November 3, 2023 Class 10: AlphaFold

Savannah Bogus A69027475

At this moment, there are 183,201 protein structures. In UniProt, there are 251,600,768 protein sequences.

```
stats<-read.csv("Data Export Summary.csv", row.names=1)
head(stats)
```

```
                          X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)          158,844 11,759 12,296              197      73    32
Protein/Oligosaccharide   9,260  2,054     34                8       1     0
Protein/NA                8,307  3,667    284                7       0     0
Nucleic acid (only)       2,730    113  1,467               13       3     1
Other                       164      9     32                0       0     0
Oligosaccharide (only)       11      0      6                1       0     4
                          Total
Protein (only)          183,201
Protein/Oligosaccharide  11,357
Protein/NA               12,265
Nucleic acid (only)       4,327
Other                       205
Oligosaccharide (only)       22
```

Note that because of the comma in the numbers, all of your numbers look like character strings :(`as.numeric` won't work with the commas. So, we're going to write a function that can remove commas any time. I used Claude to figure this out, as you suggested. There's a function called `sub()` in R that will substitute the first thing, but `gsub()` is a global substitution that can find a pattern and replace it globally, hence why Claude suggested that to me.

```
remove_commas<-function(x){
  #replace commas with nothing and convert to numeric
  as.numeric(gsub(",","",x))
}
```

```
str1<-"1,234"
remove_commas(str1)
```

```
[1] 1234
```

Now we have to use apply and apply it on the pdb thing.

```
pdbstats <- apply(stats,2,remove_commas)
```

We gotta do the row names thing again I think. If you do `rownames()` of stats vs. pdbstats, pdbstats returns a NULL.

```
rownames(pdbstats) <- rownames(stats)
pdbstats
```

```
                         X.ray    EM   NMR Multiple.methods Neutron Other
Protein (only)          158844 11759 12296              197      73    32
Protein/Oligosaccharide   9260  2054    34                8       1     0
Protein/NA                8307  3667   284                7       0     0
Nucleic acid (only)       2730   113  1467               13       3     1
Other                      164     9    32                0       0     0
Oligosaccharide (only)      11     0     6                1       0     4
                         Total
Protein (only)          183201
Protein/Oligosaccharide  11357
Protein/NA               12265
Nucleic acid (only)       4327
Other                      205
Oligosaccharide (only)      22
```

Q1

About 93% of the structures in PDB are solved by X-ray and EM. Mostly by X-ray. I asked this to Claude, but you're about to have us do code for it. Which I just found out is betterbecause Claude was wrong.

```r
totals<- apply(pdbstats,2,sum)
round(totals/totals["Total"]*100,2)
```

```
          X.ray                EM               NMR Multiple.methods 
          84.83              8.33              6.68             0.11 
        Neutron             Other             Total 
           0.04              0.02            100.00 
```

