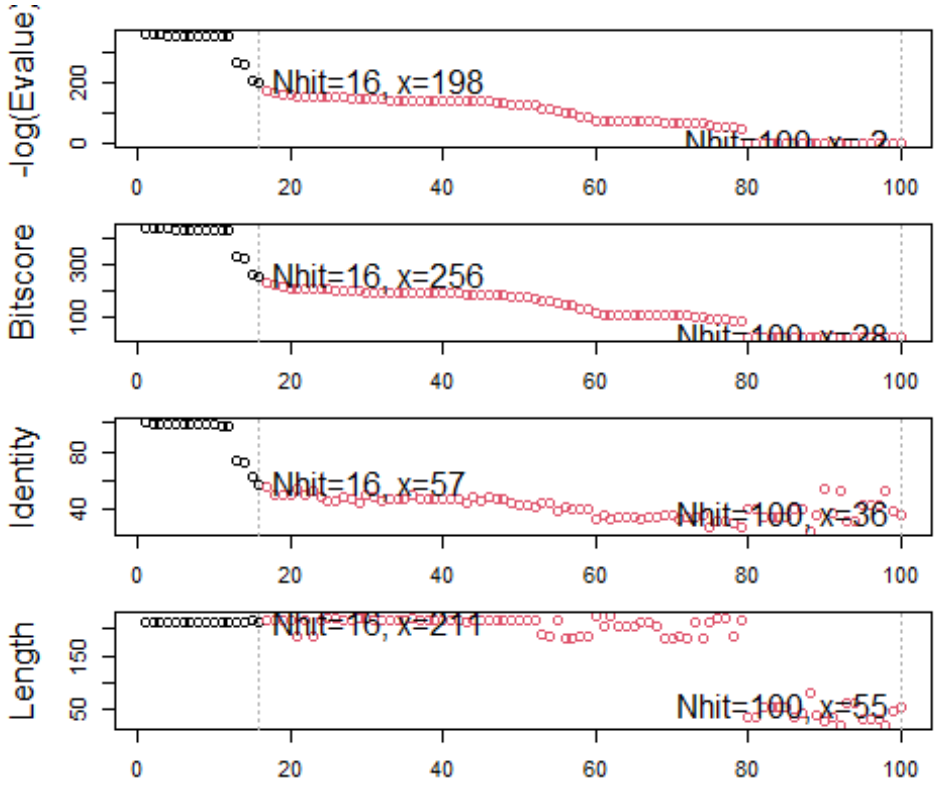
Question 12

The above output reveals there are 214 residues in this protein.

Compartive analysis continued

Now we can find related sequences.

```
# Search for further sequences
blast <- blast.pdb(adk)

##  Searching ... please wait (updates every 5 seconds) RID = 0XMV8KE5013
##  ..............................
##  Reporting 100 hits

# Plot summary statistics of results
hits <- plot(blast)

##   * Possible cutoff values:    197 -3
##             Yielding Nhits:    16 100
##
##   * Chosen cutoff value of:    197
##             Yielding Nhits:    16
```




```
hits

## $hits
##    pdb.id   acc      group
## 1  "1AKE_A" "1AKE_A" "1"
## 2  "4X8M_A" "4X8M_A" "1"
## 3  "6S36_A" "6S36_A" "1"
## 4  "6RZE_A" "6RZE_A" "1"
## 5  "4X8H_A" "4X8H_A" "1"
## 6  "3HPR_A" "3HPR_A" "1"
## 7  "1E4V_A" "1E4V_A" "1"
## 8  "5EJE_A" "5EJE_A" "1"
## 9  "1E4Y_A" "1E4Y_A" "1"
## 10 "3X2S_A" "3X2S_A" "1"
## 11 "6HAP_A" "6HAP_A" "1"
## 12 "6HAM_A" "6HAM_A" "1"
## 13 "4K46_A" "4K46_A" "1"
## 14 "4NP6_A" "4NP6_A" "1"
## 15 "3GMT_A" "3GMT_A" "1"
## 16 "4PZL_A" "4PZL_A" "1"
##
## $pdb.id
##  [1] "1AKE_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A" "3HPR_A" "1E4V_A" "5EJE_
A"
##  [9] "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "4K46_A" "4NP6_A" "3GMT_A" "4PZL_
A"
##
```

```
## $acc
##  [1] "1AKE_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A" "3HPR_A" "1E4V_A" "5EJE_
A"
##  [9] "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "4K46_A" "4NP6_A" "3GMT_A" "4PZL_
A"
##
## $inds
##   [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  T
RUE
##  [13]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE
##  [97] FALSE FALSE FALSE FALSE
##
## attr(,"class")
## [1] "blast"
```

