Structural Bioinformatics Lab

David Alvarez

```r
library("tidyverse")
```

```
Warning: package 'tidyverse' was built under R version 4.3.2
```

```
Warning: package 'readr' was built under R version 4.3.2
```

```
Warning: package 'forcats' was built under R version 4.3.2
```

```
Warning: package 'lubridate' was built under R version 4.3.2
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
data_summary <- read_csv("data_summary.csv")
```

```
Rows: 6 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Molecular Type
dbl (3): Multiple methods, Neutron, Other
```

```
num (4): X-ray, EM, NMR, Total

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data_summary
```

```
# A tibble: 6 x 8
  `Molecular Type`   `X-ray`    EM   NMR `Multiple methods` Neutron Other  Total
  <chr>                <dbl> <dbl> <dbl>              <dbl>   <dbl> <dbl>  <dbl>
1 Protein (only)      158844 11759 12296                197      73    32 183201
2 Protein/Oligosacc~    9260  2054    34                  8       1     0  11357
3 Protein/NA            8307  3667   284                  7       0     0  12265
4 Nucleic acid (onl~    2730   113  1467                 13       3     1   4327
5 Other                  164     9    32                  0       0     0    205
6 Oligosaccharide (~      11     0     6                  1       0     4     22
```

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

```
total_structures <- sum(data_summary$"Total")
total_structures
```

```
[1] 211377
```

```
# Number of structures solved by X-ray
xray <- sum(data_summary$"X-ray")

# Number of structures solved by Electron Microscopy
em <- sum(data_summary$"EM")

# Percentage of structures solved by X-Ray and Electron Microscopy
xray_percent <- (xray/total_structures) * 100
em_percent <- (em/total_structures) * 100

xray_percent
```

```
[1] 84.83231
```

```
em_percent
```

```
[1] 8.327301
```

```
# Can be seen that the percentage of structres in the PDB solved by X-Ray and Electron Mic
```

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

```
total_protein <- sum(data_summary[1:3,]$"Total")

# Proportion of structures that are protein

protein_structures <- (total_protein/total_structures)
protein_structures
```

```
[1] 0.9784556
```

```
# The proportion of structures that are proteins in the PDB are about 97%
```

Q3. Type 'HIV' in the PDB website search box and determine how many 'HIV-1' protease structures are in the current PDB?

```
# After doing a quick search, I found there to be 2,767 HIV-1 protease structures in the c
```

Q4. Why do we see just one atom per water molecule in this structure?

```
# We see only one atom per water molecule in this structure because it is a ball and stick
```

Q5. There is a critical "conserved" water molecular in the binding site. Can you identify this water molecule? What residue number does this water molecular have?

```
# I believe the critical "conserved" water molecule I found within the binding site was "H
```

```
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. How many amino acid residues are there in the pdb object?

```r
# There are 99 amino acid residues in this pdb object
```

Q8. Name one of the two non-protein residues

```r
# One of the two non-protein residues is MK1 (1)
```

Q9. How many protein chains are in this structure

```r
# There are two protein chains in this structure, either A or B
```

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

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

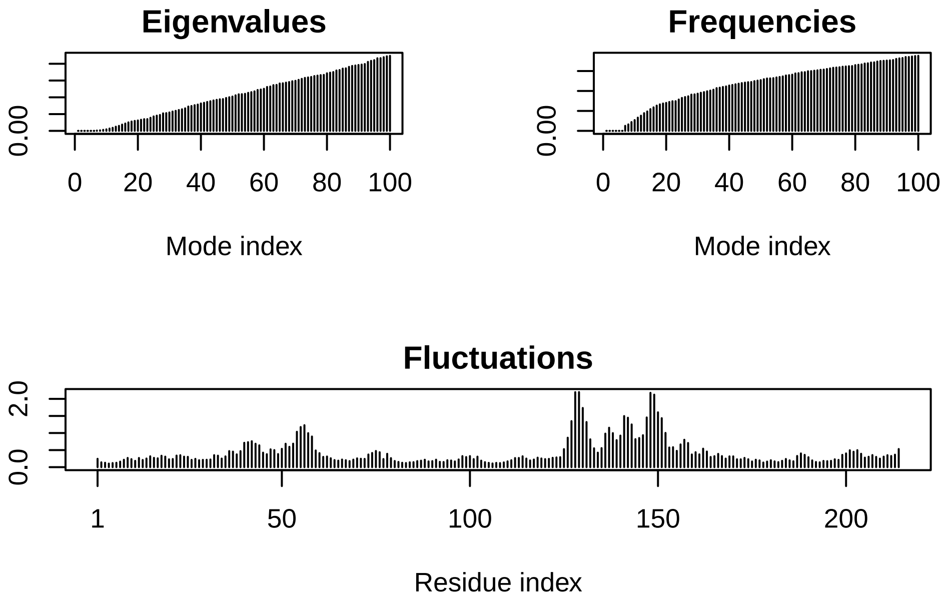
```r
adk
```

```
 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
```

```r
# Performing flexibility predictions
m <- nma(adk)
```

```
 Building Hessian...        Done in 0.01 seconds.
 Diagonalizing Hessian...   Done in 0.3 seconds.
```

