Class 9: Structural Bioinformatics Pt1

1. Introduction to the RCSD Protein Data Bank (PDB)

rcsb.org > Analyze > PDB Statistics> by Experimental Method and Moelcular Type > getcsv file

```
#Experimental Method and Molecular Type Data
#put in directory then read the file
expmetdata <- "Data Export Summary.csv"
metmol <- read.csv(expmetdata, row.names=1)
metmol
```

```
##                           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
```

Q1: What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy. 92.55757% of structures in the PDB are solved by X-Ray and EM.

```
xray <- sum(metmol$X.ray)
EM <- sum(metmol$EM)
total <-  sum(metmol$Total)
```

```
#Calculation
((xray + EM) / total)*100
```

```
## [1] 92.55757
```

Q2: What proportion of structures in the PDB are protein? In the PDB, proteins make up 0.8726292 of the total strutures proportionally.

```
protein <- metmol[1, "Total"]
protein / total
```

```
## [1] 0.8726292
```

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 4,486 structures of search "HIV" in the PDB. Searching "HIV-1 protease" yields 23,735 structures.

Downloaded the 1hsg.pdb file from PDB, this allows input into another program for 3D visualization of the HIV-1 protein.

2. Visualizing the HIV-1 Protease Structure

VMB was downloaded onto the computer The 1hsg.pdb file was loaded into VMB for visualization of structure protein=lines, red dots=water

Graphics>Representations gives you different Graphical Representation options (drawing method, selected atoms, coloring method)

> Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure? The hydrogen atoms are not represented because there are too many. The water molecules are represented by their oxygen.
>
> 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)? The residue number is H308:0
>
> 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? It seems that the dimers intersect where beta sheets form on both monomers.

3. Introduction to Bio3D in R

```
library(bio3d)
```

```
#With bio3d, can access a pdb file directly by using the PDB identifier
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? Water and "MK1" are the two non-protein residues.

Q9: How many protein chains are in this structure? There are two protein chains in the structure.

```
#Inspect PDB item attributes
attributes(pdb)
```

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

```
#Access the attributes w/ $, i.e. pdb$atom
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>
```

4. Comparative structure analysis of Adenylate Kinase

Goal: PCA on all structures for Adenylate kinase (Adk, transfers phosphaste group ATP/AMP) in PDB. Rxn reqs a "rate limiting conformational transition" Analyze shapes of transitions

```
# can use pca() fn in bio3d to do pca on biomolecular struc data
```

```
#all the programs we need for this structural analysis
library(bio3d)
library(ggplot2)
library(ggrepel)
library(devtools)
```

```
## Loading required package: usethis
```

```
library(BiocManager)
```

```
##
## Attaching package: 'BiocManager'

## The following object is masked from 'package:devtools':
##
##     install
```

```
library(msa)
```

```
## Loading required package: Biostrings

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:bio3d':
##
##     trim
```

```
## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:bio3d':
##
##     mask

## The following object is masked from 'package:base':
##
##     strsplit

##
## Attaching package: 'msa'

## The following object is masked from 'package:BiocManager':
##
##     version
```

```
#library(Grantlab/bio3d-view)
```

Q10. Which of the packages above is found only on BioConductor and not CRAN? The msa package is found only on BioConductor and not CRAN

Q11. Which of the above packages is not found on BioConductor or CRAN? The Grantlab/bio3d-view package is not found on BioConductor or CRAN, it is found on BitBucket.

Q12. True or False? Functions from the devtools package can be used to install packages from GitHub and BitBucket? True, functions from the devtools package can be used to install packages from Github and BitBucket using devtools::install_github() and devtools::install_bitbucket() with the nae of the package as the function input.

Search and Retrieve ADK Structures

get.seq() will fetch a specified query sequence using PDB or UniProt Identifiers blast.pdb() will blast search the pdb database for related structures based on query sequence

```
#We search for the query seq of Chain A of the PDB 1D 1AKE
aa <- get.seq("1ake_A")
```

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

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

