Class9

AUTHOR
Laura Biggs

Q1: What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy? 86.2% of the structures in the PDB are solved by X-ray crystallography while only 6.5% of the structures are visualized by EM.

```
#Read in PDB data
PDB <- read.csv("Data Export Summary.csv", row.names = 1)
print(sapply(PDB, class))
```

```
          X.ray             NMR              EM Multiple.methods
    "character"     "character"     "character"       "integer"
        Neutron           Other           Total
      "integer"       "integer"     "character"
```

```
#Must convert character columns into numeric
PDB$X.ray <- as.numeric(gsub(",","",PDB$X.ray))
PDB$X.ray
```

```
[1] 150342   8866   7911   2510    154     11
```

```
Xray_sum <- sum(PDB$X.ray)

PDB$EM <- as.numeric(gsub(",","",PDB$EM))
PDB$EM
```

```
[1] 8534 1540 2681   74    6    0
```

```
EM_sum <- sum(PDB$EM)

PDB$Total <- as.numeric(gsub(",","",PDB$Total))
PDB$Total
```

```
[1] 171221  10444  10876   4025    191     22
```

```
Total_sum <- sum(PDB$Total)

print(sapply(PDB, class))
```

```
          X.ray             NMR              EM Multiple.methods
      "numeric"     "character"       "numeric"         "integer"
        Neutron           Other           Total
      "integer"       "integer"       "numeric"
```

```
#% Xray
Total_Xray <- (Xray_sum/Total_sum) * 100
print(Total_Xray)
```

```
[1] 86.28665
```

```
#% EM
Total_EM <- (EM_sum/Total_sum) * 100
print(Total_EM)
```

```
[1] 6.522546
```

Q2: What proportion of structures in the PDB are protein? Most of the structures in the PDB are protein at a proportion of .87, or 87%.

```
#Convert character NMR to numeric
PDB$NMR <- as.numeric(gsub(",","",PDB$NMR))
PDB$NMR
```

```
[1] 12053    32   278  1425    31     6
```

```
print(sapply(PDB, class))
```

```
          X.ray             NMR              EM Multiple.methods
      "numeric"       "numeric"       "numeric"         "integer"
        Neutron           Other           Total
      "integer"       "integer"       "numeric"
```

```
NMR_protein <- PDB["Protein (only)", "Total"]
NMR_protein
```

```
[1] 171221
```

```
#Proportion
NMR_protein_proportion <- NMR_protein/Total_sum
NMR_protein_proportion
```

```
[1] 0.8701183
```

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 43,831 HIV-1 protease structures in the PDB.

