# Class09

Ayse

2/16/2022

### Protein structure

I downloaded the file "Data Export Summary.csv" from csv. I replaced the spaces with _.

```
#important to specify row names to make everything in the dataframe numbers
pdb_export <- read.csv("Data_Export_Summary.csv", row.names = 1)
pdb_export
```

```
##                           X.ray   NMR   EM Multiple.methods Neutron Other  Total
## Protein (only)          144433 11881 6732              182      70    32 163330
## Protein/Oligosaccharide   8543    31 1125                5       0     0   9704
## Protein/NA                7621   274 2165                3       0     0  10063
## Nucleic acid (only)       2396  1399   61                8       2     1   3867
## Other                      150    31    3                0       0     0    184
## Oligosaccharide (only)      11     6    0                1       0     4     22
```

```
#total number of structures
total_structures <- sum(pdb_export$Total)
#total XRay and NMR
total_x <- sum(pdb_export$X.ray)
total_n <- sum(pdb_export$EM)
#percent
pc_x <- (total_x/total_structures)*100
pc_em <- (total_n/total_structures)*100

#the easier way: use colSums for each column in the table, and refer to those when calculating percent.
totals <- colSums(pdb_export)
percents <- totals/totals["Total"]*100
round(percents, 3)
```

```
##            X.ray              NMR               EM Multiple.methods
##           87.169            7.278            5.389            0.106
##          Neutron            Other            Total
##            0.038            0.020          100.000
```

## Q1: What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy.

About 87.2% are solved by X-Ray, and about 5.4% are solved by EM. The percent of X-Ray structures is 87.169% and the percent of EM structures is 5.389%.

```
#already have total structures
protein_structures <- (pdb_export$Total[1]/totals["Total"])*100
```

## Q2: What proportion of structures in the PDB are protein?

About 87.263 #Q3: Type HIV in the PDB website search box on the home page and determine how many HIV-1 protease structures are in the current PDB? When I search HIV-1, I don't find any proteases in the current PDB.

### The PDB format

## Visualizing the HIV-1 protease structure

## Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure?

Hydrogens are too small to resolve. The PDB webpage showed the resolution to be 2 angstroms, and hydrogen is smaller than that. So we only see the oxygen atom for each water molecule.

## Q5: There is a conserved water molecule in the binding site. Can you identify this water molecule? What residue number does this water molecule have (see note below)?

I found the binding site by visualizing not protein (small molecule), which should be in the binding site. There was one water molecule near this small molecule, labeled HOH3O8:0.

## Q6: As you have hopefully observed HIV protease is a homodimer (i.e. it is composed of two identical chains). With the aid of the graphic display and the sequence viewer extension can you identify secondary structure elements that are likely to only form in the dimer rather than the monomer?

