Class 09 Structural Bioinformatics (Pt. 1)

Jiayi Zhou (PID:A17856751)

Table of contents


```
stats <- read.csv("Data Export Summary.csv")
stats
```

```
           Molecular.Type   X.ray     EM    NMR Integrative Multiple.methods
1          Protein (only) 176,378 20,438 12,709         342               221
2 Protein/Oligosaccharide  10,284  3,396     34           8                11
3              Protein/NA   9,007  5,931    287          24                 7
4     Nucleic acid (only)   3,077    200  1,554           2                15
5                   Other     174     13     33           3                 0
6  Oligosaccharide (only)      11      0      6           0                 1
  Neutron Other   Total
1      83    32 210,203
2       1     0  13,734
3       0     0  15,256
4       3     1   4,852
5       0     0     223
6       0     4      22
```

```
stats$Total
```

```
[1] "210,203" "13,734"  "15,256"  "4,852"   "223"     "22"
```

Oh, these are characters no numeric...

```
as.numeric(gsub(",","",stats$Total))
```

```
[1] 210203  13734  15256   4852    223     22
```

```
library(readr)

stats <- read_csv("Data Export Summary.csv")
```

```
Rows: 6 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Molecular Type
dbl (4): Integrative, 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.
```

```
stats
```

```
# A tibble: 6 x 9
  `Molecular Type`    `X-ray`    EM   NMR Integrative `Multiple methods` Neutron
  <chr>                 <dbl> <dbl> <dbl>       <dbl>               <dbl>   <dbl>
1 Protein (only)       176378 20438 12709         342                 221      83
2 Protein/Oligosacch~   10284  3396    34           8                  11       1
3 Protein/NA             9007  5931   287          24                   7       0
4 Nucleic acid (only)    3077   200  1554           2                  15       3
5 Other                   174    13    33           3                   0       0
6 Oligosaccharide (o~      11     0     6           0                   1       0
# i 2 more variables: Other <dbl>, Total <dbl>
```

```
n.total <- sum(stats$Total)
n.total
```

```
[1] 244290
```

Q1: What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy. Give your answer to 2 significant figures.

```
n.xray <- sum(stats$`X-ray`)
round(n.xray/ n.total*100)
```

```
[1] 81
```

```
n.xray <- sum(stats$`X-ray`)
percent.xray <- n.xray / n.total * 100
percent.xray
```

```
[1] 81.43231
```

81.43% of structures in the PDB are solved by X-Ray and Electron Microscopy.

There are 81.43 percent Xray structures in the PDB

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

```
round(stats$Total[1]/n.total*100, 2)
```

```
[1] 86.05
```

```
library(readr)

stats <- read_csv("Data Export Summary.csv")
stats
```

```
# A tibble: 6 x 9
  `Molecular Type`    `X-ray`    EM   NMR Integrative `Multiple methods` Neutron
  <chr>                 <dbl> <dbl> <dbl>       <dbl>               <dbl>   <dbl>
1 Protein (only)       176378 20438 12709         342                 221      83
2 Protein/Oligosacch~   10284  3396    34           8                  11       1
3 Protein/NA             9007  5931   287          24                   7       0
4 Nucleic acid (only)    3077   200  1554           2                  15       3
5 Other                   174    13    33           3                   0       0
6 Oligosaccharide (o~      11     0     6           0                   1       0
# i 2 more variables: Other <dbl>, Total <dbl>
```

86.05 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?