This suggests that we should use the top 16 hits. We can get the IDs of these as follows.

```
# Print the IDs of the hits above the threshold
hits$pdb.id
```

```
##  [1] "1AKE_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A" "3HPR_A" "1E4V_A" "5EJE_
A"
##  [9] "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "4K46_A" "4NP6_A" "3GMT_A" "4PZL_
A"
```

We can now retrieve the protein sequences of these hits.

```
# Download 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/
## 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/
## 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%
|
|====                                                                  |   6%
|
|=========                                                             |  12%
|
|=============                                                         |  19%
|
|==================                                                    |  25%
|
|======================                                                |  31%
|
|==========================                                            |  38%
|
|==============================                                        |  44%
|
|==================================                                    |  50%
|
|======================================                                |  56%
|
|==========================================                            |  62%
|
|==============================================                        |  69%
|
|==================================================                    |  75%
|
|==========================================================            |  81%
|
|==============================================================        |  88%
|
|======================================================================|  94%
|
|======================================================================| 100%
```

After which these files can be aligned.

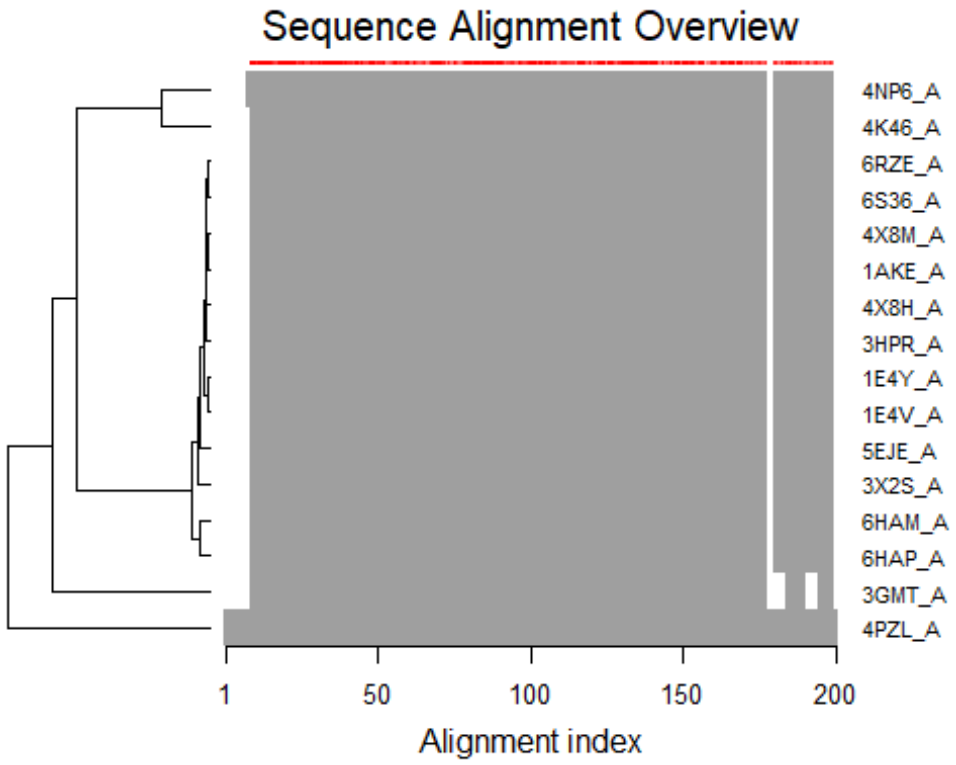
```
# Align related PDBs
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")

## Reading PDB files:
## pdbs/split_chain/1AKE_A.pdb
## pdbs/split_chain/4X8M_A.pdb
## pdbs/split_chain/6S36_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
## ....
##
## 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/4X8M_A.pdb
## pdb/seq: 3   name: pdbs/split_chain/6S36_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 4   name: pdbs/split_chain/6RZE_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 5   name: pdbs/split_chain/4X8H_A.pdb
## pdb/seq: 6   name: pdbs/split_chain/3HPR_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 7   name: pdbs/split_chain/1E4V_A.pdb
## pdb/seq: 8   name: pdbs/split_chain/5EJE_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 9   name: pdbs/split_chain/1E4Y_A.pdb
## pdb/seq: 10   name: pdbs/split_chain/3X2S_A.pdb
## pdb/seq: 11   name: pdbs/split_chain/6HAP_A.pdb
## pdb/seq: 12   name: pdbs/split_chain/6HAM_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 13   name: pdbs/split_chain/4K46_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 14   name: pdbs/split_chain/4NP6_A.pdb
## pdb/seq: 15   name: pdbs/split_chain/3GMT_A.pdb
## pdb/seq: 16   name: pdbs/split_chain/4PZL_A.pdb
```

After extracting the sequences they can be aligned and plotted.

```
# Vector containing PDB codes for figure axis
ids <- basename.pdb(pdbs$id)
```

```
# Draw schematic alignment
plot(pdbs, labels=ids)
```



The plot portrays matched bases as grey and gaps in the sequence as white. Sequence conservation is represented by the red bar at the top of the figure. The sequences are all very similar with the exception of some gaps near the end.