Q2-Q3

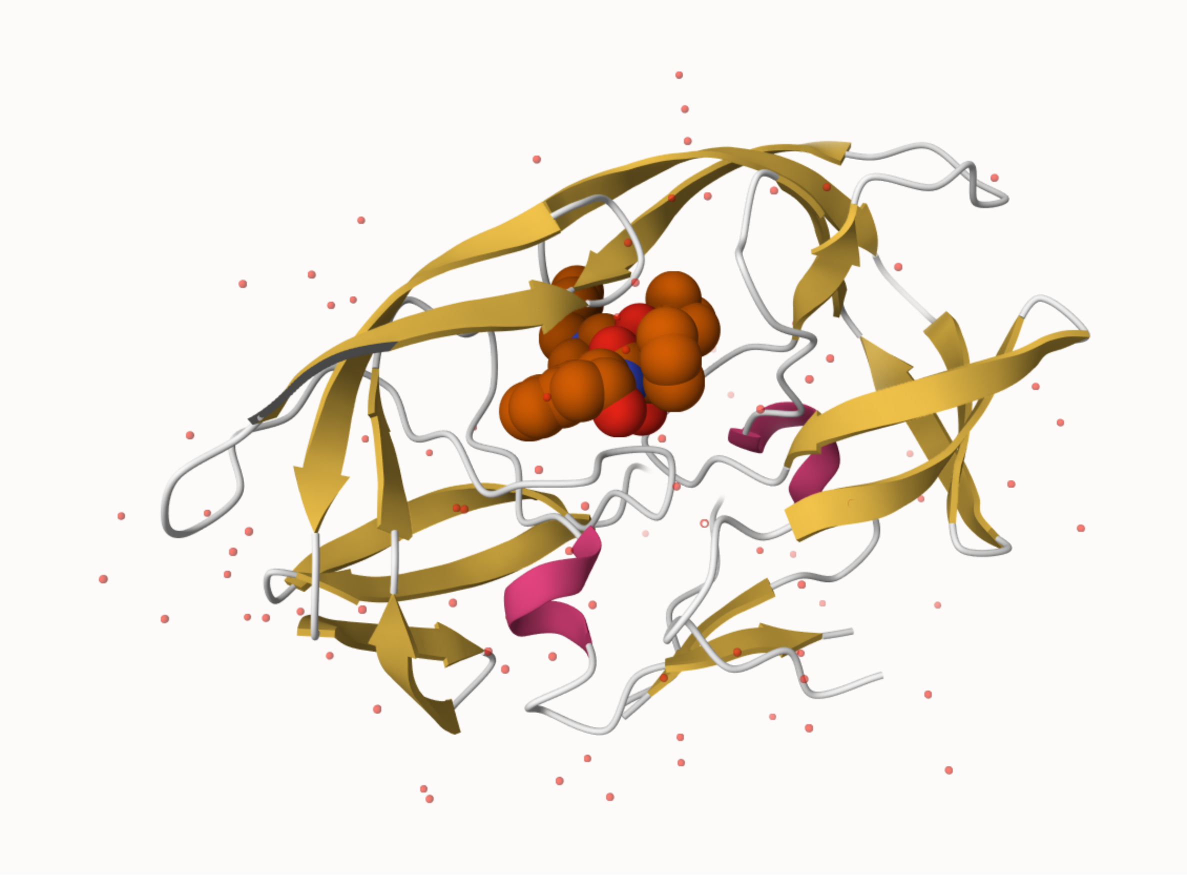
We're skipping these

Visualizing the HIV-1 protease structure

Q4

There's only one atom per water molecule because the resolution is 2.00 Angstroms, and Hydrogen is smaller than that.