The binding site

## inserting image file

### Bio3d

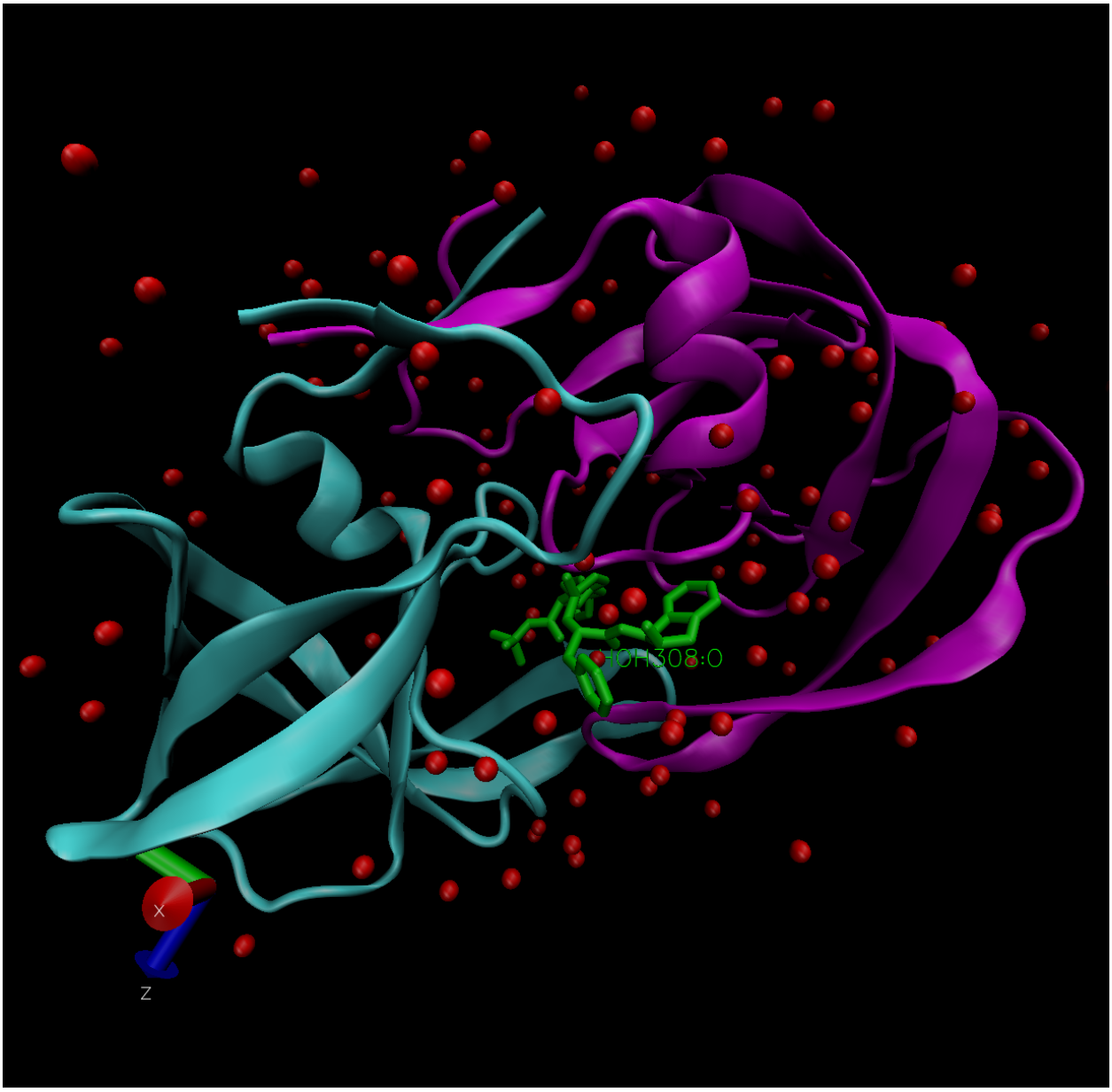



Figure 1:

```
#install and load bio3d
#install.packages("bio3d")
library(bio3d)
#read pdb file to R
pdb <- read.pdb("1hsg")
```

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

```
pdb
```

```
##
##  Call:  read.pdb(file = "1hsg")
##
##    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
```

```
#to find 3 letter  ode for amino acid
aa321("GLN")
```

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

## Q7: How many amino acid residues are there in this pdb object?

198 #Q8: Name one of the two non-protein residues? HOH (water) #Q9: How many protein chains are in this structure? 2

### Comparative structure analysis of Adenylate Kinase

```
#after installing required packages
#for packages not from CRAN
#BiocManager::install("msa")
#for packages from github or bitbucket
#devtools::install_bitbucket("Grantlab/bio3d-view")
```

Q10. Which of the packages above is found only on BioConductor and not CRAN? msa Q11. Which of the above packages is not found on BioConductor or CRAN?: bio3d_view Q12. True or False? Functions from the devtools package can be used to install packages from GitHub and BitBucket? TRUE

```
aa <- get.seq("1ake_A")
```

