class 09 - structural bioinformatics

jack olmstead

PDB stats

We need to import the data!

```
pdb <- read.csv("PDB.csv")

knitr::kable(pdb)
```

Molecular.Type	X.ray	EM	NMR	Multiple.methods	Neutron	Other	Total
Protein (only)	152,809	9,421	12,117	191	72	32	174,642
Protein/Oligosaccharide	9,008	1,654	32	7	1	0	10,702
Protein/NA	8,061	2,944	281	6	0	0	11,292
Nucleic acid (only)	2,602	77	1,433	12	2	1	4,127
Other	163	9	31	0	0	0	203
Oligosaccharide (only)	11	0	6	1	0	4	22

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

The numbers in this csv files are imported as character types. They also have commas in them, so simple coercion isn't possible. Let's write a function to clean these data and sum them.

```
char2num.sum <- function(input.str) {
  return( sum( as.numeric( gsub(",", "", input.str) ) ) )
}
```

```
sum.xr <- char2num.sum(pdb$X.ray)
sum.em <- char2num.sum(pdb$EM)
sum.total <- char2num.sum(pdb$Total)

sum.xr / sum.total * 100
```

```
[1] 85.90264
```

```
sum.em /sum.total * 100
```

```
[1] 7.017832
```

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

```
prot.types <- grep("protein", pdb$Molecular.Type, ignore.case=T)

sum.prot <- char2num.sum(pdb[prot.types,]$Total)

sum.prot / sum.total * 100
```

```
[1] 97.8347
```

> 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?

Searched "HIV" and limited results to Enzyme Classification Name = "Hydrolases". Found 1978 structures.

Molstar format

Here is an Molstar-captured image showing the stabilizing structural elements of an HIV protease inhibitor.

bio3d

Now we're going to use the `bio3d` package for structual informatics.

```
library(bio3d)

# let's fuckin get it
```

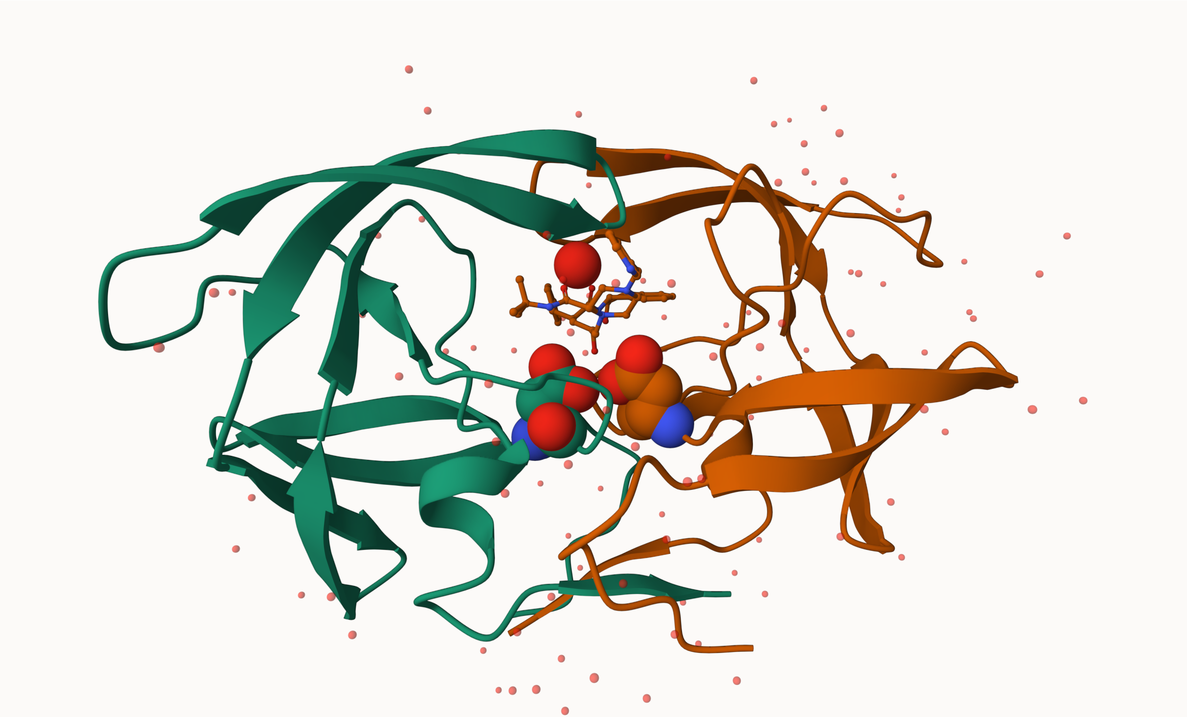


Figure 1: Spacefill model of stabilizing H2O and aspartate residues from PDB: 1HSG

```
p <- read.pdb("1HSG")
```

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

```
p
```

```

 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
```

```
head(p$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?