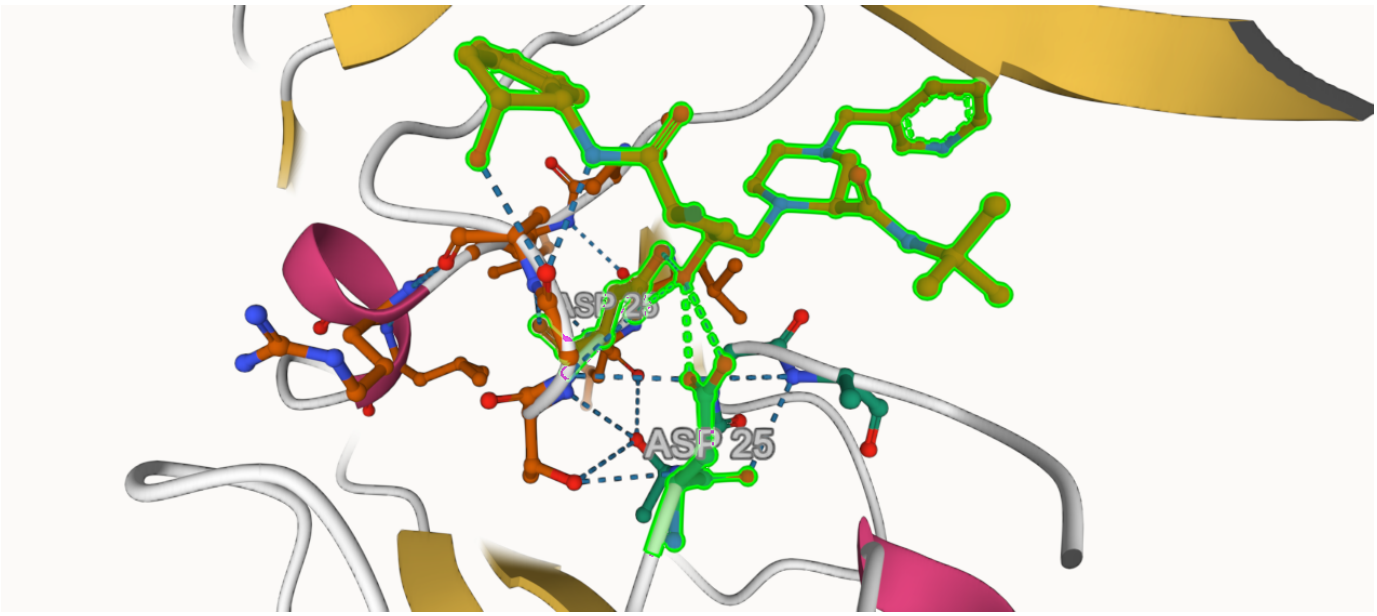
HIV-Pr

Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure? Hydrogen is too small (1.9 angstroms) to be captured by the resolution of this X ray crystallography (2 angstroms). Only the oxygens can be visualized as they are large enough.

Q5: There is a critical “conserved” water molecule in the binding site. Can you identify this water molecule? What residue number does this water molecule have? Water molecule 308 is conserved at the binding site.

Q6: Generate and save a figure clearly showing the two distinct chains of HIV-protease along with the ligand. You might also consider showing the catalytic residues ASP 25 in each chain (we recommend “Ball & Stick” for these side-chains). Add this figure to

your Quarto document.



Bio3D

```
# Load in Bio3D
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.1.3
```

```
# Read in PDB file
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
```

Q7: How many amino acid residues are there in this pdb object? There are 198 amino acid residues.

Q8: Name one of the two non-protein residues? The non-protein residues are the water molecules and MK1, the protease inhibitor drug.

Q9: How many protein chains are in this structure? There are 2 protein chains, chain A and B of the HIV protease.

Inspect pdb further

```
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>
```

Package Setup

```
#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 is a package only available on BioConductor.

Q11. Which of the above packages is not found on BioConductor or CRAN?: 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

Search & retrieve ADK structures

```
library(bio3d)
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? There are 214 amino acids in the sequence