We can also view their structures superimposed:

```
# Set up
library(bio3d.view)
library(rgl)

# Plot
view.pdbs(pdbs)
```

Ideally we should also annotate out sequences.

```
# Annotate
anno <- pdb.annotate(ids)

unique(anno$source)

## [1] "Escherichia coli"
## [2] "Escherichia coli K-12"
## [3] "Escherichia coli O139:H28 str. E24377A"
## [4] "Escherichia coli str. K-12 substr. MDS42"
## [5] "Photobacterium profundum"
```

```
## [6] "Vibrio cholerae O1 biovar El Tor str. N16961"
## [7] "Burkholderia pseudomallei 1710b"
## [8] "Francisella tularensis subsp. tularensis SCHU S4"

anno

##        structureId chainId macromoleculeType chainLength experimentalTechn
ique
## 1AKE_A        1AKE       A           Protein         214                X
-ray
## 4X8M_A        4X8M       A           Protein         214                X
-ray
## 6S36_A        6S36       A           Protein         214                X
-ray
## 6RZE_A        6RZE       A           Protein         214                X
-ray
## 4X8H_A        4X8H       A           Protein         214                X
-ray
## 3HPR_A        3HPR       A           Protein         214                X
-ray
## 1E4V_A        1E4V       A           Protein         214                X
-ray
## 5EJE_A        5EJE       A           Protein         214                X
-ray
## 1E4Y_A        1E4Y       A           Protein         214                X
-ray
## 3X2S_A        3X2S       A           Protein         214                X
-ray
## 6HAP_A        6HAP       A           Protein         214                X
-ray
## 6HAM_A        6HAM       A           Protein         214                X
-ray
## 4K46_A        4K46       A           Protein         214                X
-ray
## 4NP6_A        4NP6       A           Protein         217                X
-ray
## 3GMT_A        3GMT       A           Protein         230                X
-ray
## 4PZL_A        4PZL       A           Protein         242                X
-ray
##        resolution       scopDomain                    pfam         ligandId
## 1AKE_A      2.000 Adenylate kinase Adenylate kinase (ADK)             AP5
## 4X8M_A      2.600             <NA> Adenylate kinase (ADK)            <NA>
## 6S36_A      1.600             <NA> Adenylate kinase (ADK) CL (3),NA,MG (2)
## 6RZE_A      1.690             <NA> Adenylate kinase (ADK)    NA (3),CL (2)
## 4X8H_A      2.500             <NA> Adenylate kinase (ADK)            <NA>
## 3HPR_A      2.000             <NA> Adenylate kinase (ADK)             AP5
## 1E4V_A      1.850 Adenylate kinase Adenylate kinase (ADK)             AP5
## 5EJE_A      1.900             <NA> Adenylate kinase (ADK)          AP5,CO
## 1E4Y_A      1.850 Adenylate kinase Adenylate kinase (ADK)             AP5
```

```
## 3X2S_A      2.800             <NA> Adenylate kinase (ADK)   JPY (2),AP5,MG
## 6HAP_A      2.700             <NA> Adenylate kinase (ADK)              AP5
## 6HAM_A      2.550             <NA> Adenylate kinase (ADK)              AP5
## 4K46_A      2.010             <NA> Adenylate kinase (ADK)      ADP,AMP,PO4
## 4NP6_A      2.004             <NA> Adenylate kinase (ADK)             <NA>
## 3GMT_A      2.100             <NA> Adenylate kinase (ADK)          SO4 (2)
## 4PZL_A      2.100             <NA> Adenylate kinase (ADK)       CA,FMT,GOL
##
ligandName
## 1AKE_A                                                 BIS(ADENOSINE)-5'-P
ENTAPHOSPHATE
## 4X8M_A
<NA>
## 6S36_A                                  CHLORIDE ION (3),SODIUM ION,MAGN
ESIUM ION (2)