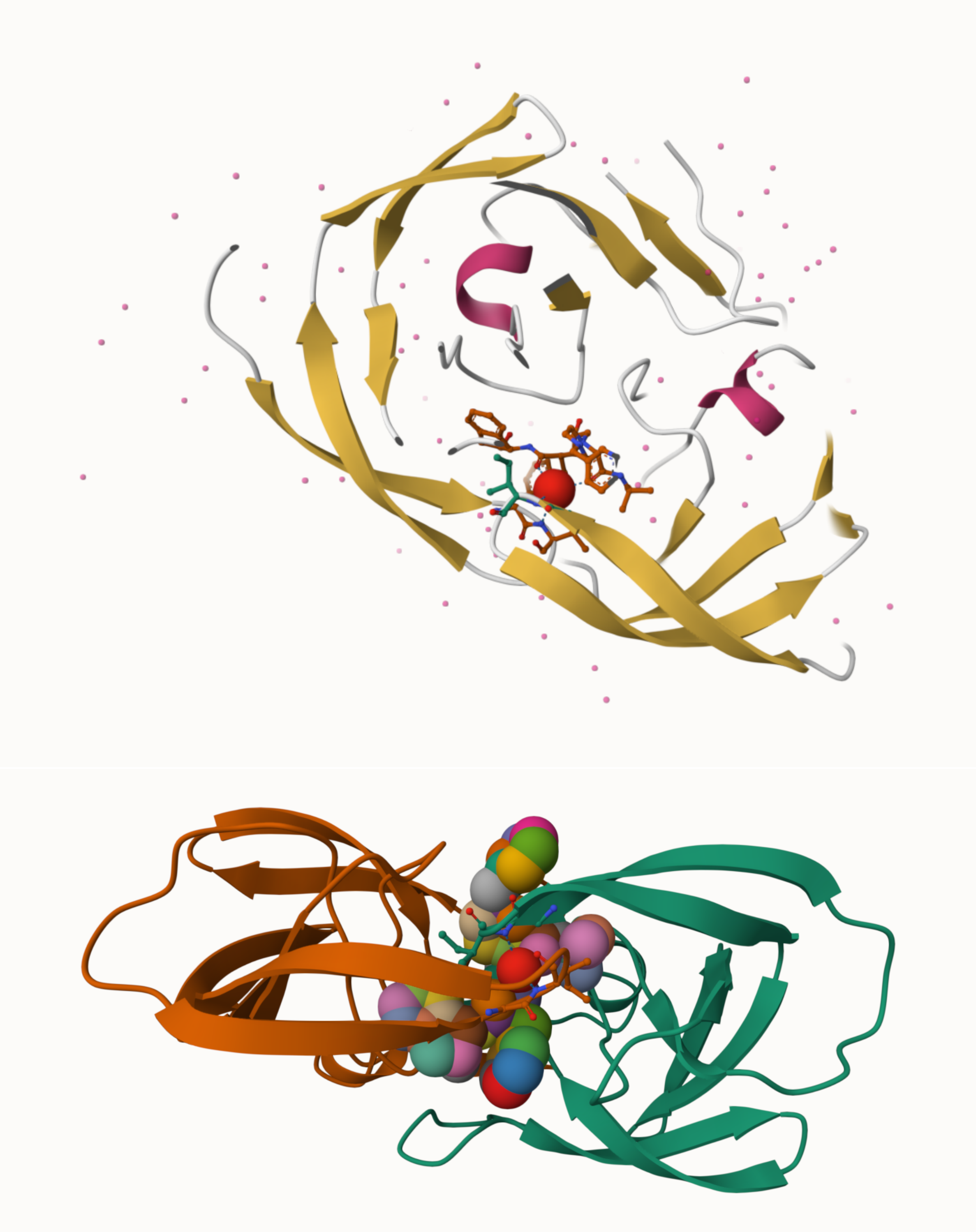
HIV-Pr image below.



Q5

Water 308 is conserved.

Q6



Q7 [optional]

That area that the H2O 308 binds wouldn't be there without the homodimer.

Introduction to Bio3D in R

```r
library(bio3d)
pdb<-read.pdb("1hsg")
```

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

```r
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
```

Q7 [for real this time]

198

Q8

HOH (water) and MK1 (ligand)

Q9

There are 2 chains in the structure.

```r
attributes(pdb)
```

```
$names
[1] "atom"   "xyz"    "seqres" "helix"  "sheet"  "calpha" "remark" "call"  

$class
[1] "pdb" "sse"
```

```r
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>
```

```r
adk<-read.pdb("6s36")
```

```
  Note: Accessing on-line PDB file
   PDB has ALT records, taking A only, rm.alt=TRUE
```

```
adk
```

```

 Call:  read.pdb(file = "6s36")

   Total Models#: 1
     Total Atoms#: 1898,  XYZs#: 5694  Chains#: 1  (values: A)

     Protein Atoms#: 1654  (residues/Calpha atoms#: 214)
     Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)

     Non-protein/nucleic Atoms#: 244  (residues: 244)
     Non-protein/nucleic resid values: [ CL (3), HOH (238), MG (2), NA (1) ]

   Protein sequence:
      MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
      DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDKI
      VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
      YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

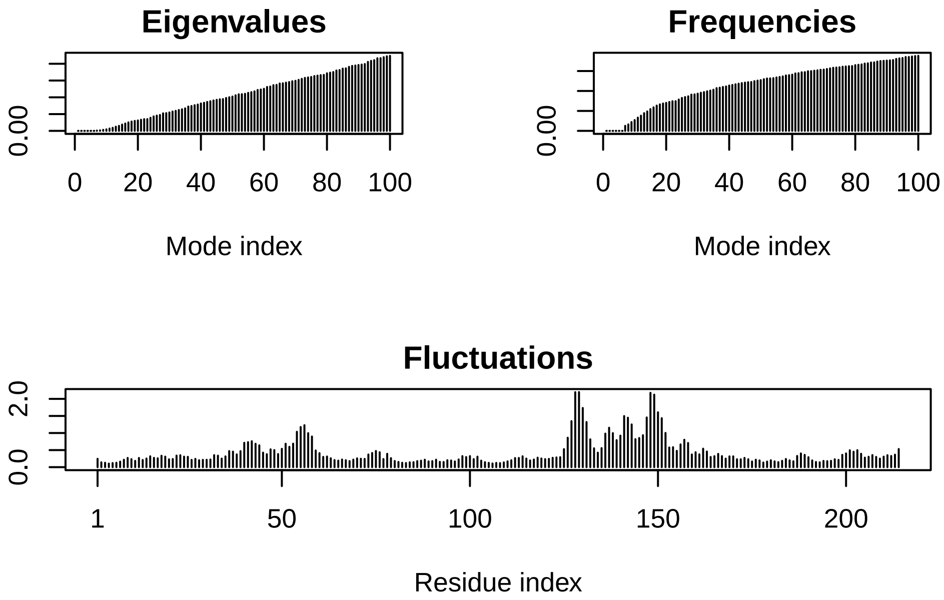
Looks like we're doing a normal mode analysis (NMA) which predicts protein flexibility and potential functional motions/conformational changes.