```
#output amino acid (query) seq
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
```

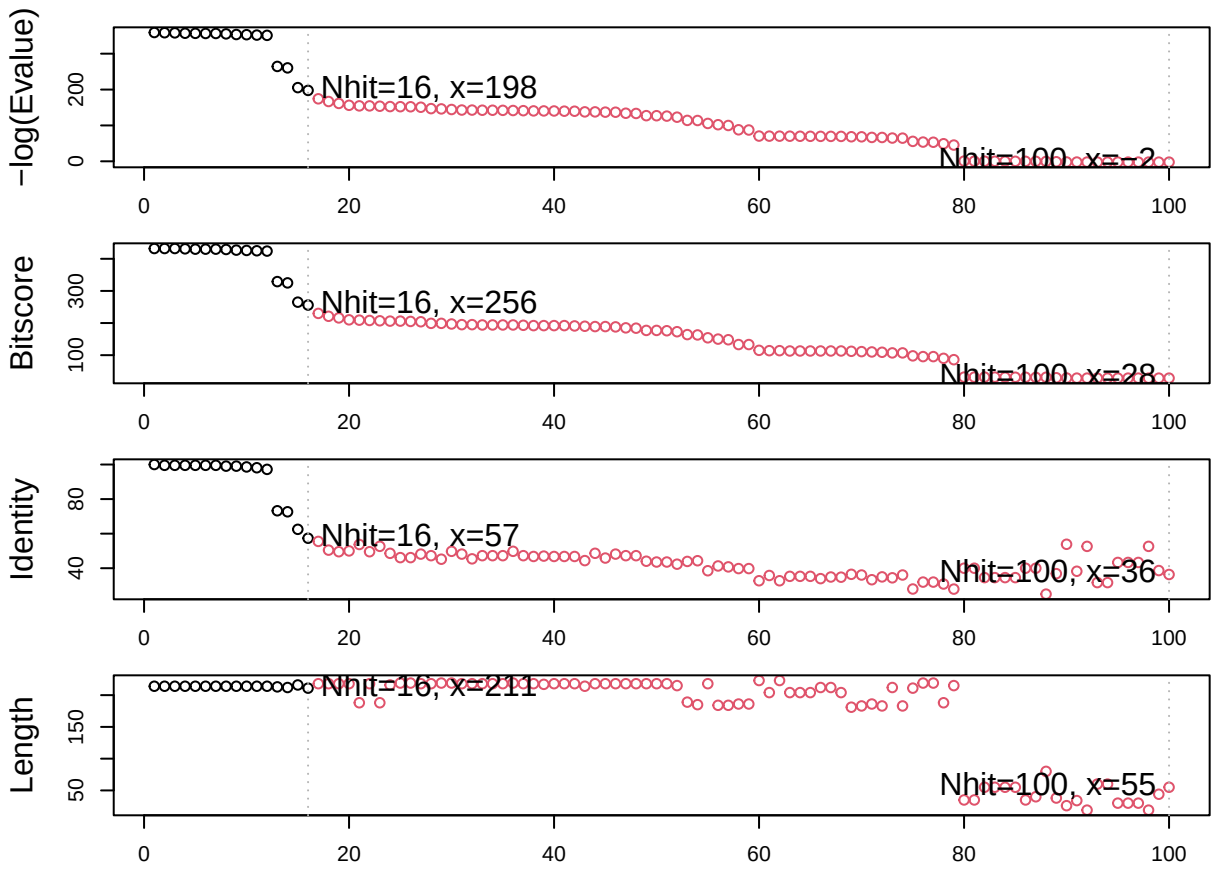
Q13. How many amino acids are in this sequence, i.e. how long is this sequence? There are 214 amino acids in the sequence.

```
#we use the query seq to BLAST search PDB for similar seqs/strucs
b <- blast.pdb(aa)
```

```
##  Searching ... please wait (updates every 5 seconds) RID = 0XZGJB5E013
##  .........................
##  Reporting 100 hits
```

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

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




```
#Listing some of the top hits
head(hits$pdb.id)
```

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

```
#Download 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

##   |                                                                              |
```

Align and Superpose Structures

pdbaln() fn will align and optionally fit/superimpose identified PDB strucs