```r
plot(m)
```



```r
mktrj(m, file="adk_m7.pdb")
```

```r
library("msa")
```

```
Loading required package: Biostrings

Loading required package: BiocGenerics


Attaching package: 'BiocGenerics'

The following objects are masked from 'package:lubridate':

    intersect, setdiff, union

The following objects are masked from 'package:dplyr':

    combine, intersect, setdiff, union
```

```
The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs


The following objects are masked from 'package:base':

    anyDuplicated, aperm, 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:lubridate':

    second, second<-


The following objects are masked from 'package:dplyr':

    first, rename


The following object is masked from 'package:tidyr':

    expand


The following object is masked from 'package:utils':

    findMatches


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:lubridate':

    %within%

The following objects are masked from 'package:dplyr':

    collapse, desc, slice

The following object is masked from 'package:purrr':

    reduce

The following object is masked from 'package:grDevices':

    windows

Loading required package: XVector


Attaching package: 'XVector'

The following object is masked from 'package:purrr':

    compact

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
```

> 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 which is why we had to prov
```

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

```
# It seems as though the "Grant-lab/bio3d-view" package is found on neither which is why w
```

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

```
# True
```

```
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?

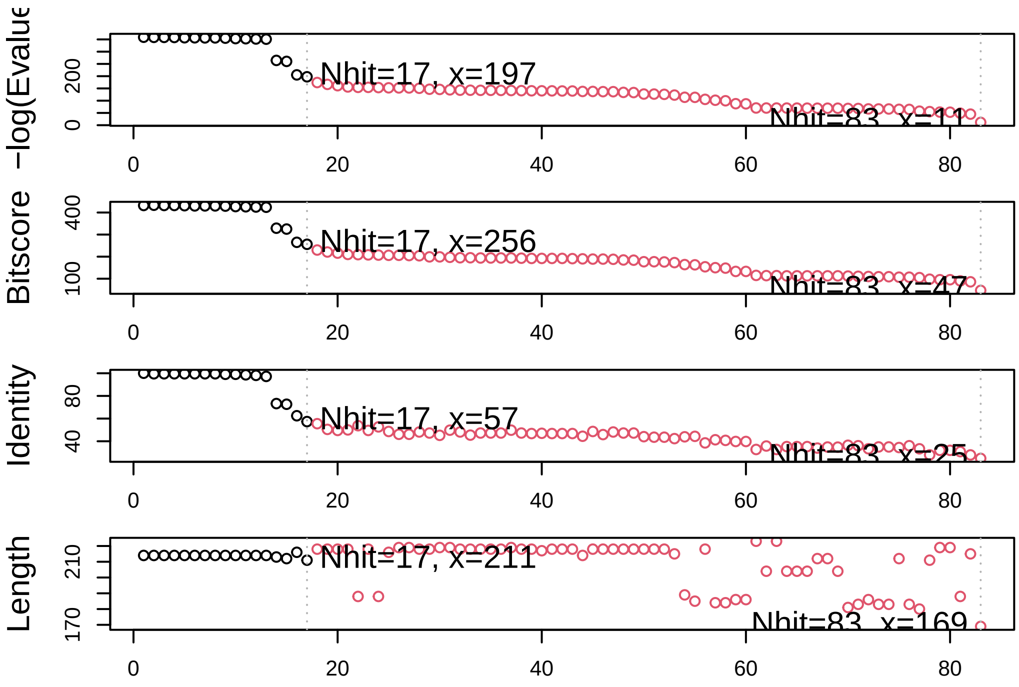
```
# After observing the sequence, it seems there are 214 amino acids.
```

```
# Blast or hmmer search
b <- blast.pdb(aa)
```

```
 Searching ... please wait (updates every 5 seconds) RID = MT1JM0CV013
 .........
 Reporting 83 hits
```

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

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

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



```r
head(hits$pdb.id)
```

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

```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 exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8BQF.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%
|
|============                                                          |  18%
|
|================                                                      |  24%
|
|=====================                                                 |  29%
|
|=========================                                             |  35%
|
|=============================                                         |  41%
|
|=================================                                     |  47%
|
|=====================================                                 |  53%
|
|=========================================                             |  59%
|
|=============================================                         |  65%
|
|=================================================                     |  71%
|
|======================================================                |  76%
|
|==========================================================            |  82%
|
|==============================================================        |  88%
|
|==================================================================    |  94%
|
|======================================================================| 100%
```

```
# Align releated PDBs
# pdbs <- pdbaln(files, fit = TRUE)
# Vector containing PDB codes for figure axis
#ids <- basename.pdb(pdbs$id)

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

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

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

# 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)
```

```
# Visualize first principal component
# pc1 <- mktrj(pc.xray, pc=1, file="pc_1.pdb")
```

```
#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
# p
```

```
# NMA of all structures
# modes <- nma(pdbs)
#plot(modes, pdbs, col=grps.rd)
```

Q14. What do you notice about this plot?

```
# I notice that the green and red lines stay consistent with each other throughout the dur
```