Current HIV-1 protease structures in the PDB: 873

Exploring PDB Structures

Package for structual bioinfomatics

```
library(bio3d)

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

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

```
hiv
```

```
 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
```

Let's first use the Mol* viewer to explore this structure.

Figure 1: My first view of HIV-Pr

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

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/hw/mb7lcvrd0v17c4nnmhd27vr40000gn/T//RtmplMpHUG/1hsg.pdb exists.
Skipping download
```

```
hiv
```

```
 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
```

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

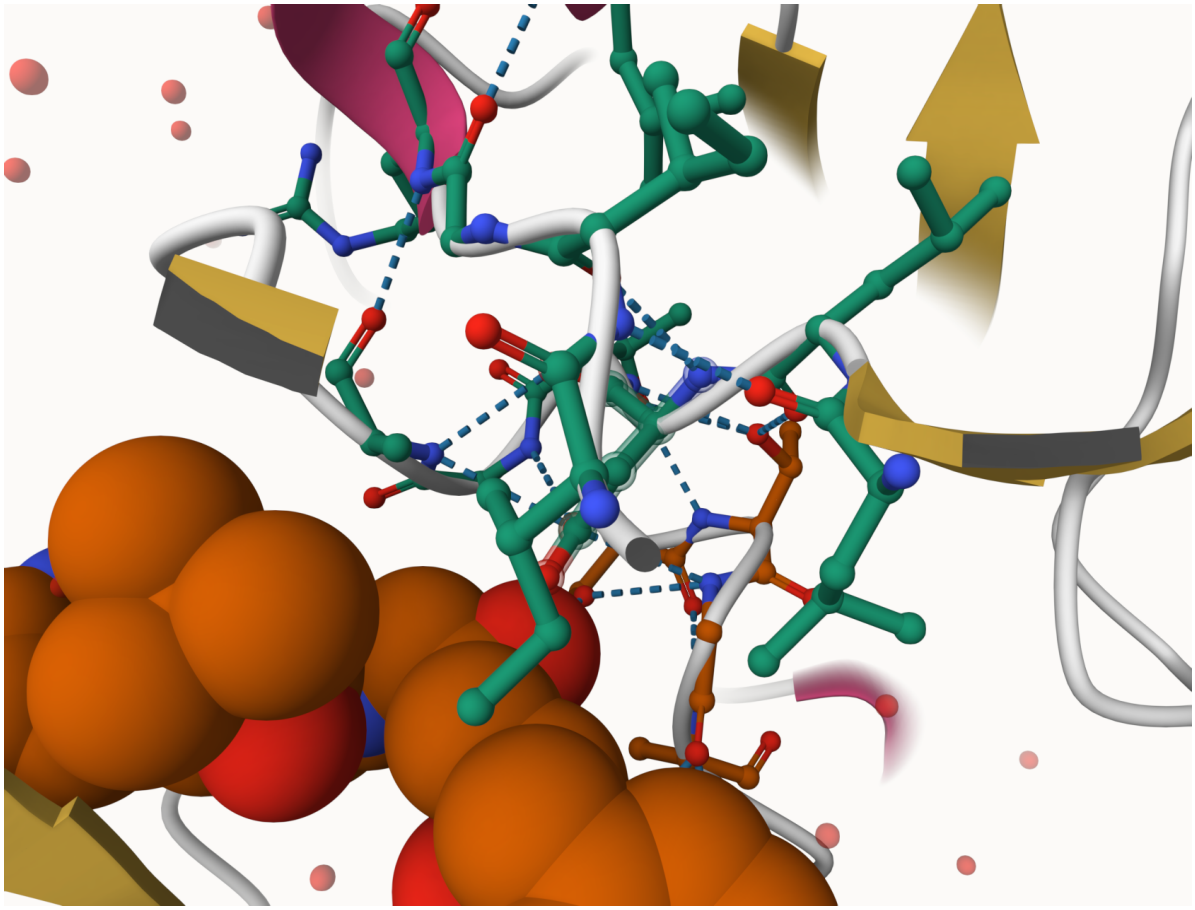
In X-ray crystal structures like HIV-1 protease (PDB: 1HSG), only the oxygen atom is visible and recorded in the PDB file because hydrogen atoms are not detected by X-ray crystallography, so each water appears as a single atom in the structure.

> 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?