```
library(muscle)
```

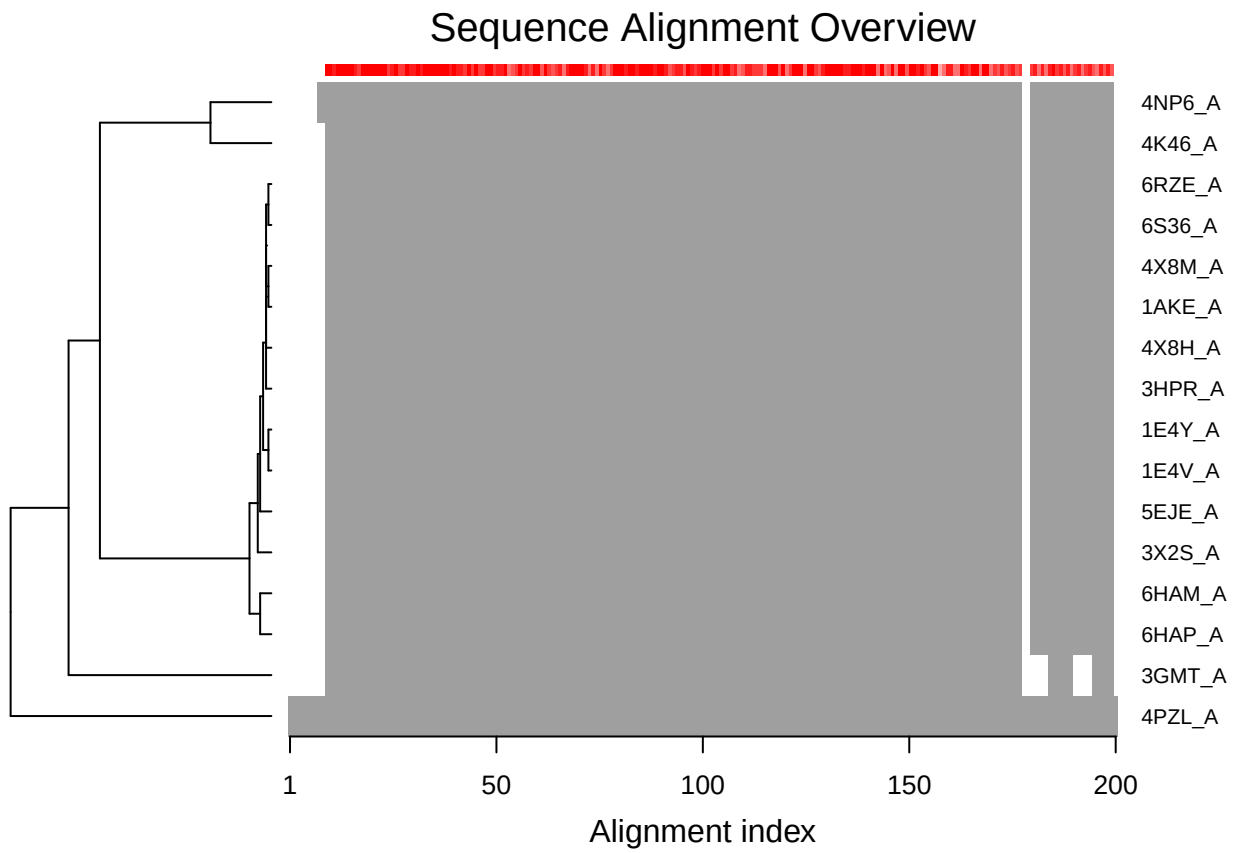
```
library(bio3d)
```