```
#flexibility prediciton
m<-nma(adk)
```

```

 Building Hessian...        Done in 0.015 seconds.
 Diagonalizing Hessian...   Done in 0.275 seconds.
```

```
plot(m)
```



To view a "movie" of these predicted motions we can generate a molecular "trajectory" with the mktrj() function.

```
mktrj(m,file="adk_modes.pdb")
```

Comparative structure analysis of Adenylate Kinase

Q10

msa is the package found on bioconductor and not CRAN

Q11

Q12

devtools can be used to install packages from github and bitbucket.

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

```
Warning in get.seq("1ake_A"): 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
```

Q13

214 amino acids.

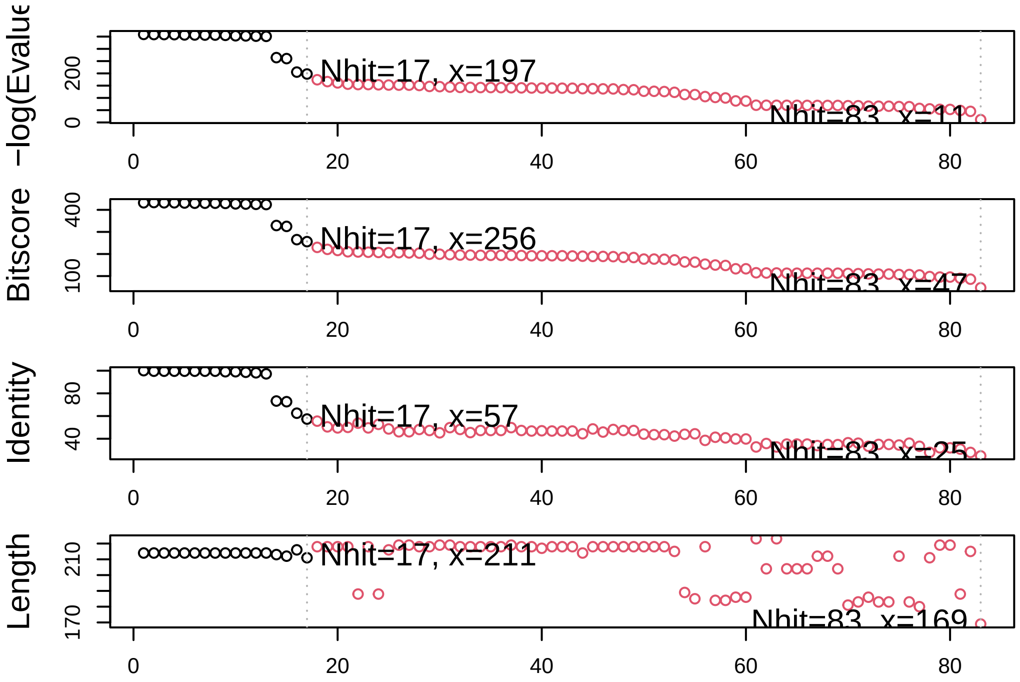
```r
blast<-blast.pdb(aa)
```