```
max(p$atom$resno)
```

```
[1] 902
```

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

```
aa321(p$atom$resid[1])
```

```
[1] "P"
```

> Q9: How many protein chains are in this structure?

Let's do a Normal Mode Analysis

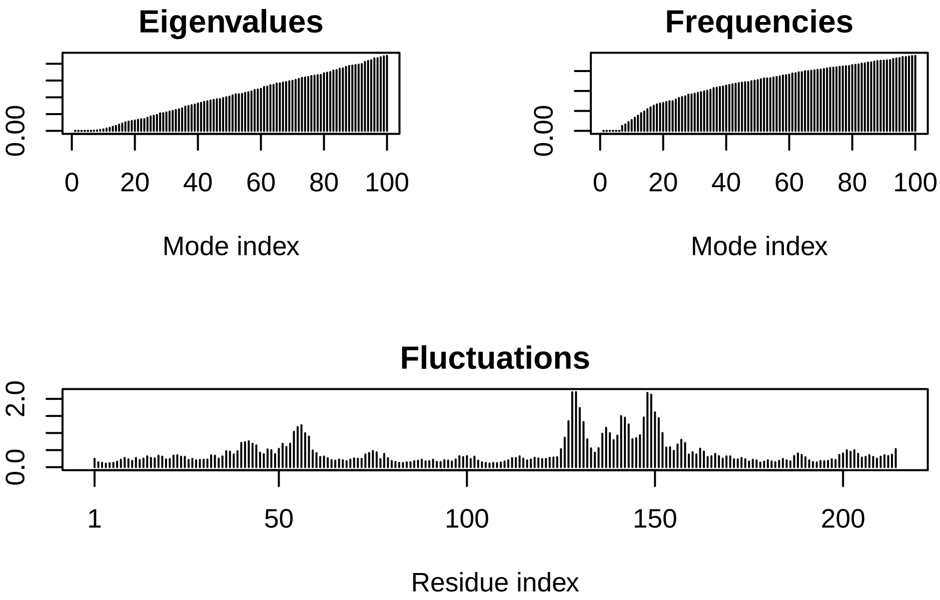
```
# read an input structure
adk <- read.pdb("6s36")
```

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

```
m <- nma(adk)
```

```
 Building Hessian...        Done in 0.056 seconds.
 Diagonalizing Hessian...   Done in 0.515 seconds.
```

```
plot(m)
```



```
# make a trajectory file
mktrj(m, file="adk_m7.pdb")
```

> 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?:

Grantlab/bio3d-view

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

T

PCA of adenylate cyclase

```
library(BiocManager)

# adk_seq <- get.seq("1AKE_a")
# adk_seq
```

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

214

```
# adk_blasts <- blast.pdb(adk_seq)
# hits <- get.pdb(adk_blasts)

# get the hits from hard-coded structures
hits <- NULL
hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A','
```

Now we will download all these structures

```
files <- get.pdb(hits$pdb.id, path="pdbs", split=T, gzip=T)
```

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


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


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


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


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


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


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


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

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

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

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

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

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

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

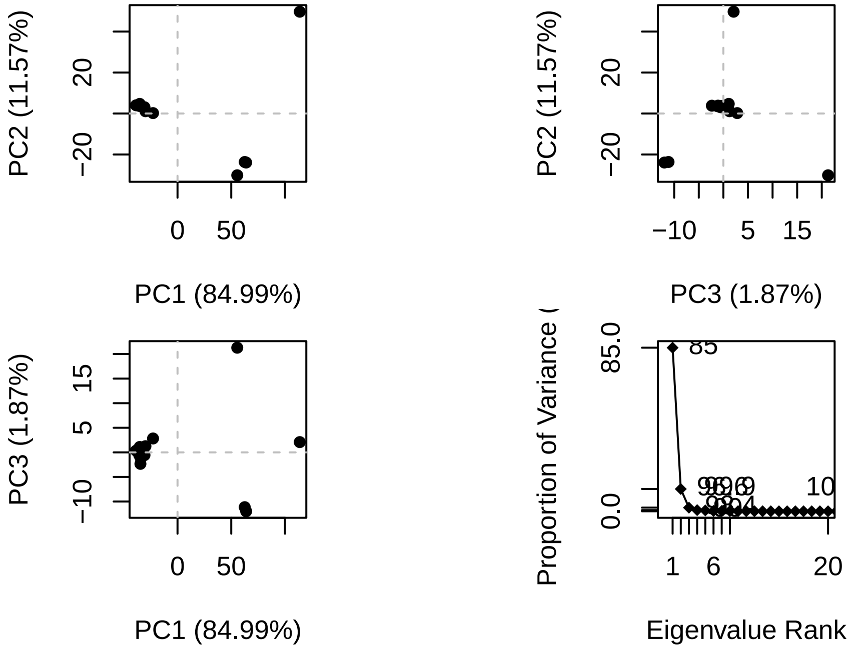
```r
pdbs <- pdbaln(files, fit=T, exefile="msa")
```

```
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
```

Now we will do the PCA

```
pdbs.xray <- pca(pdbs)
plot(pdbs.xray)
```



These 3 PCs correspond to dimensions in space. Let's use our new PCA axes to make a trajectory between different confirmations!

```
mktrj(pdbs.xray, pc=1, file="pc1.pdb")
```