```
# Align releated 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
```

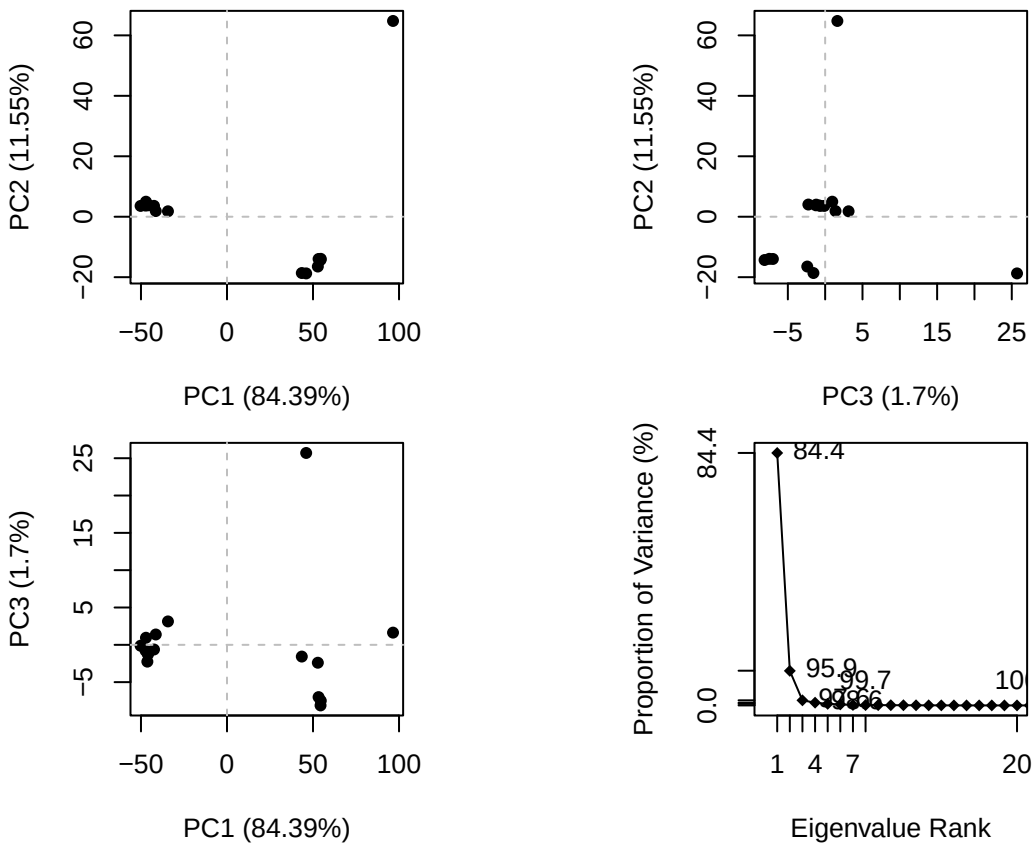
```
# Vector containing PDB codes for figure axis
ids <- basename.pdb(pdbs$id)

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



Principal Component Analysis

pca.xyz() and pca() fns will perform PCA on structural ensemble stored in pdbs obj

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

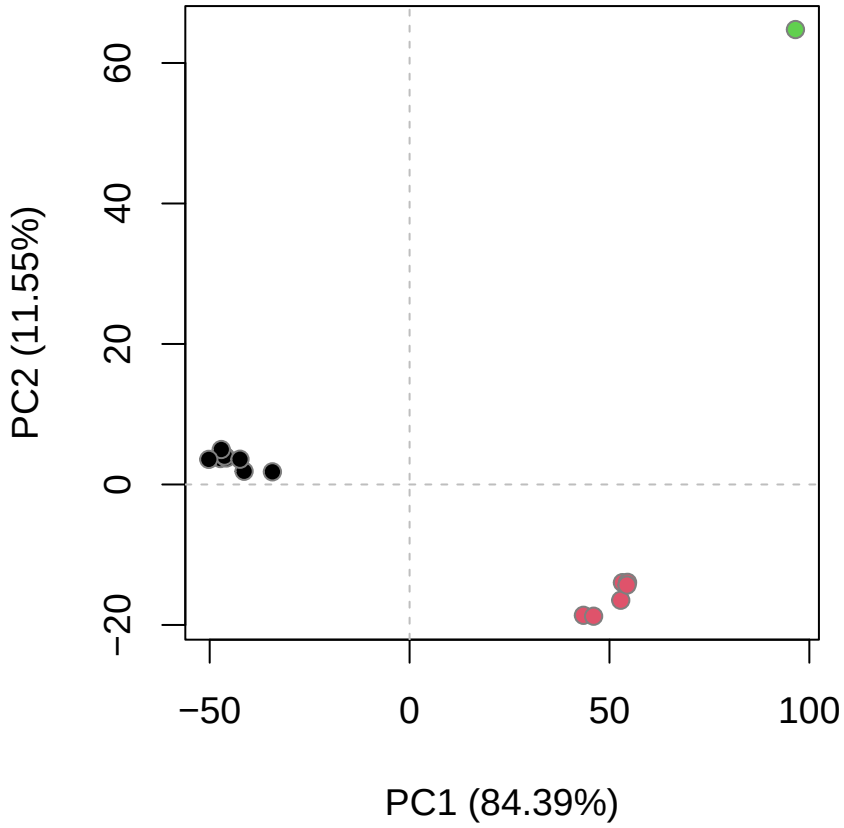


rmsd() calcs all pairwise RMSD values

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

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

```
#Struc-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)
```


PC2 (11.55%)
60
40
20
0
−20
−50
0
50
100
PC1 (84.39%)