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

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

```
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
```

## Q13. How many amino acids are in this sequence, i.e. how long is this sequence?

214 amino acids

To find related sequences:

```
# Blast or hmmer search
#b <- blast.pdb(aa)
#did not use for markdown because it takes too long
```

```
# Plot a summary of search results
#hits <- plot(b)
```

```
# List out some 'top hits'
#head(hits$pdb.id)
hits <- NULL
hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A','6HAP_A','6HAM
```

## Align and superpose structures

```
# Download releated 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.gz exists. Skipping download

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

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

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

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

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

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

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

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

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

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

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

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

```
##
  |
  |                                                                      |   0%
  |
  |=====                                                                 |   8%
  |
  |===========                                                           |  15%
  |
  |================                                                      |  23%
  |
  |======================                                                |  31%
  |
  |============================                                          |  38%
  |
  |==================================                                    |  46%
  |
  |========================================                              |  54%
  |
  |==============================================                        |  62%
  |
  |====================================================                  |  69%
  |
  |==========================================================            |  77%
  |
  |================================================================      |  85%
  |
  |======================================================================|  92%
  |
  |======================================================================| 100%
```

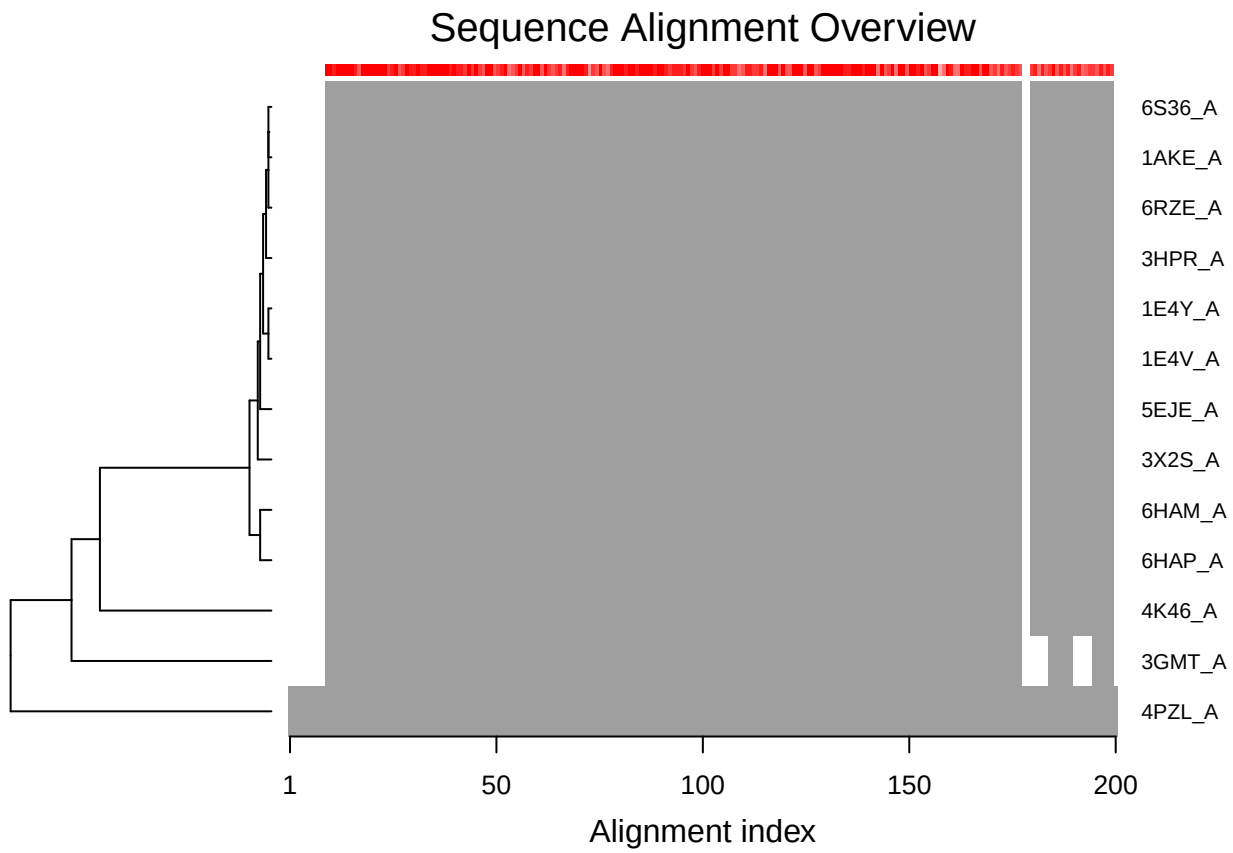
```
#if (!require("BiocManager", quietly = TRUE))
    #install.packages("BiocManager")
# Align releated PDBs
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