```
Searching ... please wait (updates every 5 seconds) RID = MJGHPEXV01N
.
Reporting 83 hits
```

```
hits<-plot(blast)
```

```
 * Possible cutoff values:    197 11
            Yielding Nhits:    17 83

 * Chosen cutoff value of:    197
            Yielding Nhits:    17
```



```
#hit em with the top hits
head(hits$pdb.id)
```

```
[1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A"
```

Now we're going to download related PDB files. I don't understand this code. That's probably my fault.

```r
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/8BQF.pdb.gz exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8M.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/4X8H.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/4NP6.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%
  |
  |====                                                                  |   6%
  |
  |========                                                              |  12%
  |
  |============                                                          |  18%
  |
  |================                                                      |  24%
  |
  |=====================                                                 |  29%
  |
  |=========================                                             |  35%
  |
  |=============================                                         |  41%
  |
  |=================================                                     |  47%
  |
  |=====================================                                 |  53%
  |
  |=========================================                             |  59%
  |
  |=============================================                         |  65%
  |
  |=================================================                     |  71%
  |
  |======================================================                |  76%
  |
  |==========================================================            |  82%
  |
  |==============================================================        |  88%
  |
  |==================================================================    |  94%
  |
  |======================================================================| 100%
```

Now we can align and superimpose structures. `pdbaln()` is how we align.

```
pdbalign<-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/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
```

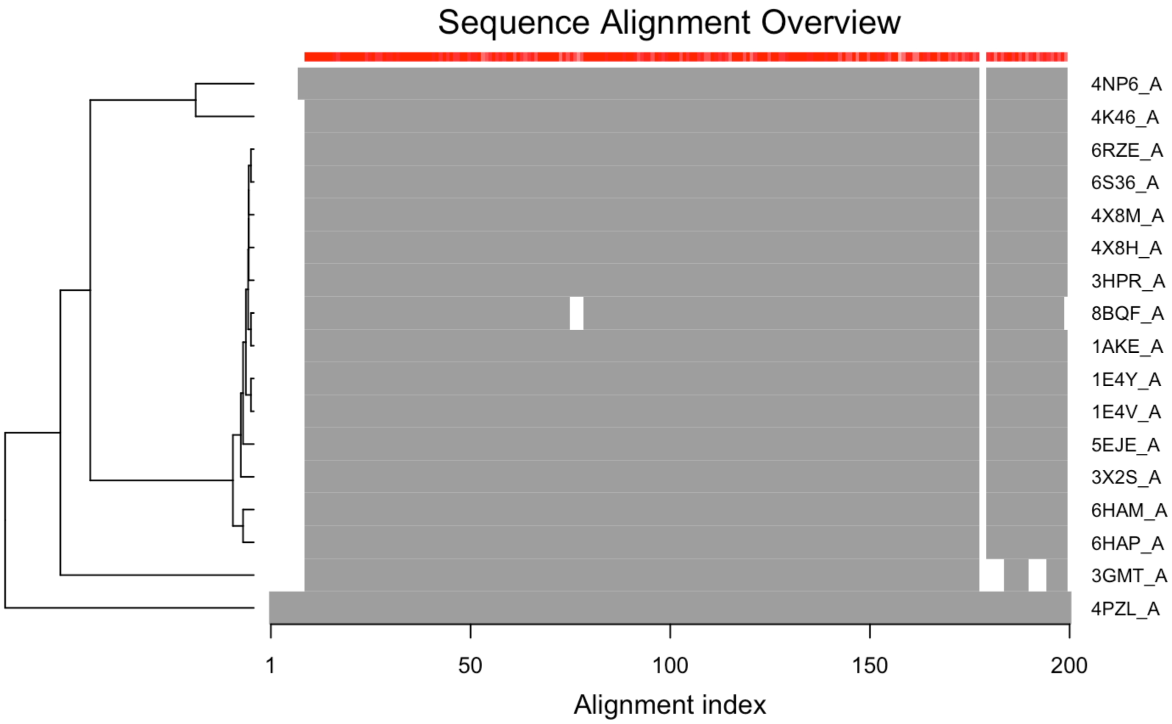
```
....