HOH 308 could be a conserved water molecule in the HIV-1 protease binding site because it is positioned closest to both the ligand and the catalytic Asp 25 residues, which is characteristic of the functionally important conserved water in this enzyme.

> 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 and the critical water (we recommend "Ball & Stick" for these side-chains). Add this figure to your Quarto document.

And a view of the ligand (ball and stick) with catalytic ASP 25 amino-acids (spacefill) and the all important active site water molecule (spacefill):



Indinavir and larger ligands or substrates can enter the HIV-1 protease binding site through the opening of flexible flaps formed by segments of each chain, which move apart to expose the catalytic site and then close to secure the ligand for catalysis or inhibition.

> Q7. 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 can you identify

secondary structure elements that are likely to only form in the dimer rather than the monomer?

In HIV protease, -sheets at the dimer interface are secondary structure elements that are likely to only form in the dimer and not in the monomer, as they are stabilized by interactions between the two identical chains visible in the graphic display

PDB objects in R

```
library(bio3d)
```

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

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/hw/mb7lcvrd0v17c4nnmhd27vr40000gn/T//RtmplMpHUG/1hsg.pdb exists.
Skipping download
```

```
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?

198 residues.

```
sum(pdb$calpha)
```

```
[1] 198
```

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

HOH (water) or MK1 (indinavir).

```
unique(pdb$atom$resid[pdb$atom$type == "HETATM"])
```

```
[1] "MK1" "HOH"
```

Q9: How many protein chains are in this structure?

2 chains (A and B).

```
unique(pdb$atom$chain[pdb$atom$type == "ATOM"])
```

```
[1] "A" "B"
```

```
length(unique(pdb$atom$chain[pdb$atom$type == "ATOM"]))
```

```
[1] 2
```

```
head(hiv$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>
```

Extract the sequence

```
pdbseq(hiv)
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
"P" "Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K"
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
"E" "A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G"
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
"R" "W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D"
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
"Q" "I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T"
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99   1
"P" "V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F" "P"
  2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
"Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K" "E"
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
"A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G" "R"
 42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
"W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D" "Q"
 62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81
"I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T" "P"
 82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99
"V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F"
```

```
chainA_seq <- pdbseq(trim.pdb(hiv, chain="A"))
```

```
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>
```

I can interactively view these PDB objects in R with the new **bio3dview** package. This is not yet on CRAN.

To install this I can setup install **bio3dview pak** package and use it to run from Github. In my console I first run

```
#install.packages("pak")
pak::pak("bioboot/bio3dview")
```

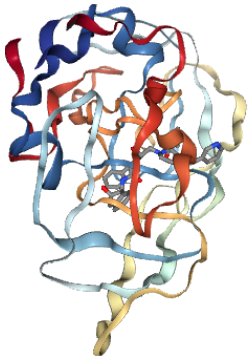
```
! Using bundled GitHub PAT. Please add your own PAT using `gitcreds::gitcreds_set()`.

  Loading metadata database

  Loading metadata database ... done




  No downloads are needed

  1 pkg + 40 deps: kept 40 [7s]
```

```
#install.packages("NGLVieweR")
```

```
library(bio3dview)

view.pdb(hiv)
```

```
PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installe
```

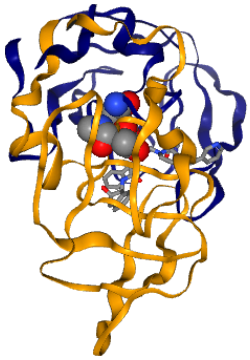
```
file:////private/var/folders/hw/mb7lcvrd0v17c4nnmhd27vr40000gn/T/RtmplMpHUG/file2de02886d17a
```



Change some settings

```
# Select the important ASP 25 residue
sele <- atom.select(pdb, resno=25)

# and highlight them in spacefill representation
view.pdb(hiv, highlight = sele,
         highlight.style = "spacefill",
         colorScheme = "chain",
         col=c("navy", "orange"),
         backgroundColor = "lightyellow")
```

file:////private/var/folders/hw/mb7lcvrd0v17c4nnmhd27vr40000gn/T/RtmplMpHUG/file2de059dc2e8a/



Predict Protein Flexibility

We can run a bioinformatic calculation to predict protein dynamics - i.e. functional motions.