Blast query

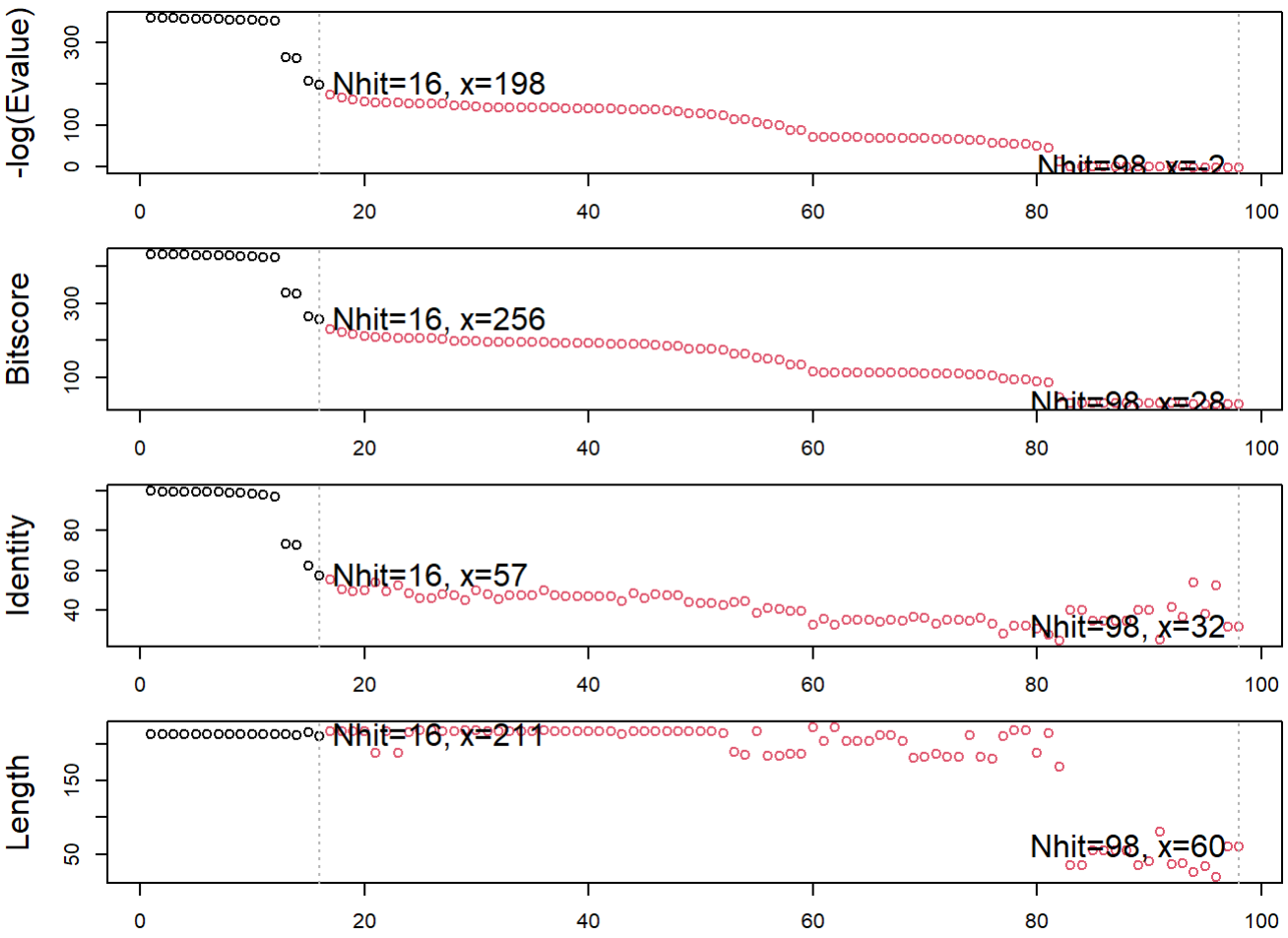
```
b <- blast.pdb(aa)
```

```
 Searching ... please wait (updates every 5 seconds) RID = NH620BTZ013
 ..............................
 Reporting 98 hits
```

```
#hits <- NULL
#hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_
```

```
#Summary of search results
hits <- plot(b)
```

```
 * Possible cutoff values:    197 -3
          Yielding Nhits:    16 98

 * Chosen cutoff value of:    197
          Yielding Nhits:    16
```




```
#'Top hits'
head(hits$pdb.id)
```

```
[1] "1AKE_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A" "3HPR_A"
```