## 6RZE_A                                                  SODIUM ION (3),CHL
ORIDE ION (2)
## 4X8H_A
<NA>
## 3HPR_A                                                 BIS(ADENOSINE)-5'-P
ENTAPHOSPHATE
## 1E4V_A                                                 BIS(ADENOSINE)-5'-P
ENTAPHOSPHATE
## 5EJE_A                                BIS(ADENOSINE)-5'-PENTAPHOSPHATE,CO
BALT (II) ION
## 1E4Y_A                                                 BIS(ADENOSINE)-5'-P
ENTAPHOSPHATE
## 3X2S_A N-(pyren-1-ylmethyl)acetamide (2),BIS(ADENOSINE)-5'-PENTAPHOSPHATE,
MAGNESIUM ION
## 6HAP_A                                                 BIS(ADENOSINE)-5'-P
ENTAPHOSPHATE
## 6HAM_A                                                 BIS(ADENOSINE)-5'-P
ENTAPHOSPHATE
## 4K46_A                  ADENOSINE-5'-DIPHOSPHATE,ADENOSINE MONOPHOSPHATE,
PHOSPHATE ION
## 4NP6_A
<NA>
## 3GMT_A                                                                  SU
LFATE ION (2)
## 4PZL_A                                                 CALCIUM ION,FORMIC
ACID,GLYCEROL
##                                                 source
## 1AKE_A                                Escherichia coli
## 4X8M_A                                Escherichia coli
## 6S36_A                                Escherichia coli
## 6RZE_A                                Escherichia coli
## 4X8H_A                                Escherichia coli
## 3HPR_A                             Escherichia coli K-12
## 1E4V_A                                Escherichia coli
## 5EJE_A           Escherichia coli O139:H28 str. E24377A
```

```
## 1E4Y_A                                    Escherichia coli
## 3X2S_A         Escherichia coli str. K-12 substr. MDS42
## 6HAP_A           Escherichia coli O139:H28 str. E24377A
## 6HAM_A                              Escherichia coli K-12
## 4K46_A                           Photobacterium profundum
## 4NP6_A     Vibrio cholerae O1 biovar El Tor str. N16961
## 3GMT_A                   Burkholderia pseudomallei 1710b
## 4PZL_A Francisella tularensis subsp. tularensis SCHU S4
##
structureTitle
## 1AKE_A STRUCTURE OF THE COMPLEX BETWEEN ADENYLATE KINASE FROM ESCHERICHIA
COLI AND THE INHIBITOR AP5A REFINED AT 1.9 ANGSTROMS RESOLUTION: A MODEL FOR
A CATALYTIC TRANSITION STATE
## 4X8M_A
Crystal structure of E. coli Adenylate kinase Y171W mutant
## 6S36_A
Crystal structure of E. coli Adenylate kinase R119K mutant
## 6RZE_A
Crystal structure of E. coli Adenylate kinase R119A mutant
## 4X8H_A
Crystal structure of E. coli Adenylate kinase P177A mutant
## 3HPR_A
Crystal structure of V148G adenylate kinase from E. coli, in complex with Ap5
A
## 1E4V_A
Mutant G10V of adenylate kinase from E. coli, modified in the Gly-loop
## 5EJE_A
Crystal structure of E. coli Adenylate kinase G56C/T163C double mutant in com
plex with Ap5a
## 1E4Y_A
Mutant P9L of adenylate kinase from E. coli, modified in the Gly-loop
## 3X2S_A
Crystal structure of pyrene-conjugated adenylate kinase
## 6HAP_A
Adenylate kinase
## 6HAM_A
Adenylate kinase
## 4K46_A
Crystal Structure of Adenylate Kinase from Photobacterium profundum
## 4NP6_A
Crystal Structure of Adenylate Kinase from Vibrio cholerae O1 biovar eltor
## 3GMT_A
Crystal structure of adenylate kinase from burkholderia pseudomallei
## 4PZL_A
The crystal structure of adenylate kinase from Francisella tularensis subsp.
tularensis SCHU S4
##                                                        citation rObserved
rFree
## 1AKE_A                 Muller, C.W., et al. J Mol Biol (1992)   0.19600
NA
```