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

Now I'm going to make a vector containing PDB codes for figure axes and then draw a schematic alignment.

```r
ids<-basename.pdb(pdbalign$id)
#plot(pdbalign,labels=ids, dev='png')
```



My plot doesn't have the sequence alignment overview? »»»me from the future couldn't render because of this plot so I asked Claude and Claude told me to set `dev='png'` or any other raster format which worked beautifully and fixed my plot And apparently my plot will NOT render because the "figure margins are too large" no matter what I try or what I ask Claude to do or even asking a coding friend what to do. I inserted the png so you know I did it, and then I made the plot code a comment so that I can actually render this whole thing. Now apparently I can 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 experimentalTechnique
1AKE_A        1AKE       A           Protein         214                 X-ray
8BQF_A        8BQF       A           Protein         234                 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
8BQF_A      2.050             <NA> Adenylate kinase (ADK)              AP5
4X8M_A      2.600             <NA> Adenylate kinase (ADK)             <NA>
6S36_A      1.600             <NA> Adenylate kinase (ADK) NA,MG (2),CL (3)
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'-PENTAPHOSPHATE
8BQF_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
4X8M_A                                                                             <NA>
```

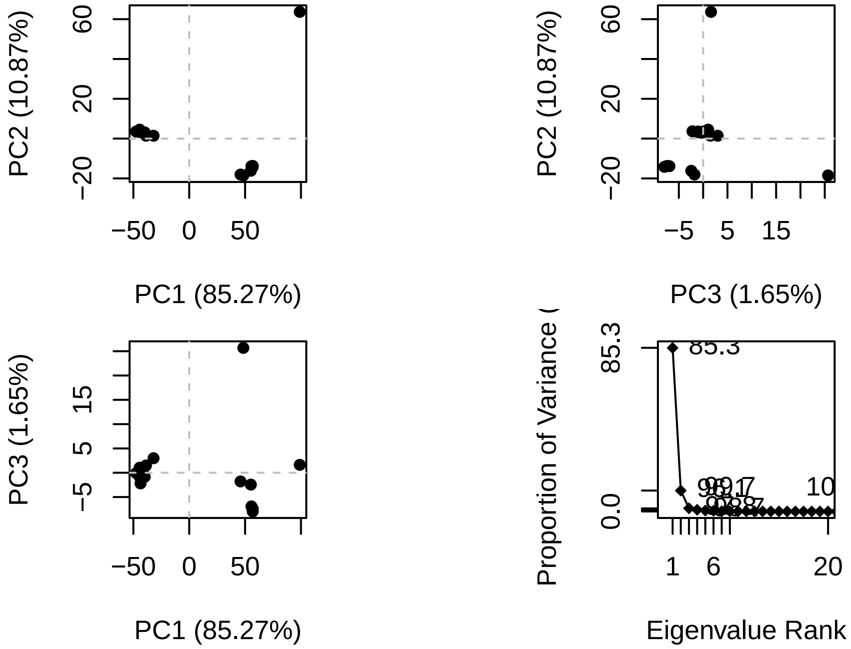
```
6S36_A                                    SODIUM ION,MAGNESIUM ION (2),CHLORIDE ION (3)
6RZE_A                                                  SODIUM ION (3),CHLORIDE ION (2)
4X8H_A                                                                             <NA>
3HPR_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
1E4V_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
5EJE_A                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE,COBALT (II) ION
1E4Y_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
3X2S_A N-(pyren-1-ylmethyl)acetamide (2),BIS(ADENOSINE)-5'-PENTAPHOSPHATE,MAGNESIUM ION
6HAP_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6HAM_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
4K46_A                   ADENOSINE-5'-DIPHOSPHATE,ADENOSINE MONOPHOSPHATE,PHOSPHATE ION
4NP6_A                                                                             <NA>
3GMT_A                                                                  SULFATE ION (2)
4PZL_A                                                 CALCIUM ION,FORMIC ACID,GLYCEROL
                                                 source
1AKE_A                                 Escherichia coli
8BQF_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

1AKE_A STRUCTURE OF THE COMPLEX BETWEEN ADENYLATE KINASE FROM ESCHERICHIA COLI AND THE INHIBI
8BQF_A
4X8M_A
6S36_A
6RZE_A
4X8H_A
3HPR_A
1E4V_A
5EJE_A                                                                                  Crys
1E4Y_A
```

```
3X2S_A
6HAP_A
6HAM_A
4K46_A
4NP6_A
3GMT_A
4PZL_A                                                                              The crys
                                                     citation rObserved   rFree
1AKE_A                 Muller, C.W., et al. J Mol Biol (1992)   0.19600      NA
8BQF_A   Scheerer, D., et al. Proc Natl Acad Sci U S A (2023)   0.22073 0.25789
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
8BQF_A 0.21882  P 2 21 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
```

Alright, time to start performing PCA.

```
pc.xray<-pca(pdbalign)
plot(pc.xray)
```



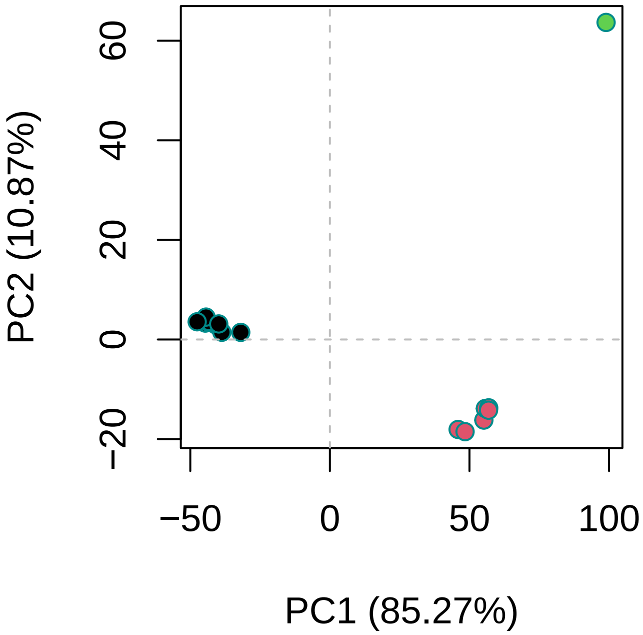
Next we're going to calculate RMSD.

```
rd<-rmsd(pdbalign)
```

```
Warning in rmsd(pdbalign): No indices provided, using the 199 non NA positions
```

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

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



Optional further visualization

We can VISUALIZE the principal components!

```
pc1<- mktrj(pc.xray,pc=1,file="pc_1.pdb")
```

CRAZY.

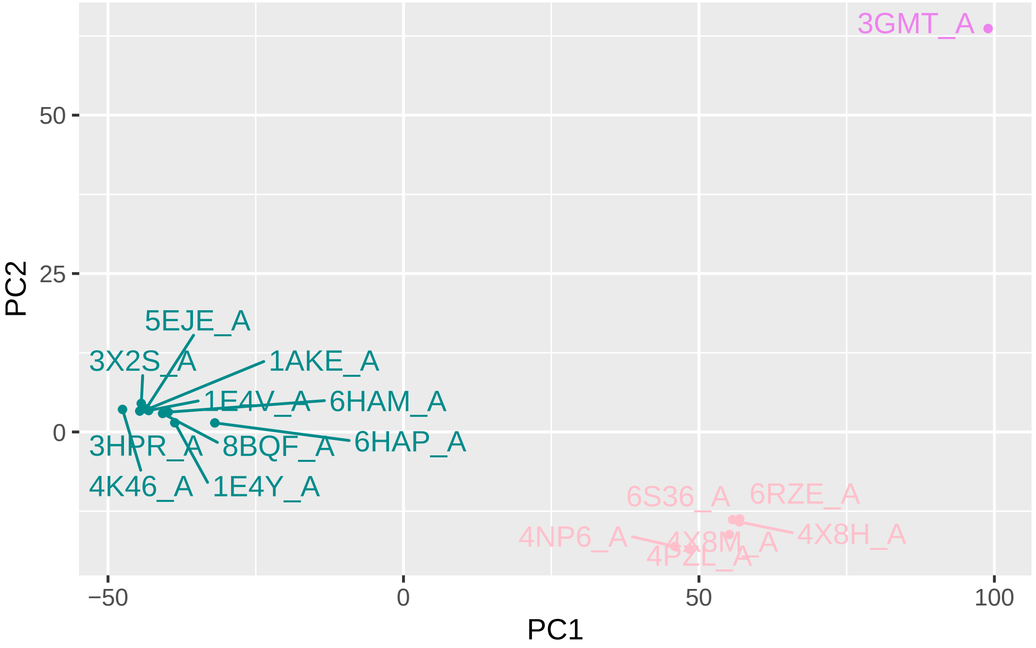
we can also plot this in ggplot and with ggrepel.

```
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=1)+
  geom_text_repel(max.overlaps=20)+
  theme(legend.position="none")+
  scale_color_manual(values=c("darkcyan","pink","violet"))

p
```



Normal Mode Analysis

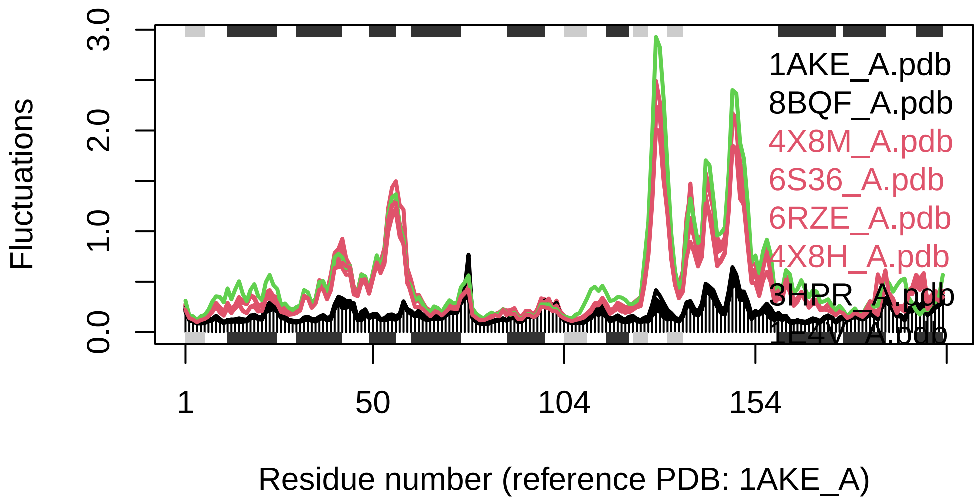
```
modes<-nma(pdbalign)
```

```
Warning in nma.pdbs(pdbalign): 8BQF_A.pdb might have missing residue(s) in structure:
   Fluctuations at neighboring positions may be affected.


Details of Scheduled Calculation:
  ... 17 input structures
  ... storing 591 eigenvectors for each structure
  ... dimension of x$U.subspace: ( 597x591x17 )
  ... coordinate superposition prior to NM calculation
  ... aligned eigenvectors (gap containing positions removed)
  ... estimated memory usage of final 'eNMA' object: 45.9 Mb
  |
  |                                                                      |   0%
  |
  |====                                                                  |   6%
  |
  |========                                                              |  12%
  |
  |============                                                          |  18%
  |
  |================                                                      |  24%
  |
  |=====================                                                 |  29%
  |
  |=========================                                             |  35%
  |
  |=============================                                         |  41%
  |
  |=================================                                     |  47%
  |
  |=====================================                                 |  53%
  |
  |=========================================                             |  59%
  |
  |=============================================                         |  65%
  |
  |=================================================                     |  71%
  |
  |======================================================                |  76%
  |
  |==========================================================            |  82%
  |
  |==============================================================        |  88%
  |
  |==================================================================    |  94%
  |
  |======================================================================| 100%
```

```
plot(modes,pdbalign,col=grps.rd)
```

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



Q14

A lot of the upper lines are almost like an amplified version of the black lines. I think it indicates areas that are not as conserved and are able to have fluctuations without impacting main function as majorly.