Download hits as 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%
```

Align and superimpose structure data

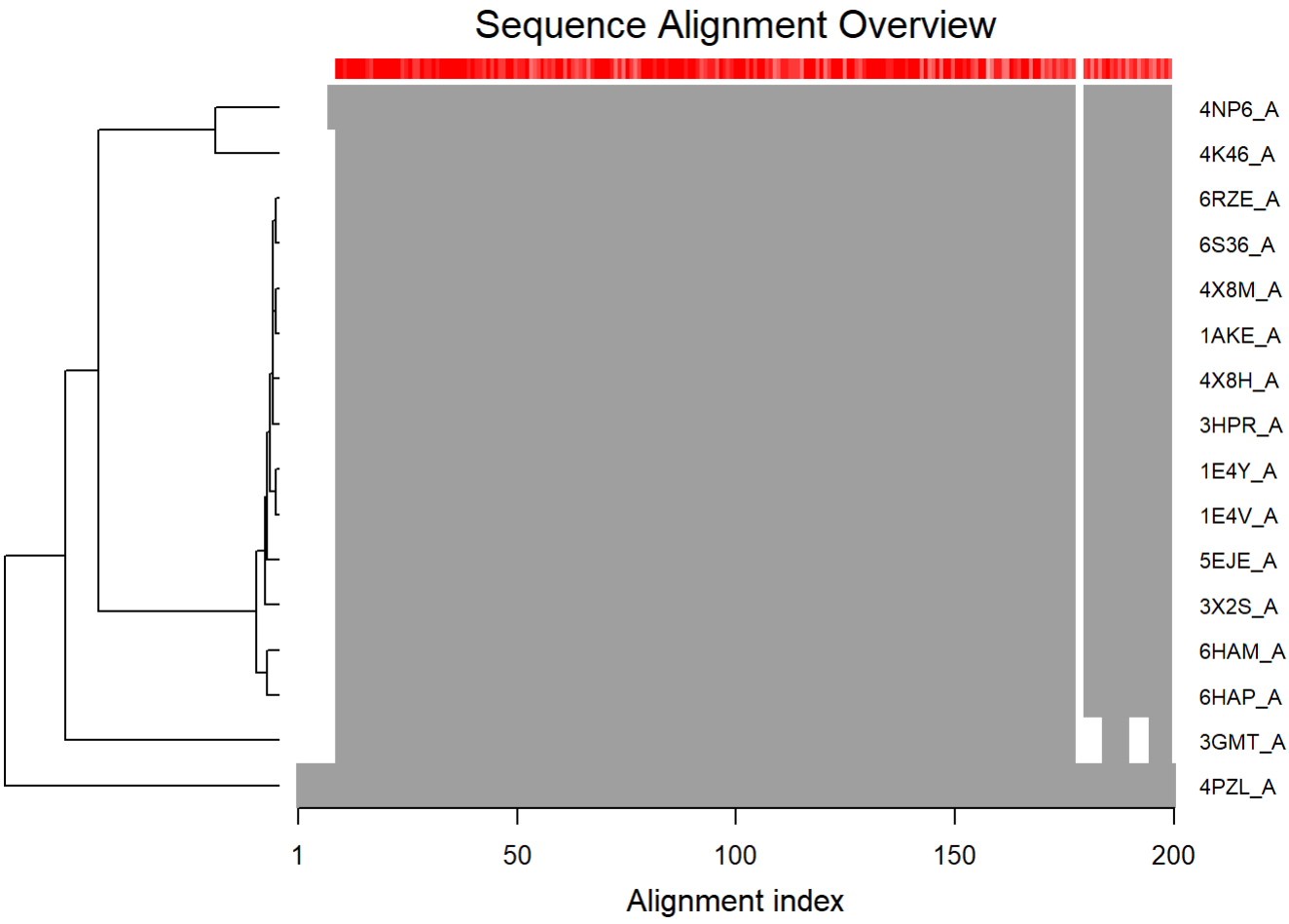
```
# Align related PDBs
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
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
```

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

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



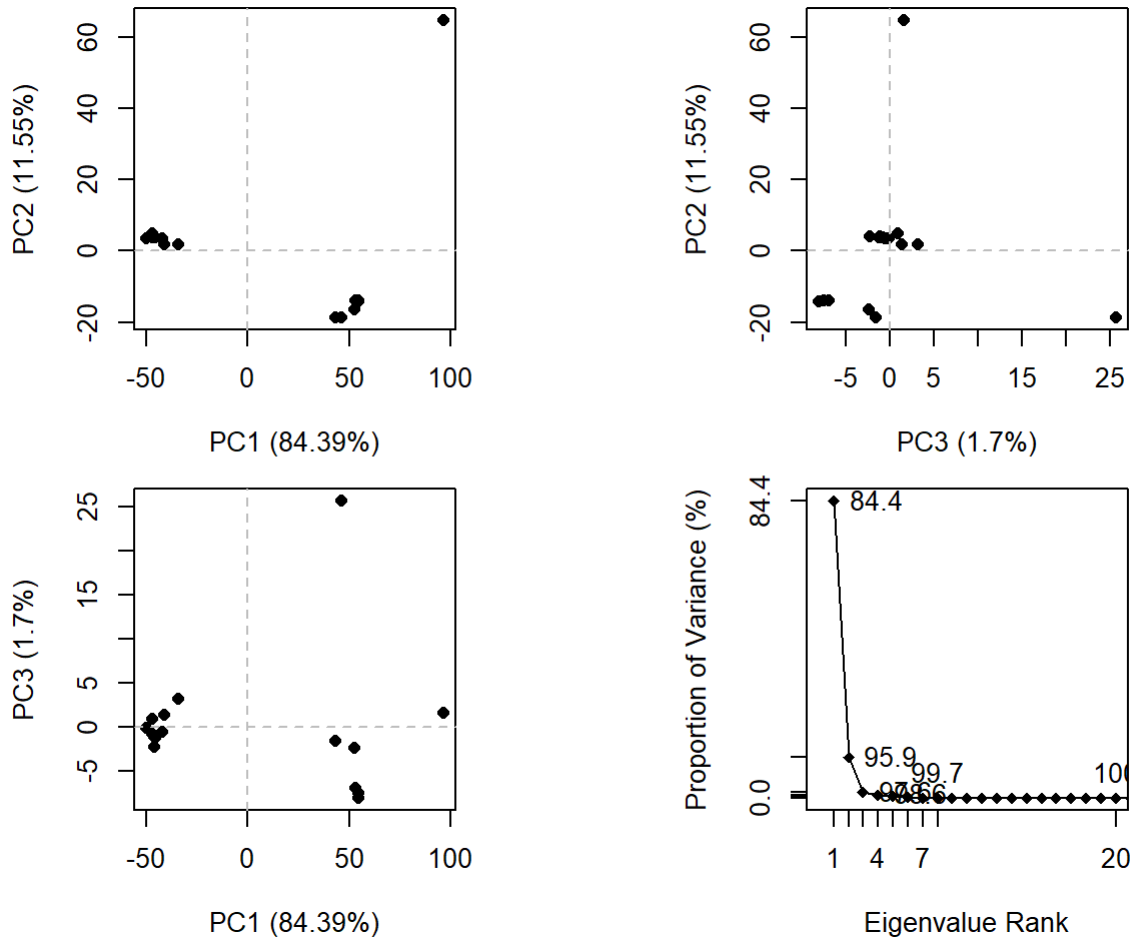
Annotate PDB structures