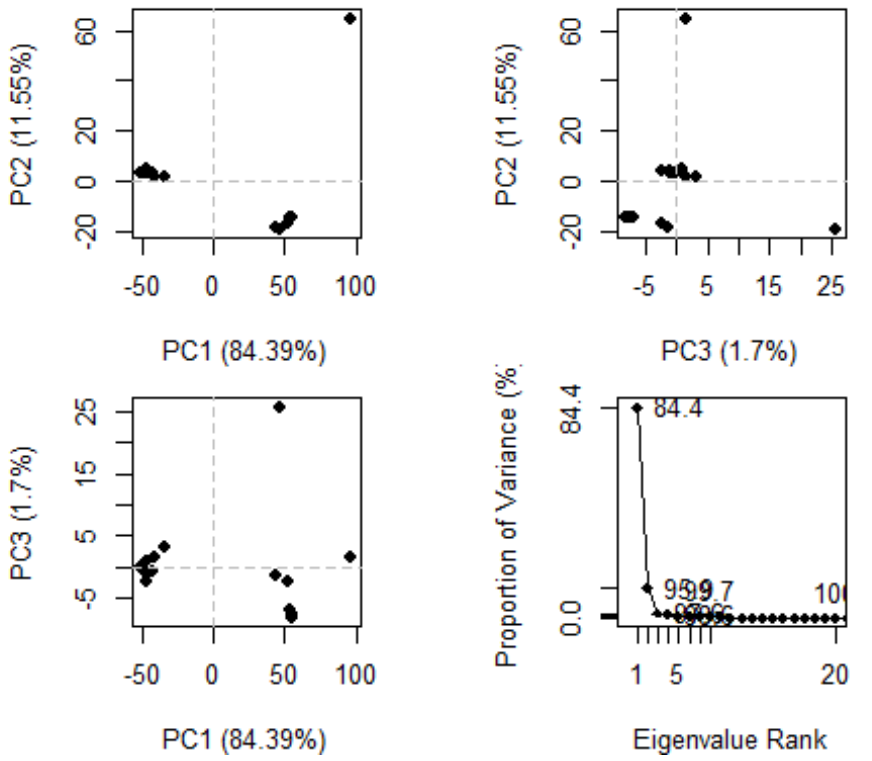
```
## 4X8M_A                Kovermann, M., et al. Nat Commun (2015)   0.24910 0.
30890
## 6S36_A                  Rogne, P., et al. Biochemistry (2019)   0.16320 0.
23560
## 6RZE_A                  Rogne, P., et al. Biochemistry (2019)   0.18650 0.
23500
## 4X8H_A                Kovermann, M., et al. Nat Commun (2015)   0.19610 0.
28950
## 3HPR_A  Schrank, T.P., et al. Proc Natl Acad Sci U S A (2009)   0.21000 0.
24320
## 1E4V_A                   Muller, C.W., et al. Proteins (1993)   0.19600
NA
## 5EJE_A  Kovermann, M., et al. Proc Natl Acad Sci U S A (2017)   0.18890 0.
23580
## 1E4Y_A                   Muller, C.W., et al. Proteins (1993)   0.17800
NA
## 3X2S_A                Fujii, A., et al. Bioconjug Chem (2015)   0.20700 0.
25600
## 6HAP_A               Kantaev, R., et al. J Phys Chem B (2018)   0.22630 0.
27760
## 6HAM_A               Kantaev, R., et al. J Phys Chem B (2018)   0.20511 0.
24325
## 4K46_A                    Cho, Y.-J., et al. To be published    0.17000 0.
22290
## 4NP6_A                       Kim, Y., et al. To be published    0.18800 0.
22200
## 3GMT_A Buchko, G.W., et al. Biochem Biophys Res Commun (2010)   0.23800 0.
29500
## 4PZL_A                       Tan, K., et al. To be published    0.19360 0.
23680
##          rWork spaceGroup
## 1AKE_A 0.19600  P 21 2 21
## 4X8M_A 0.24630    C 1 2 1
## 6S36_A 0.15940    C 1 2 1
## 6RZE_A 0.18190    C 1 2 1
## 4X8H_A 0.19140    C 1 2 1
## 3HPR_A 0.20620  P 21 21 2
## 1E4V_A 0.19600  P 21 2 21
## 5EJE_A 0.18630  P 21 2 21
## 1E4Y_A 0.17800   P 1 21 1
## 3X2S_A 0.20700 P 21 21 21
## 6HAP_A 0.22370    I 2 2 2
## 6HAM_A 0.20311       P 43
## 4K46_A 0.16730 P 21 21 21
## 4NP6_A 0.18600       P 43
## 3GMT_A 0.23500   P 1 21 1
## 4PZL_A 0.19130       P 32
```