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

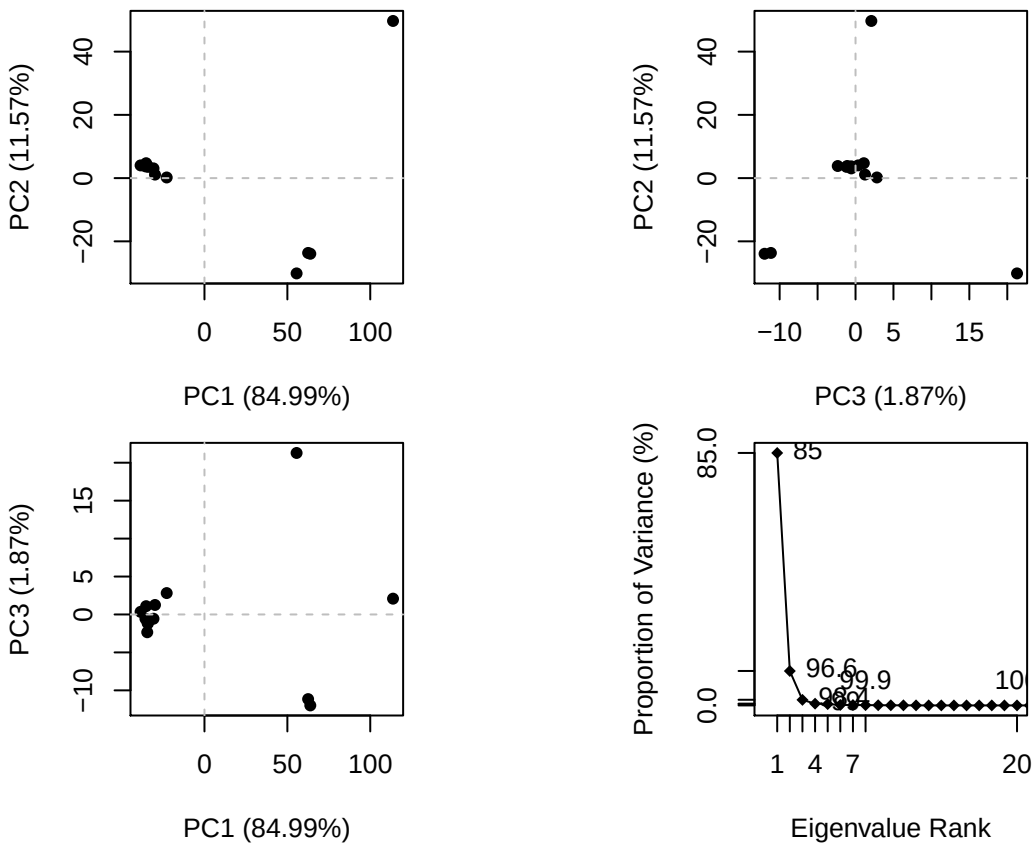
```
# Vector containing PDB codes for figure axis
ids <- basename.pdb(pdbs$id)

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




## Principal component analysis

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

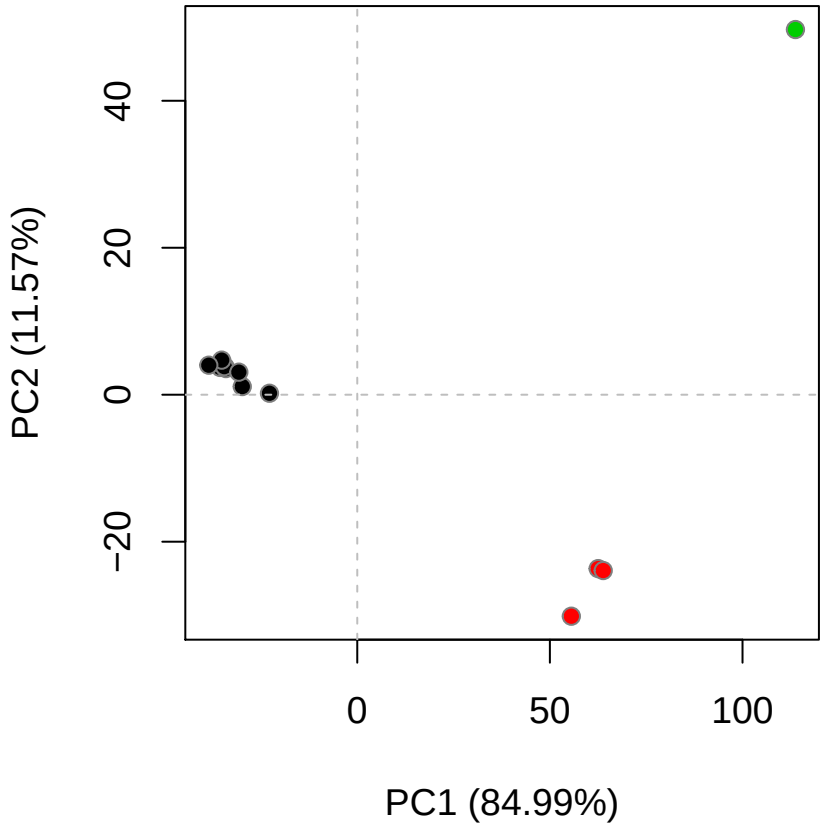