We will use the `nma()` function:

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

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

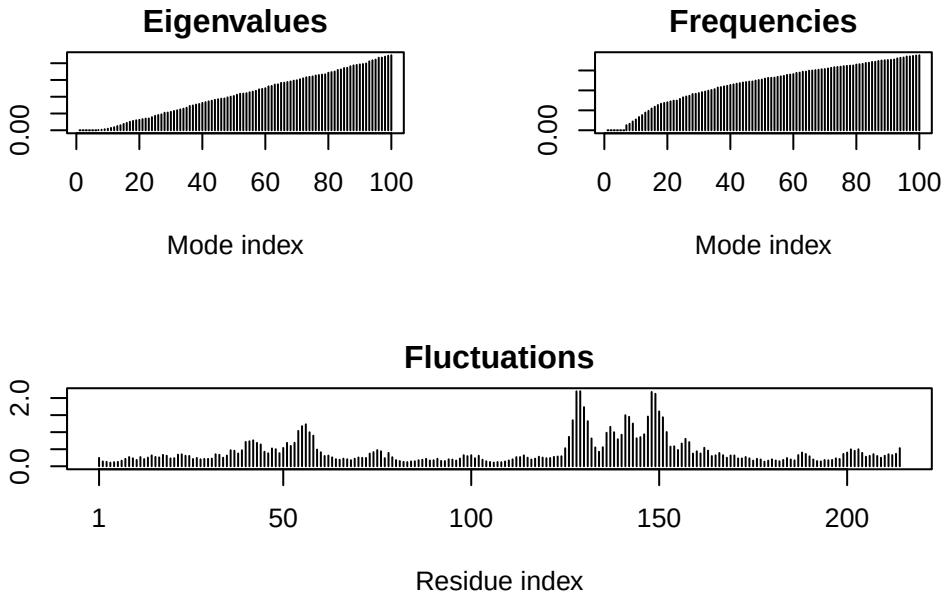
```
adk
```

```
 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
```

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

```
 Building Hessian...        Done in 0.022 seconds.
 Diagonalizing Hessian...   Done in 0.417 seconds.
```

```
plot(m)
```

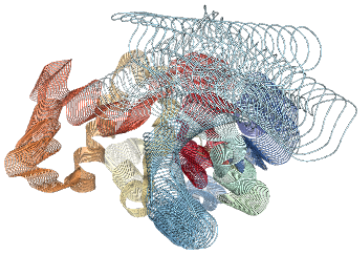


Generate a "trrajectory" of predicted motion

```
mktrj(m, file="ADK_nma.pdb")
```

```
view.nma(m)
```

```
file:////private/var/folders/hw/mb7lcvrd0v17c4nnmhd27vr40000gn/T/RtmplMpHUG/file2de03e888b79/
```



Comparative structure analysis of Adenylate Kinase

```
# Install packages in the R console NOT your Rmd/Quarto file

#install.packages("bio3d")
#install.packages("NGLVieweR")

#install.packages("pak")
#pak::pak("bioboot/bio3dview")

#install.packages("BiocManager")
#BiocManager::install("msa")
```

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

msa. It's a Bioconductor package (installed via `BiocManager::install("msa")`), not on CRAN.

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

bio3dview. It's GitHub-only (`pak::pak("bioboot/bio3dview")`), not on CRAN or Bioconductor.

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

True. pak can install from GitHub and Bitbucket using remote specs (e.g., "user/repo" for GitHub, `bitbucket::user/repo`).

Search and 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?

214 amino acids.