```
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"
```

PCA

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

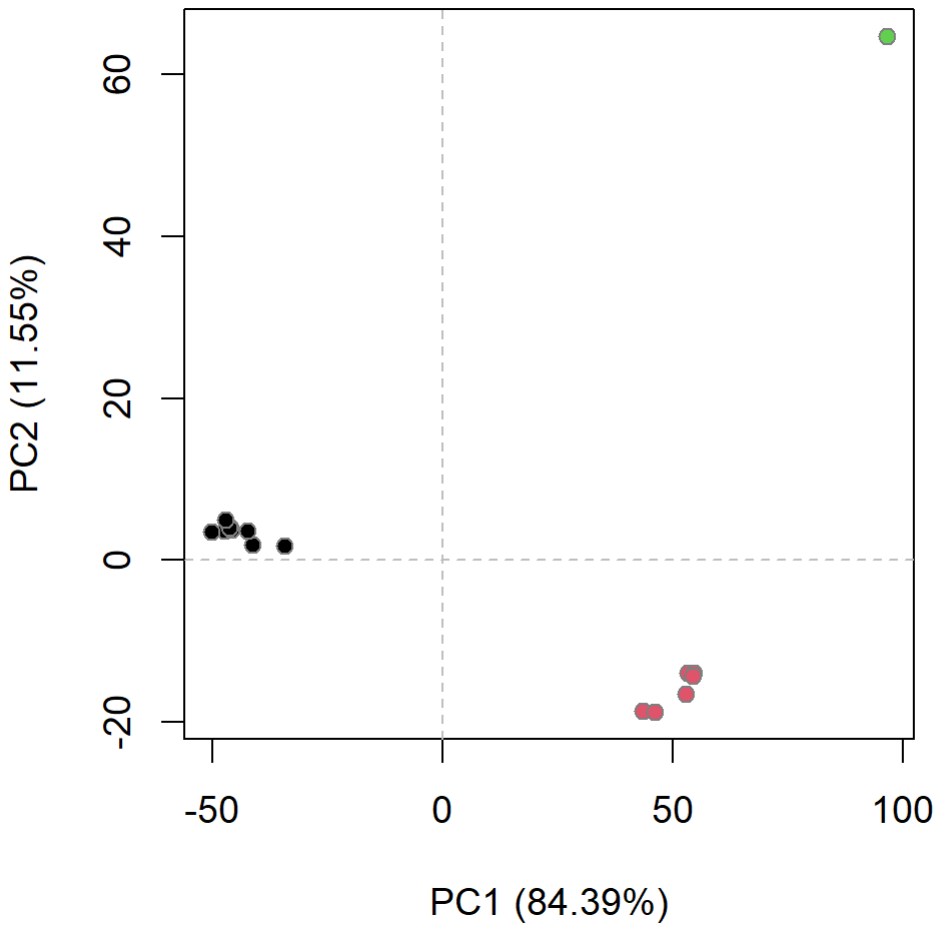


Pairwise RMSD