```
# Calculate RMSD
rd <- rmsd(pdbs)
```

```
## 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)

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




## Alphafold structure prediction for find-a-gene match

https://www.ebi.ac.uk/Tools/sss/ncbiblast/ >1-1773_3 1236 CloneMiner (cat#11144-010) Aureococcus anophagefferens cDNA 5', protein sequence LRAGHRARAAALGGVGAAGSAEFAEPVVPLRAPARGRGAAAAGRGRPGRRRRRAPRRLAR RAAAAHRAHRRRGRVAEIGAPSPLGAARRAPPAGAGLPPEFAIEVAARRAAPRGRGHHAG AADRRAAARRPGAPRLRGAHPRRDPMAREVARGARGGEPPVLRRRRRRRHAADPAAAPRG ALRGSDDASRGGVPAGLRAGSGRGRGCLDDYERHARAEDRLGRGGDRGDGDARRRLREDA DAALNQAWDLYYTVFRRVNKQLPQLTTLELRYVSPALLGARSLDLAVPGTYRVDGAGARI SRFSPSVHVITSKQRPRRLAMKGEDGREYGFLLKGHEDLRQDERAMQLFGLANALLAKDR RTREHGHLSIQRYAVTPLSHNCGVVGWVPACDTLHALVRDFRDARKIVLNVEHRVMLQLA PDYDALSLPQKVEVFDAALANTAGHDLSKVLWLKSSHSEQWLERRTHYARSLAAMSMVGH ILGLGDRHPSNLMLDRRTGKVLHIDX