THis provides us with a list of the hits and some details about them that are useful to store for further reference.

PCA

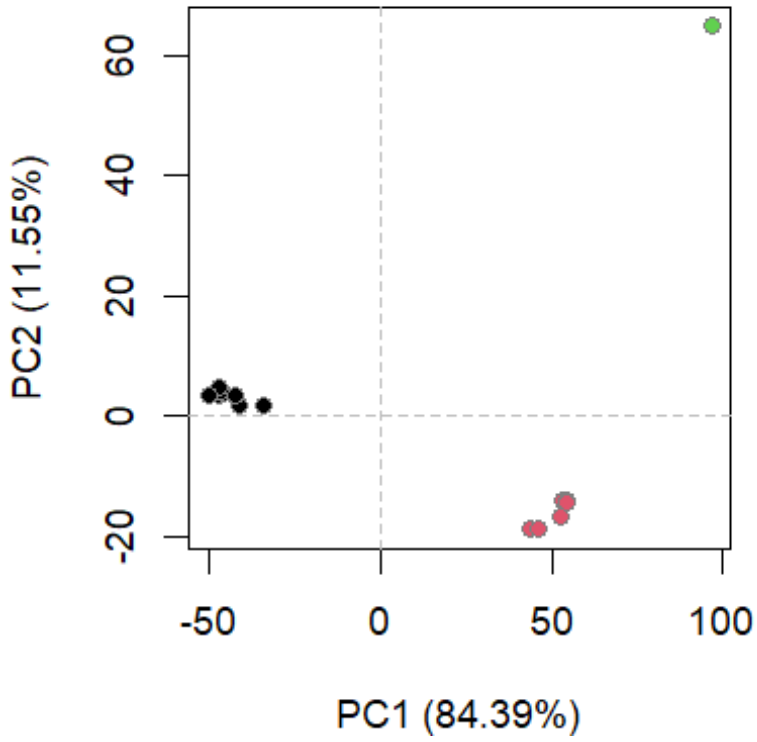
Finally, we can perform principle component analysis.

```
# Perform PCA
pc.xray <- pca(pdbs)
plot(pc.xray)
```

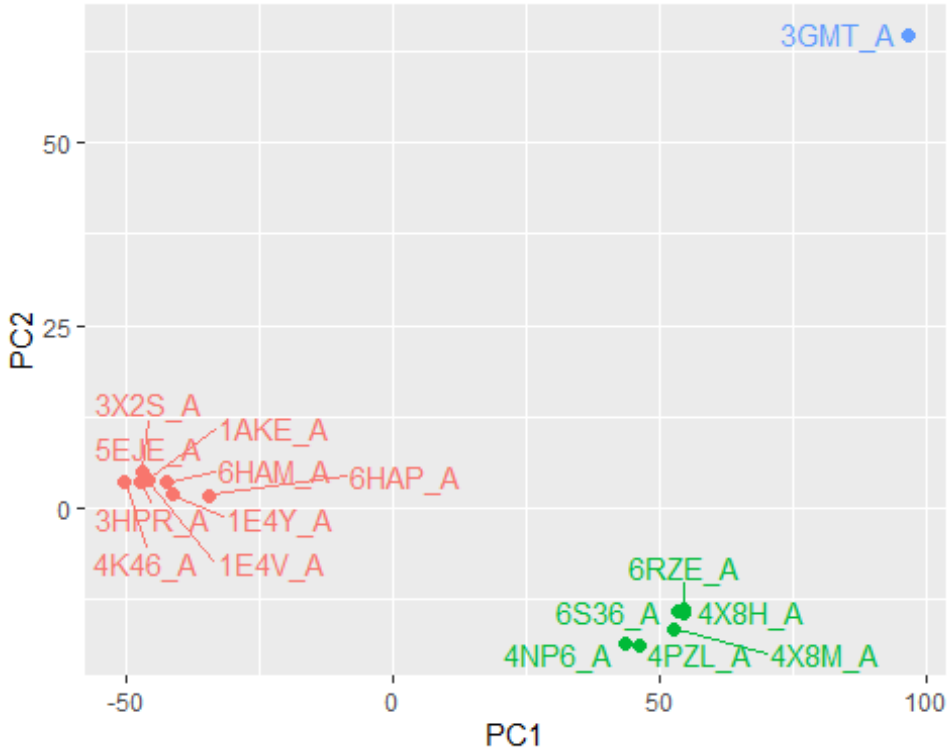


The graphs provide a snapshot of where the adenylate kinases vary most in structure. This can be used for clustering into more similar structures.

```
# Calculate RMSD
rd <- rmsd(pdbs)

## Warning in rmsd(pdbs): No indices provided, using the 204 non NA positions

# Structure-based clustering
hc.rd <- hclust(dist(rd))
grps.rd <- cutree(hc.rd, k=3)


# Plotting
plot(pc.xray, 1:2, col="grey50", bg=grps.rd, pch=21, cex=1)
```




```
#Plotting results with ggplot2
library(ggplot2)
library(ggrepel)

df <- data.frame(PC1=pc.xray$z[,1],
                 PC2=pc.xray$z[,2],
                 col=as.factor(grps.rd),
                 ids=ids)

p <- ggplot(df) +
  aes(PC1, PC2, col=col, label=ids) +
  geom_point(size=2) +
  geom_text_repel(max.overlaps = 20) +
  theme(legend.position = "none")
p
```



There are clearly three groups of proteins which cluster separately.