```
rd <- rmsd(pdbs)
```

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

```
#Hierarchical 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)
```



Alternative ggplot

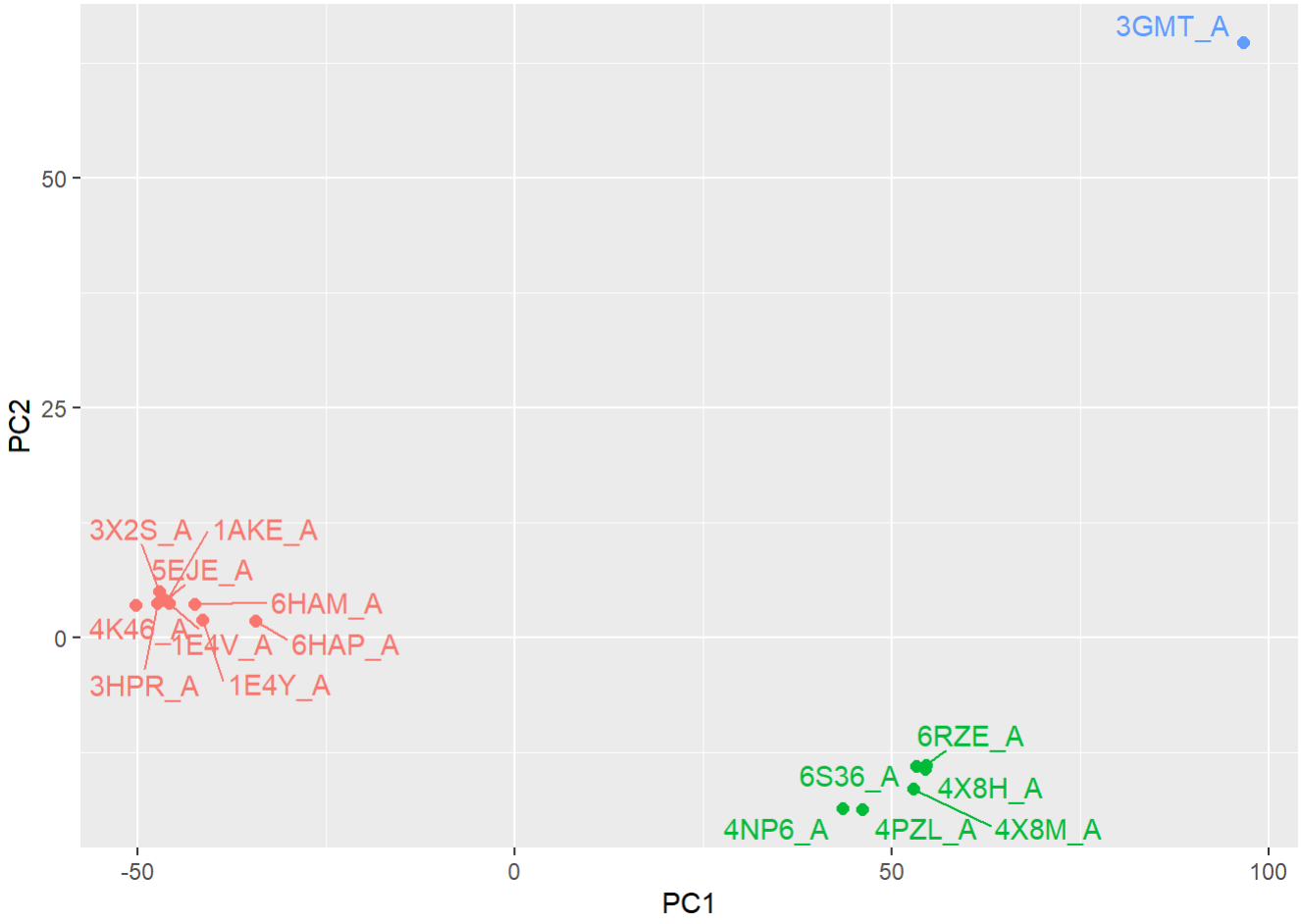
```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(ggrepel)
```