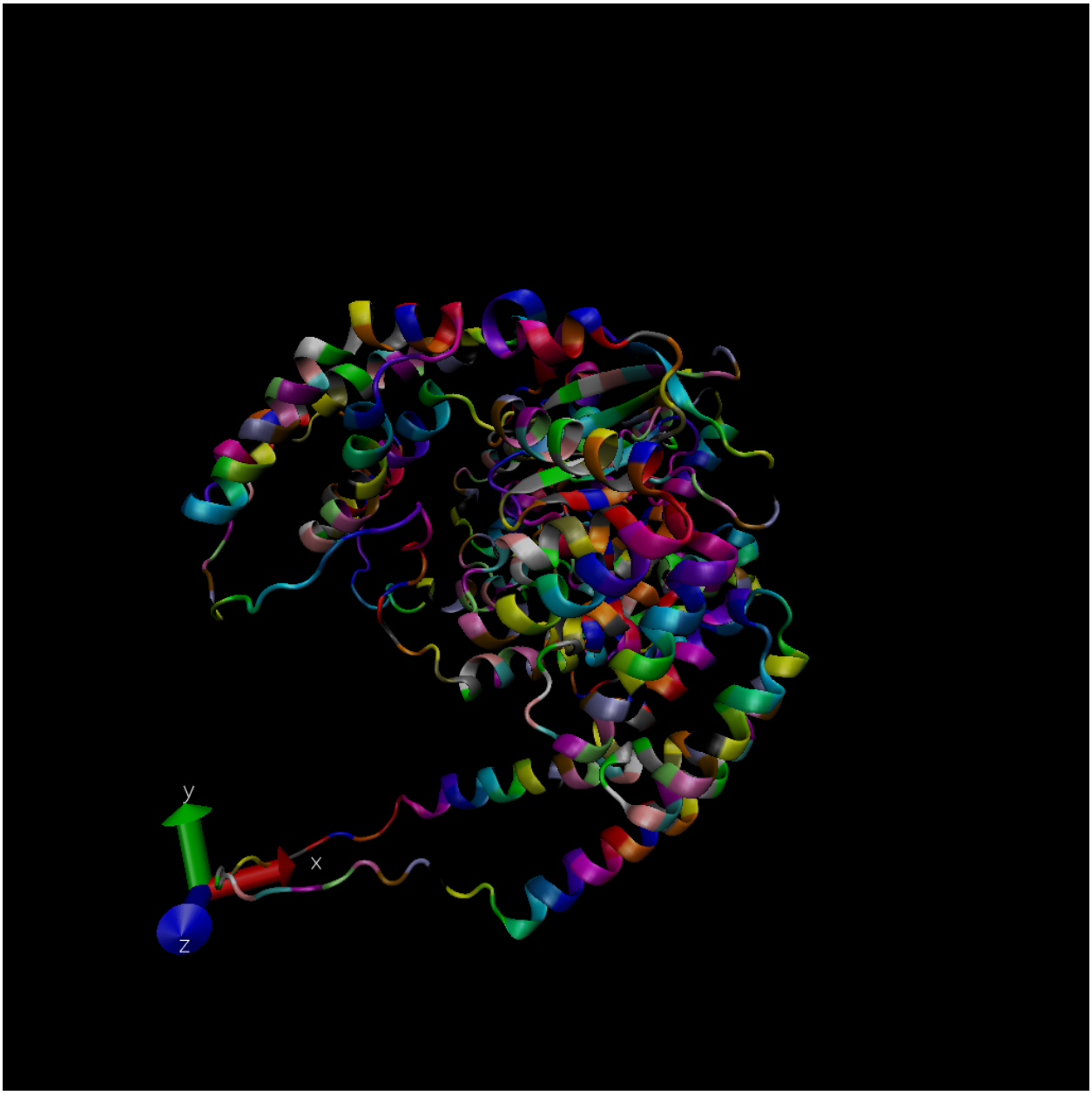



Figure 2: