

Lab 11 report Bangyan Hu (Part 1 and Part 2)

Bangyan Hu (PID: A15540189)

11/2/2021

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

	X.ray	NMR	EM	Multiple.methods	Neutron	Other	Total
## Protein (only)	142303	11804	5999	177	70	32	160385
## Protein/Oligosaccharide	8414	31	979	5	0	0	9429
## Protein/NA	7491	274	1986	3	0	0	9754
## Nucleic acid (only)	2368	1372	60	8	2	1	3811
## Other	149	31	3	0	0	0	183
## Oligosaccharide (only)	11	6	0	1	0	4	22

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

```
method.sums <- colSums(db)
round((method.sums/method.sums["Total"]) * 100, 2)
```

##	X.ray	NMR	EM	Multiple.methods
##	87.55	7.36	4.92	0.11
##	Neutron	Other	Total	
##	0.04	0.02	100.00	

87.55 percentage of structures in the PDB are solved by X-Ray, and 4.92 percentage of structures in the PDB are solved by Electron Microscopy

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

```
type.sums <- rowSums(db)
type.sums[1]/method.sums["Total"]
```

```
## Protein (only)
## 1.747266
```

```
round(db$Total/method.sums["Total"] * 100, 2)
```

```
## [1] 87.36 5.14 5.31 2.08 0.10 0.01
```

```
proportion_of_protein = 87.36 + 5.14 + 5.31
proportion_of_protein
```

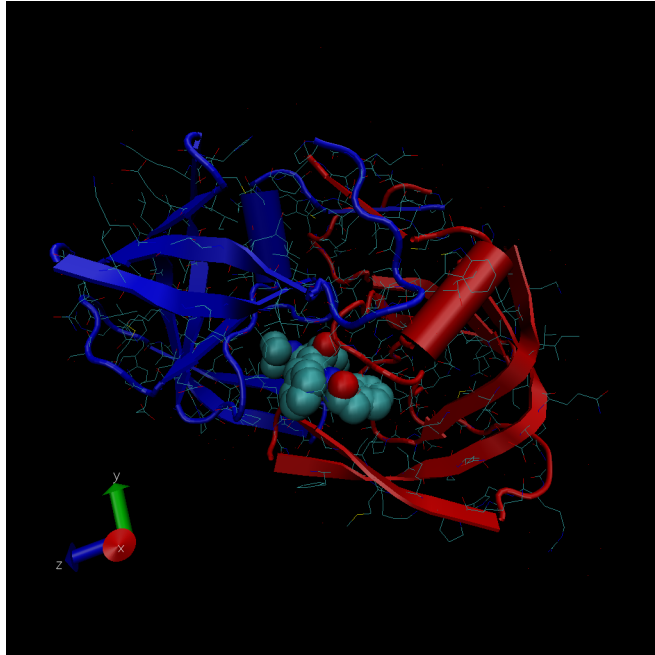
[1] 97.81

97.81% (0.9781) of structures in the PDB are protein.

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?

There are 1828 HIV-1 protease structures in the current PDB.

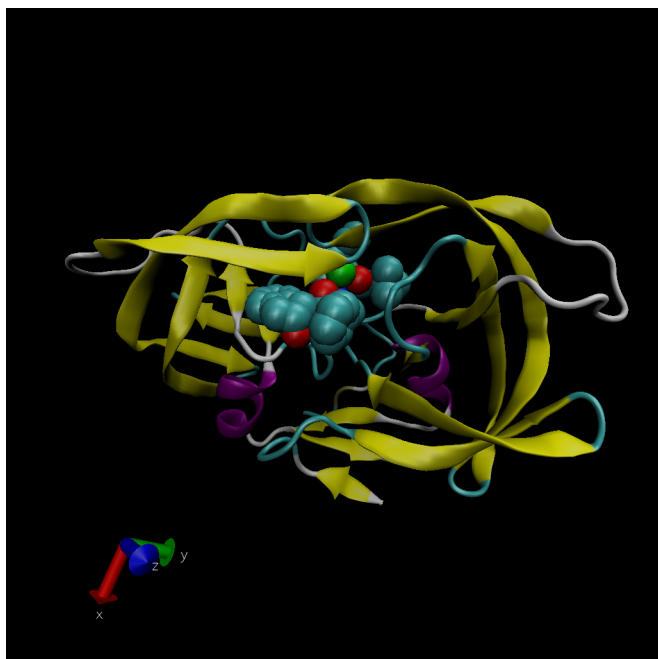
VMD structure visualization image



Just write text but you have some **formatting** options

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

Water molecules normally have 3 atoms but we do see just one atom (the oxygen atom that is relatively big enough to be seen under the resolution capacity) per water molecule in this structure because the resolution capacity is not high enough to observe the hydrogen atoms that are too small.



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)?

Yes, I can identify this water molecule. That is resid 308 (The residue number this water molecule has is 308).

#Sequence Viewer Extension [OPTIONAL]

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?

Launching the Sequence Viewer click VMD Main window through Extensions and Analysis and Sequence Viewer, I observed different color scales beside the sequence correspond to the B-factor and Secondary structure type. I found that the major ones being Extended (beta) in yellow and Helix in purple (yellow E: extended conformation; purple H: alpha helix). As the major secondary structure elements are extended conformation and alpha helix, I can identify that these elements are likely to only form in the dimer rather than the monomer.

Introduction to Bio3D in R

I need load the library

Reading PDB file data into R

```
library(bio3d)

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

Note: Accessing on-line PDB file

```
pdb
```

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

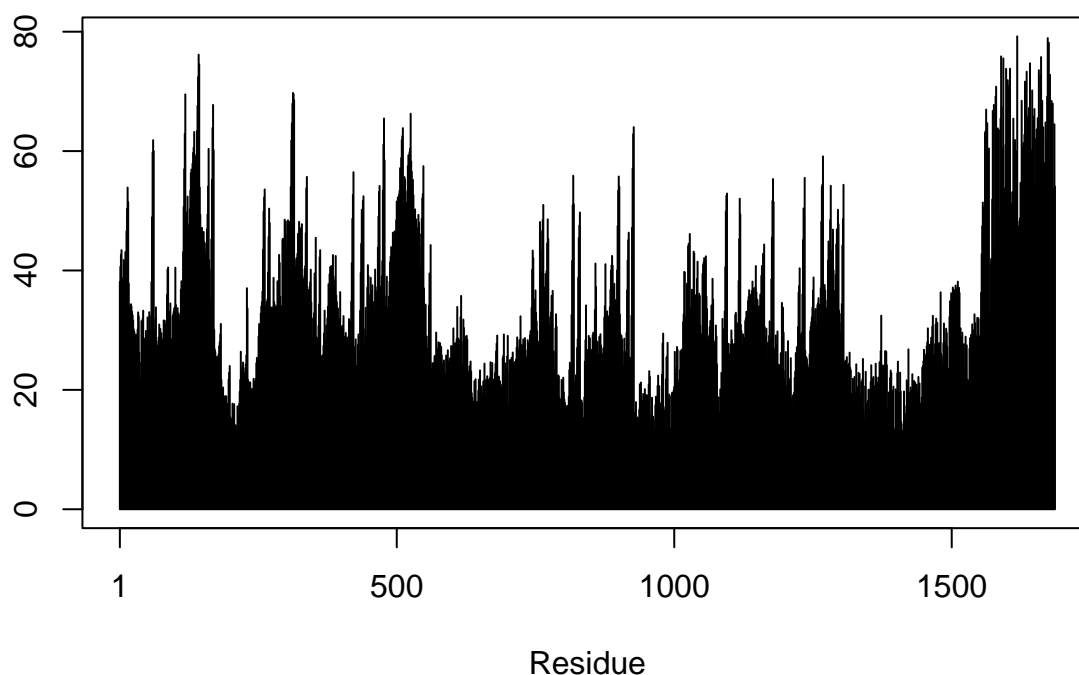
```
##
## Total Models#: 1
## Total Atoms#: 1686, XYZs#: 5058 Chains#: 2 (values: A B)
##
## Protein Atoms#: 1514 (residues/Calpha atoms#: 198)
## Nucleic acid Atoms#: 0 (residues/phosphate atoms#: 0)
##
## Non-protein/nucleic Atoms#: 172 (residues: 128)
## Non-protein/nucleic resid values: [ HOH (127),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

```
aa123( pdbseq(pdb))
```

```
## [1] "PRO" "GLN" "ILE" "THR" "LEU" "TRP" "GLN" "ARG" "PRO" "LEU" "VAL" "THR"
## [13] "ILE" "LYS" "ILE" "GLY" "GLY" "GLN" "LEU" "LYS" "GLU" "ALA" "LEU" "LEU"
## [25] "ASP" "THR" "GLY" "ALA" "ASP" "ASP" "THR" "VAL" "LEU" "GLU" "GLU" "MET"
## [37] "SER" "LEU" "PRO" "GLY" "ARG" "TRP" "LYS" "PRO" "LYS" "MET" "ILE" "GLY"
## [49] "GLY" "ILE" "GLY" "GLY" "PHE" "ILE" "LYS" "VAL" "ARG" "GLN" "TYR" "ASP"
## [61] "GLN" "ILE" "LEU" "ILE" "GLU" "ILE" "CYS" "GLY" "HIS" "LYS" "ALA" "ILE"
## [73] "GLY" "THR" "VAL" "LEU" "VAL" "GLY" "PRO" "THR" "PRO" "VAL" "ASN" "ILE"
## [85] "ILE" "GLY" "ARG" "ASN" "LEU" "LEU" "THR" "GLN" "ILE" "GLY" "CYS" "THR"
## [97] "LEU" "ASN" "PHE" "PRO" "GLN" "ILE" "THR" "LEU" "TRP" "GLN" "ARG" "PRO"
## [109] "LEU" "VAL" "THR" "ILE" "LYS" "ILE" "GLY" "GLY" "GLN" "LEU" "LYS" "GLU"
## [121] "ALA" "LEU" "LEU" "ASP" "THR" "GLY" "ALA" "ASP" "ASP" "THR" "VAL" "LEU"
## [133] "GLU" "GLU" "MET" "SER" "LEU" "PRO" "GLY" "ARG" "TRP" "LYS" "PRO" "LYS"
## [145] "MET" "ILE" "GLY" "GLY" "ILE" "GLY" "GLY" "PHE" "ILE" "LYS" "VAL" "ARG"
## [157] "GLN" "TYR" "ASP" "GLN" "ILE" "LEU" "ILE" "GLU" "ILE" "CYS" "GLY" "HIS"
## [169] "LYS" "ALA" "ILE" "GLY" "THR" "VAL" "LEU" "VAL" "GLY" "PRO" "THR" "PRO"
## [181] "VAL" "ASN" "ILE" "ILE" "GLY" "ARG" "ASN" "LEU" "LEU" "THR" "GLN" "ILE"
## [193] "GLY" "CYS" "THR" "LEU" "ASN" "PHE"
```

```
plot.bio3d(pdb$atom$b, sse=pdb)
```

```
## Warning in plotb3(...): Length of input 'sse' does not equal the length of input
## 'x'; Ignoring 'sse'
```



The ATOM records

```
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>
```

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

There are 198 amino acid residues in this pdb object.

Q8: Name one of the two non-protein residues?

One of the two non-protein residues is HOH.

Q9: How many protein chains are in this structure?

2 protein chains are in this structure.

```
attributes(pdb)
```

```
## $names
## [1] "atom" "xyz" "seqres" "helix" "sheet" "calpha" "remark" "call"
##
## $class
## [1] "pdb" "sse"
```

```
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>
```

Comparative analysis of protein structures

Setup

```
# Install packages in the R console not your Rmd

#install.packages("bio3d")
#install.packages("ggplot2")
#install.packages("ggrepel")
#install.packages("devtools")
#install.packages("BiocManager")
#BiocManager::install("msa")
#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?

The package bio3d-view is not found on BioConductor or CRAN.

Q12. True or False? Functions from the devtools package can be used to install packages from GitHub and BitBucket?

True.

Using the bio3d package.

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

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

```
pdb
```

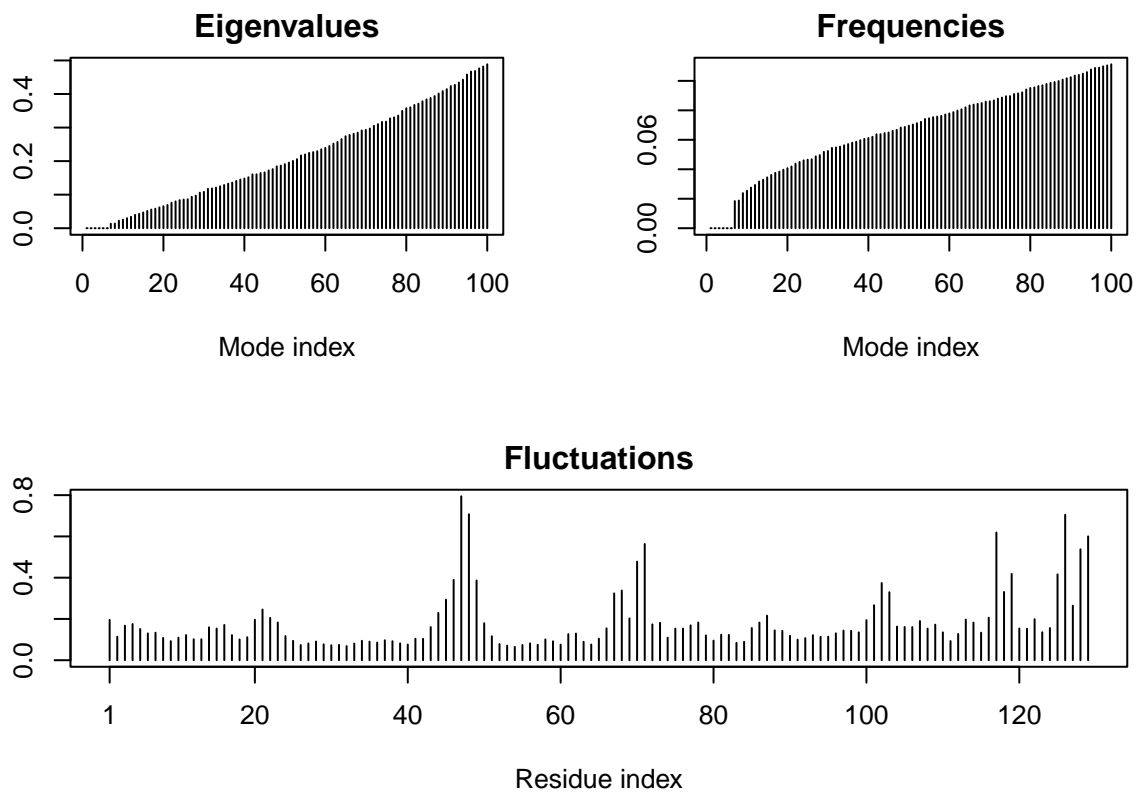
```
##
## Call: read.pdb(file = "1hel")
##
## Total Models#: 1
## Total Atoms#: 1186, XYZs#: 3558 Chains#: 1 (values: A)
##
## Protein Atoms#: 1001 (residues/Calpha atoms#: 129)
## Nucleic acid Atoms#: 0 (residues/phosphate atoms#: 0)
##
## Non-protein/nucleic Atoms#: 185 (residues: 185)
## Non-protein/nucleic resid values: [ HOH (185) ]
##
## Protein sequence:
## KVFGRCELAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTDGSTDYGILQINS
## RWWCNDGRTPGSRNLCNIPCSALLSSDITASVNCAKKIVSDGNGMNAWVAWRNRCKGTDV
## QAWIRGCRL
##
## + attr: atom, xyz, seqres, helix, sheet,
## calpha, remark, call
```

Let's use NMA to predict the dynamics (flexibility) of this enzyme.

```
modes <- nma(pdb)
```

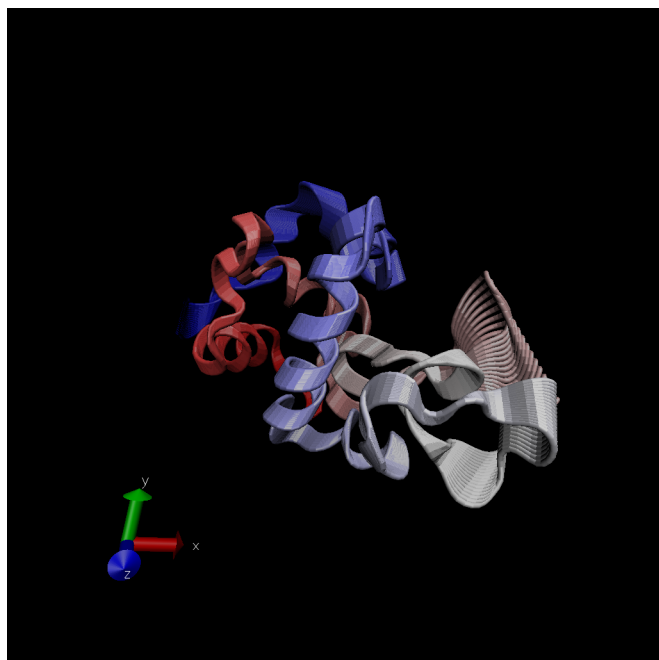
```
## Building Hessian... Done in 0.007 seconds.
## Diagonalizing Hessian... Done in 0.065 seconds.
```

```
plot(modes)
```



Make a “move” of its predicted motion - “trajectory”

```
mktrj(modes, file = "nma.pdb")
```



Analysis of ADE

```
library(bio3d)
aa<- get.seq("lake_A")
```

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

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

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

```
## Searching ... please wait (updates every 5 seconds) RID = S9U26MPN01R
```

```
## .
```

```
## Reporting 100 hits
```

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

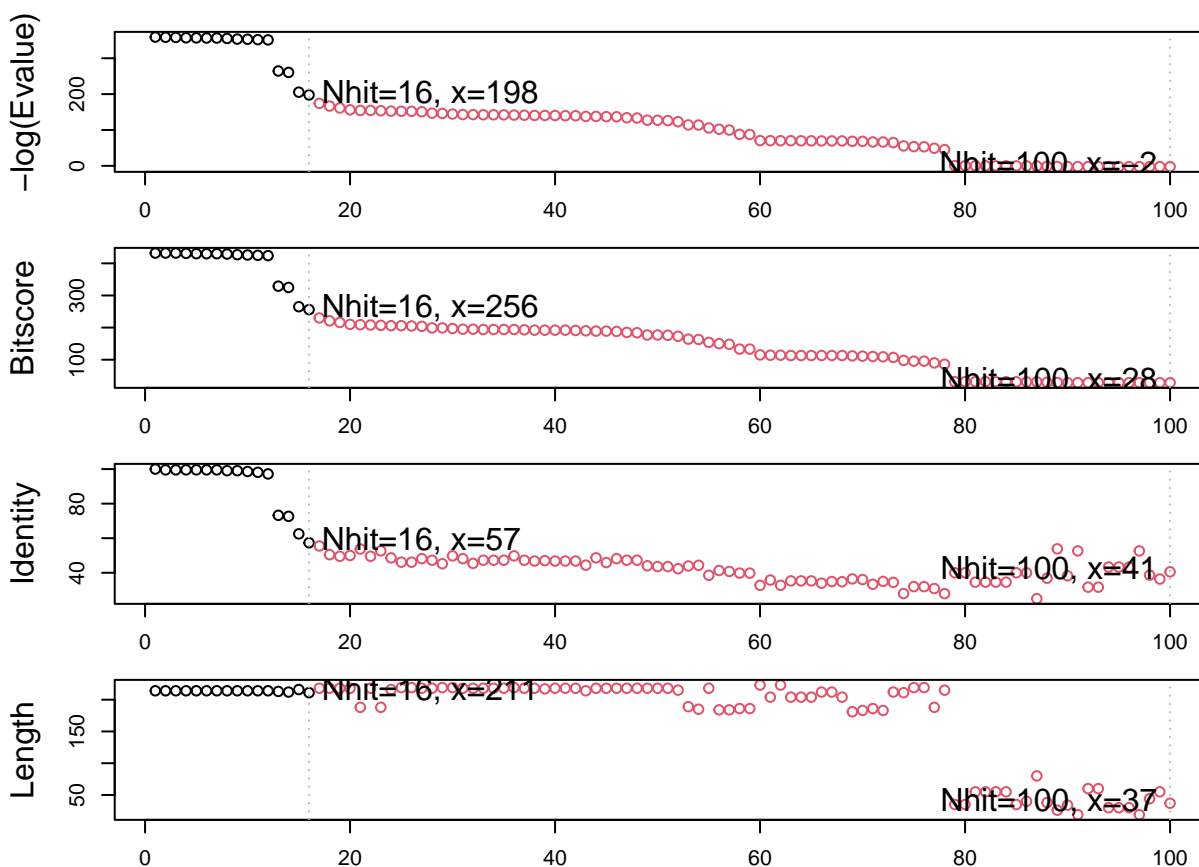
```
## * Possible cutoff values: 197 -3
```

```
##      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 TRUE
## [13] TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [97] FALSE FALSE FALSE FALSE
##
## attr(,"class")
## [1] "blast"
```

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

214 amino acids are in this sequence.

```
# List out some 'top hits'
#head(hits$pdb.id)
```

```
hits <- NULL
hits$pdb.id <- c('1AKE_A','4X8M_A','6S36_A','6RZE_A','4X8H_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A')
```

```

# 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.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

```

```
## |
```

```
|
```

Align and superpose structures

```
# Align related PDBs
```

```
pdbbs <- 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
```

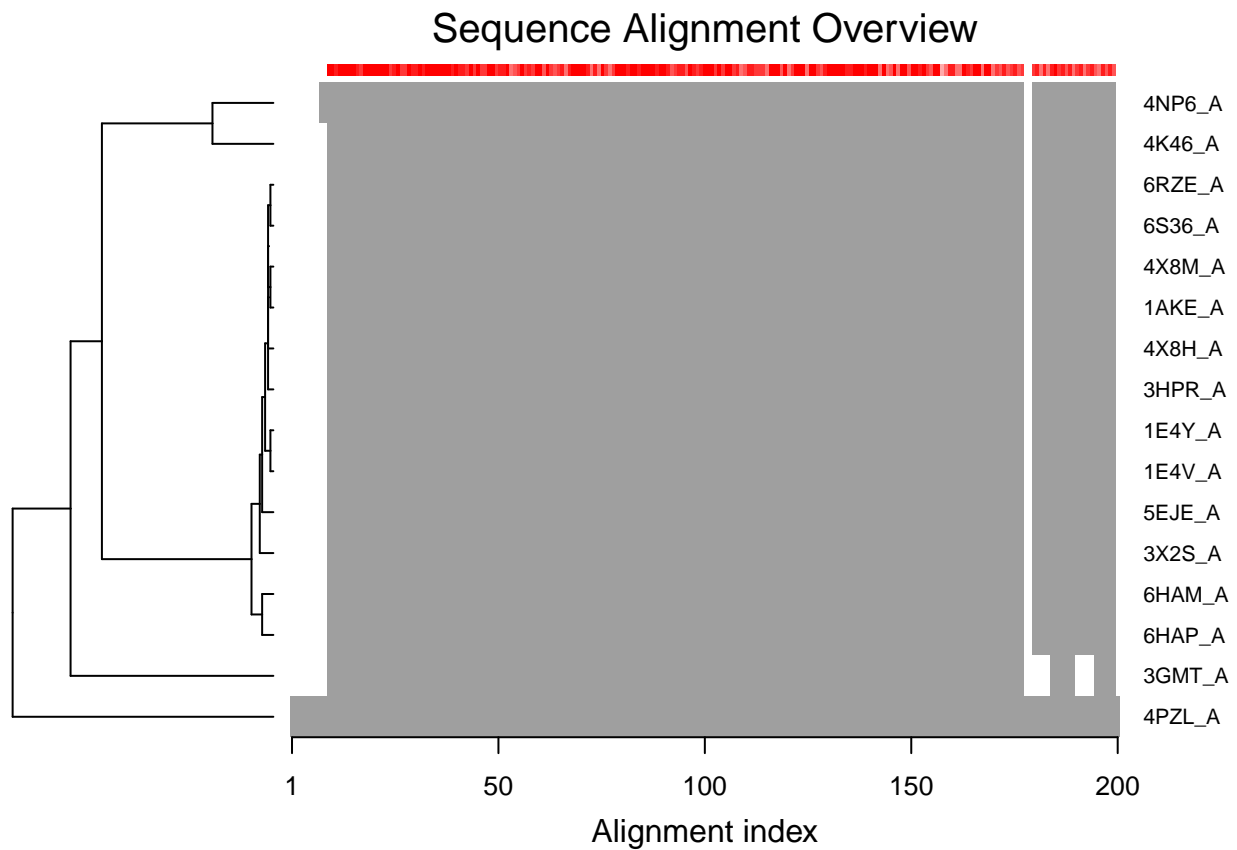
```
## pdb/seq: 12 name: pdbs/split_chain/6HAM_A.pdb
```

```
## 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
```

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

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



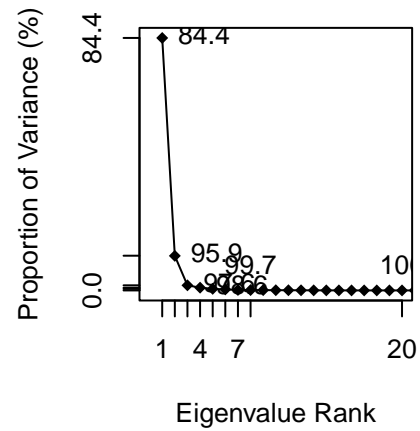
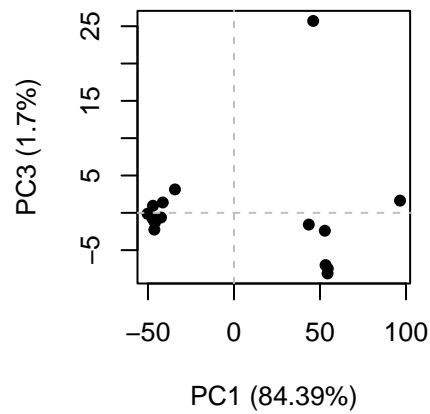
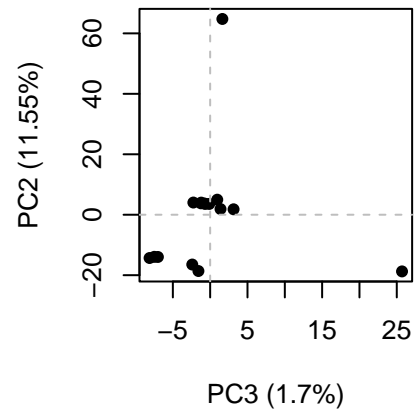
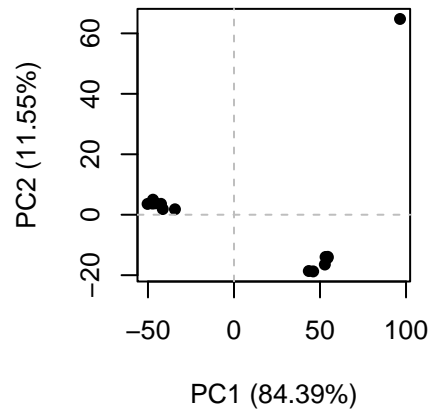
Optional: Viewing our superposed structures

```
library(bio3d.view)
library(rgl)

view.pdb(pdb)
```

Principal component analysis

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

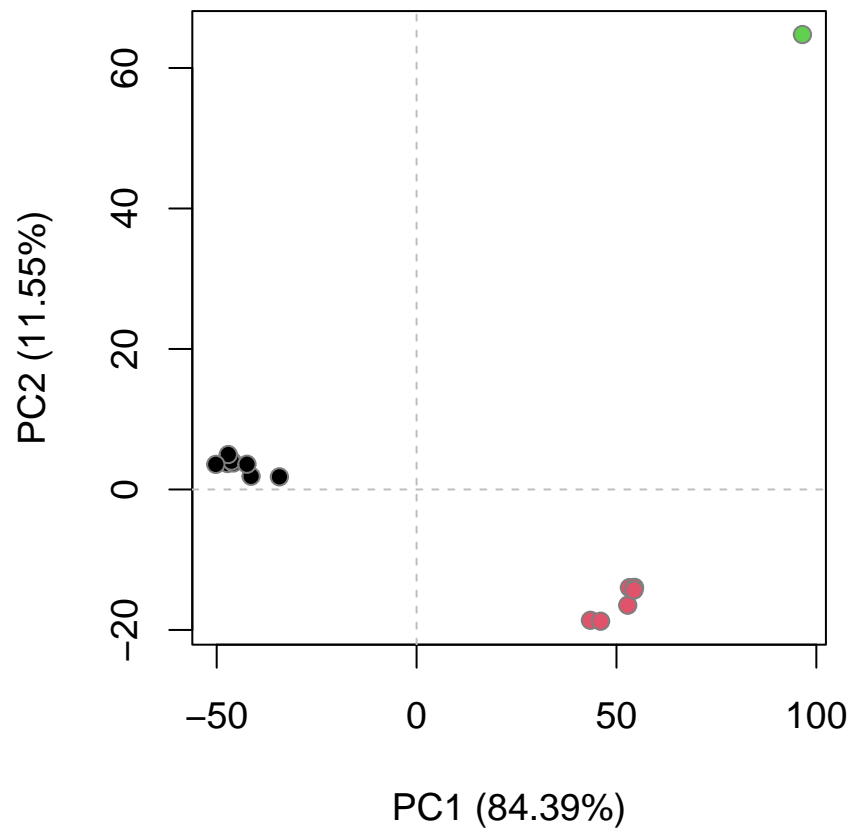


```
# Calculate RMSD
rd <- rmsd(pdb)
```

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

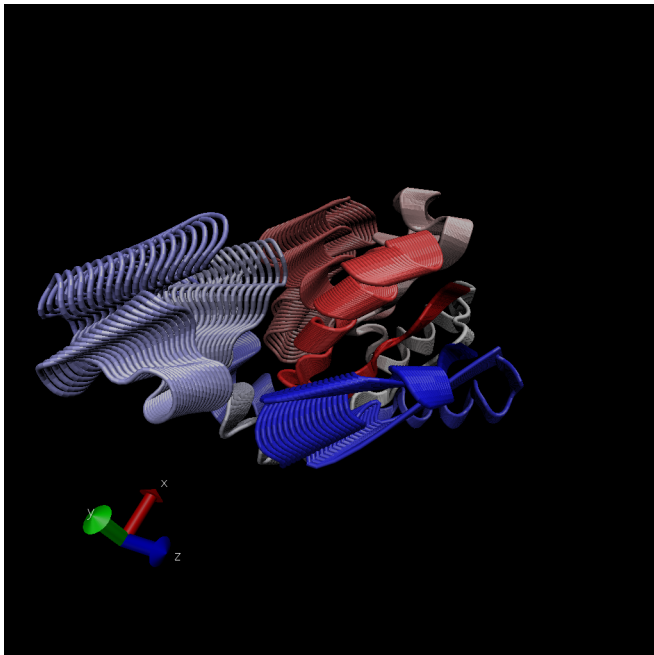
```
# Structure-based clustering
hc.rd <- hclust(dist(rd))
grps.rd <- cutree(hc.rd, k=3)

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



Optional further visualization

```
# 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.connectivity()
```

```
view.xyz(pc1, col=vec2color( rmsf(pc1) ))
```

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

plot our main PCA results with ggplot

```
#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




```
#Normal mode analysis
```

```
# NMA of all structures
```

```
modes <- nma(pdb)
```

```
##
```

```
## 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)
```

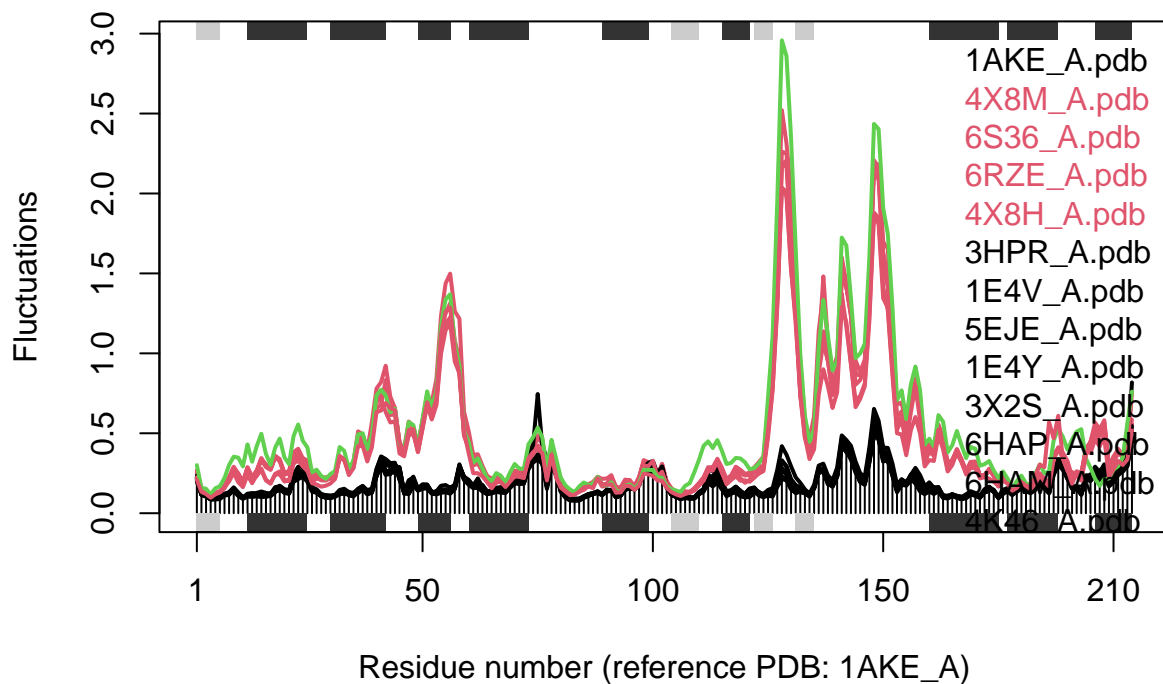
```
## ... estimated memory usage of final 'eNMA' object: 45.4 Mb
```

```
##
```

```
## |
```

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

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



Q14. What do you note about this plot? Are the black and colored lines similar or different? Where do you think they differ most and why?

I note that this plot demonstrates two patterns of fluctuations between colored lines and black lines. Their pattern are different. I think they differ most between about 40 to 60 residues and about 120 to 160 residues,

as these differ by a collective low frequency displacement of two nucleotide-binding site regions that display distinct flexibilities upon nucleotide binding (these two sites demonstrate higher fluctuations or flexibilities compared to the reference, which might indicate potential nucleotide-binding site regions that should be further investigated through experiments). These results collectively suggest the existence of two major distinct conformational states for Adk.