```
Warning: package 'ggrepel' was built under R version 4.1.3
```

```
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

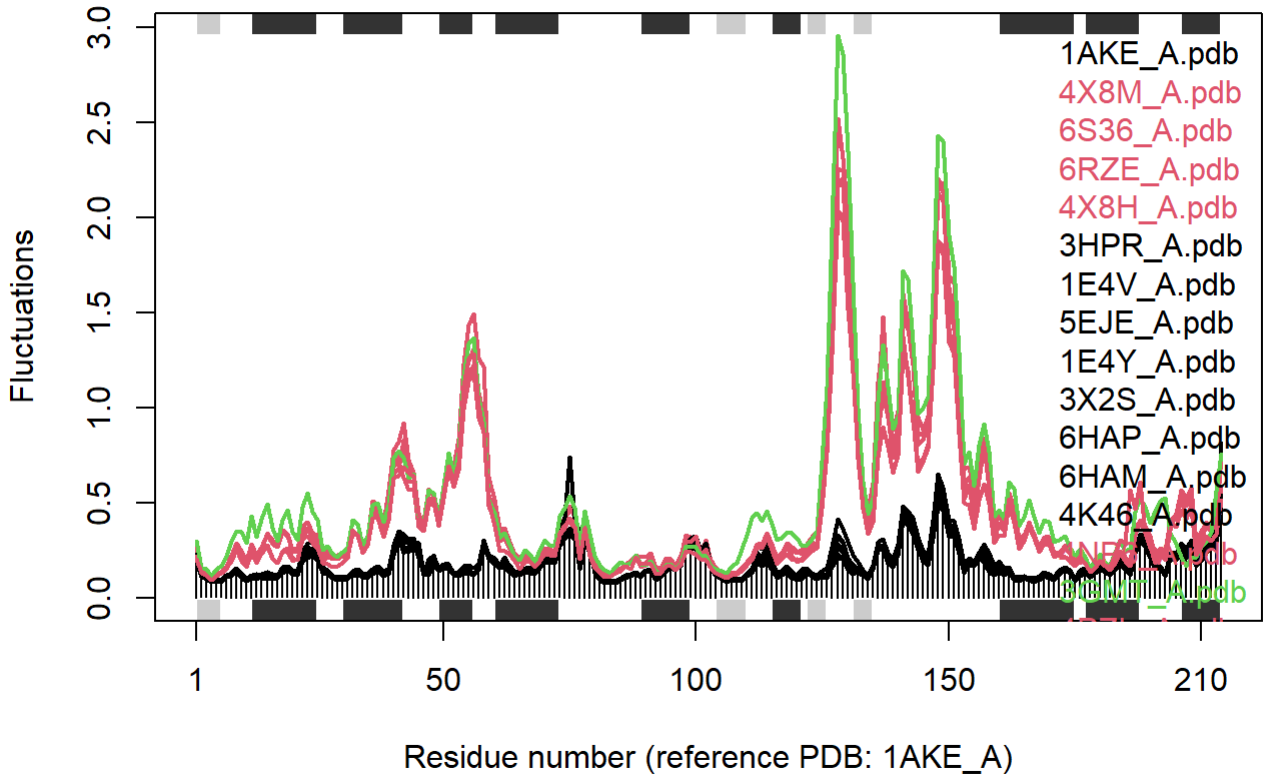
```
modes <- nma(pdbs)
```

```
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%
```

```
56%
  |
  |=========================================                        |
62%
  |
  |============================================                     |
69%
  |
  |================================================                 |
75%
  |
  |====================================================             |
81%
  |
  |========================================================         |
88%
  |
  |============================================================     |
94%
  |
  |================================================================|
100%
```

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

```
Extracting SSE from pdbs$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? The black and colored lines differ in the degree of fluctuation, where colored lines are generally more dynamic. These lines differ most at the functional regions of the protein, ie: where the conformations change.