To visualuse these differences we can return to the structures.

```
# Visualize first principal component
pc1 <- mktrj(pc.xray, pc=1, file="pc_1.pdb")

view.xyz(pc1)
```

```
## Potential all C-alpha atom structure(s) detected: Using calpha.connectivit
y()
```

```
# Set colours to mirror variability
view.xyz(pc1, col=vec2color( rmsf(pc1) ))
```

```
## Potential all C-alpha atom structure(s) detected: Using calpha.connectivit
y()
```

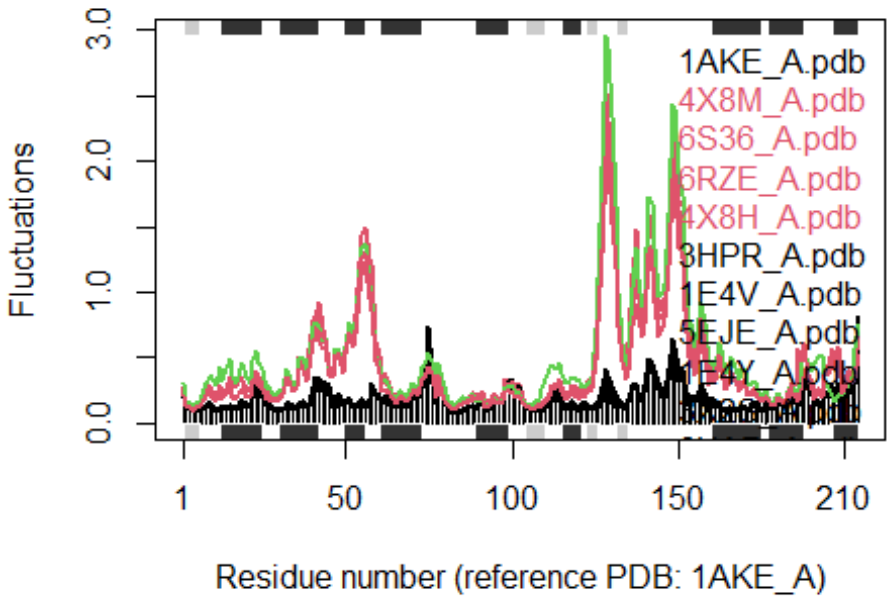
Finally, we can plot the variability in 2D.

```
# NMA of all structures
modes <- nma(pdbs)

##
## Details of Scheduled Calculation:
##    ... 16 input structures
##    ... storing 606 eigenvectors for each structure
##    ... dimension of x$U.subspace: ( 612x606x16 )
##    ... coordinate superposition prior to NM calculation
```

```
##   ... aligned eigenvectors (gap containing positions removed)
##   ... estimated memory usage of final 'eNMA' object: 45.4 Mb
##
##   |
|                                                                      |   0%
|
|====                                                                  |   6%
|
|=========                                                             |  12%
|
|=============                                                         |  19%
|
|==================                                                    |  25%
|
|======================                                                |  31%
|
|==========================                                            |  38%
|
|==============================                                        |  44%
|
|==================================                                    |  50%
|
|======================================                                |  56%
|
|==========================================                            |  62%
|
|==============================================                        |  69%
|
|==================================================                    |  75%
|
|==========================================================            |  81%
|
|==============================================================        |  88%
|
|==================================================================    |  94%
|
|======================================================================| 100%
```

```
# Plot
plot(modes, pdbs, col=grps.rd)
```

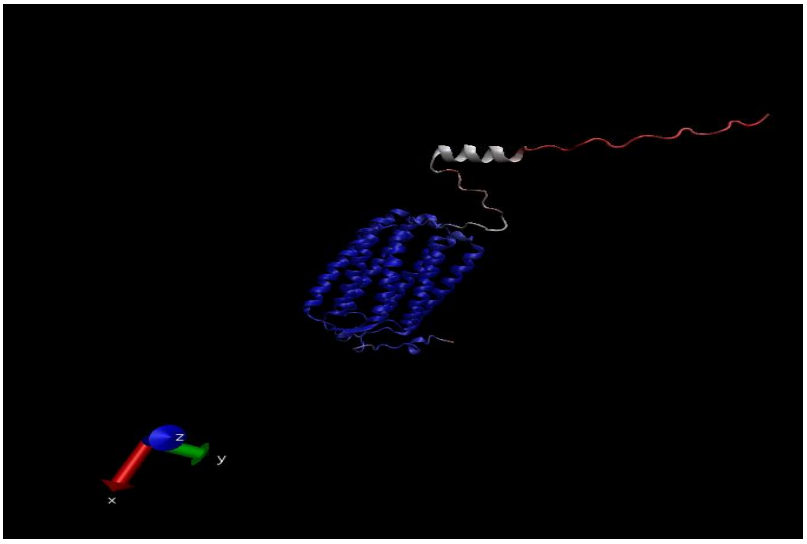
```
## Extracting SSE from pdbs$sse attribute
```



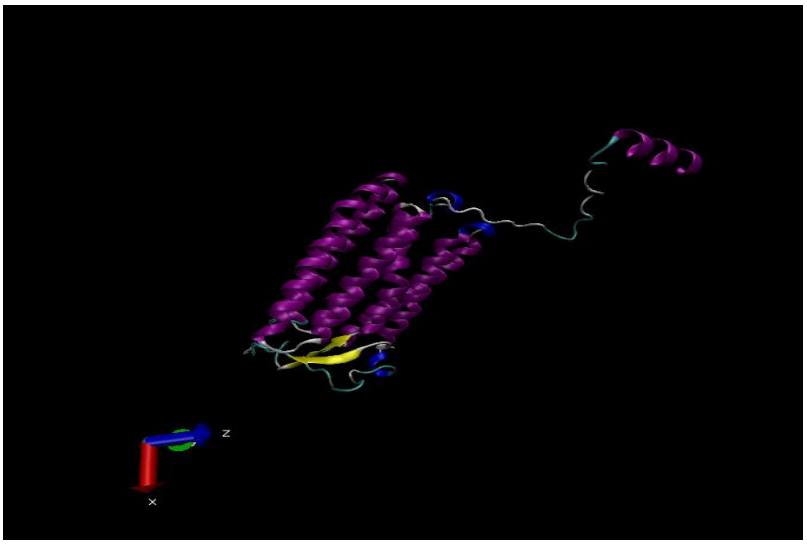
Structures are divided mainly into a red and a black group which vary in their fluctuations. A single protein belonging to the third 'group' is in green and most similar to the red group. Based on the hinging motion shown in the VMD visualization it is possible that these two main groups represent different formations (e.g. substrate bound and unbound) or two classes of kinases with relatively different sized substrates, requiring the greater hinging of the group portrayed in red above.

AlphaFold

Using the predicted ORF sequence of my Find A Gene Project gene I find an 53.5% identity top hit in the AlphaFold database. Below is an image of this top hit predicted structure of an uncharacteristic protein as created in VMD.



Above is a picture with prediction score colouring (red = low prediction score, red, high prediction score), below is the same protein structure, but with any score below 50 removed, and coloured by secondary structure.



Session Information

```
# Record data on the session
sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
```

```
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] rgl_0.108.3           bio3d.view_0.1.0.9000 msa_1.26.0
##  [4] Biostrings_2.62.0     GenomeInfoDb_1.30.1   XVector_0.34.0
##  [7] IRanges_2.28.0        S4Vectors_0.32.3      BiocGenerics_0.40.0
## [10] BiocManager_1.30.16   devtools_2.4.3        usethis_2.1.5
## [13] ggrepel_0.9.1         ggplot2_3.3.5         bio3d_2.4-3.9000
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8             prettyunits_1.1.1      ps_1.6.0
##  [4] rprojroot_2.0.2        digest_0.6.29          utf8_1.2.2
##  [7] R6_2.5.1               evaluate_0.14          highr_0.9
## [10] httr_1.4.2             pillar_1.7.0           zlibbioc_1.40.0
## [13] rlang_1.0.1            curl_4.3.2             rstudioapi_0.13
## [16] callr_3.7.0            rmarkdown_2.11         labeling_0.4.2
## [19] desc_1.4.0             stringr_1.4.0          htmlwidgets_1.5.4
## [22] RCurl_1.98-1.6         munsell_0.5.0          compiler_4.1.2
## [25] xfun_0.29              pkgconfig_2.0.3        pkgbuild_1.3.1
## [28] htmltools_0.5.2        tidyselect_1.1.1       GenomeInfoDbData_1.2.7
## [31] tibble_3.1.6           fansi_1.0.2            crayon_1.5.0
## [34] dplyr_1.0.8            withr_2.4.3            bitops_1.0-7
## [37] brio_1.1.3             grid_4.1.2             jsonlite_1.7.3
## [40] gtable_0.3.0           lifecycle_1.0.1        magrittr_2.0.2
## [43] scales_1.1.1           cli_3.2.0              stringi_1.7.6
## [46] cachem_1.0.6           farver_2.1.0           fs_1.5.2
## [49] remotes_2.4.2          testthat_3.1.2         ellipsis_0.3.2
## [52] generics_0.1.2         vctrs_0.3.8            tools_4.1.2
## [55] glue_1.6.1             purrr_0.3.4            processx_3.5.2
## [58] pkgload_1.2.4          parallel_4.1.2         fastmap_1.1.0
## [61] yaml_2.2.2             colorspace_2.0-2       sessioninfo_1.2.2
## [64] memoise_2.0.1          knitr_1.37
```