

# BIMM 143 Lab 9

Ashley Martinez (PID: A17891957)

## Table of contents

PDB Statistics . . . . .	1
Bio3D in R . . . . .	7
Predict protein flexibility . . . . .	9
Comparative Structure Analysis . . . . .	11

## PDB Statistics

The main database for structural biology is called the PDB. Let's have a look at what it contains: Let's first load the PDB CSV file, while also using 'readr'.

```
library(readr)
stats<-read.csv("PDB.CSV")
stats
```

	Molecular.Type	X.ray	EM	NMR	Integrative	Multiple.methods
1	Protein (only)	176,204	20,299	12,708	342	218
2	Protein/Oligosaccharide	10,279	3,385	34	8	11
3	Protein/NA	9,007	5,897	287	24	7
4	Nucleic acid (only)	3,066	200	1,553	2	15
5	Other	173	13	33	3	0
6	Oligosaccharide (only)	11	0	6	0	1
	Neutron	Other	Total			
1	83	32	209,886			
2	1	0	13,718			
3	0	0	15,222			
4	3	1	4,840			
5	0	0	222			
6	0	4	22			

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

**Answer:** 81.48%

This code will help us convert character values to numeric.

```
stats$Total <- as.numeric(gsub(",", "", stats$Total))
stats$X.ray <- as.numeric(gsub(",", "", stats$X.ray))
```

Now we can treat the values as numbers to calculate the totals for the wanted columns.

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

```
[1] 243910
```

```
n.xray<-sum(stats$X.ray)
n.xray
```

```
[1] 198740
```

```
percent.xray<-n.xray/ n.total*100
percent.xray
```

```
[1] 81.48087
```

There are 'r round(percent.xray,2)' percent X-ray structures in the PDB.

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

**Answer:** 86.05%

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

```
[1] 86.05
```

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?

**Answer:** 2,406

```
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
```

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

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/wt/ybcqjt6s597bdw5zz7qg3klm0000gn/T//RtmpaQI6I2/1hsg.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
```

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

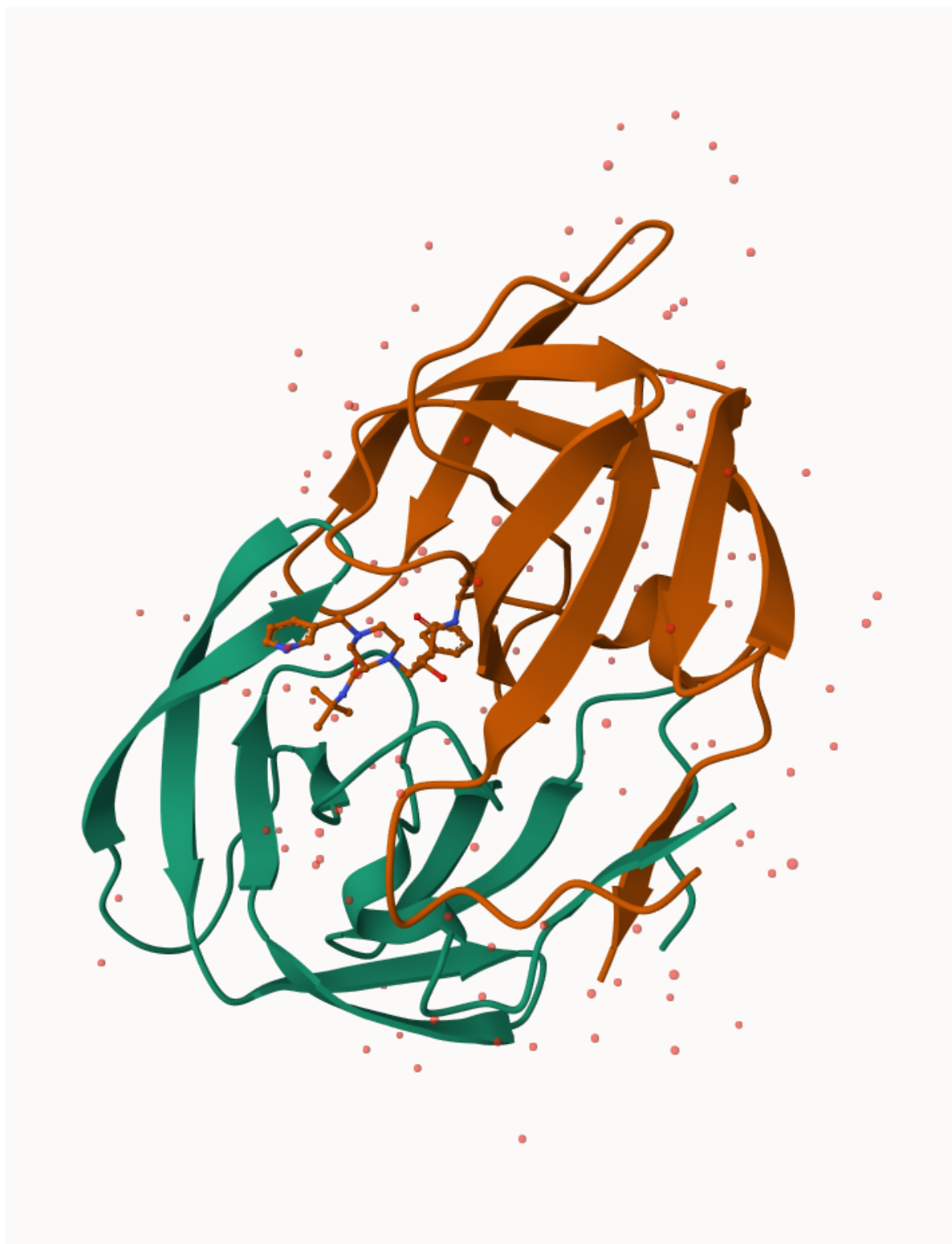


Figure 1: My first view of HIV=Pr

Q6: Now let's view the molecule with labeled Asp 25 and water.

Ligands can enter the binding site when the active site flaps are open allowing the ligand to enter and bind.



Figure 2: View of labeled HSG molecule

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

**Answer:** This is because proteins are analyzed with X-ray crystallography, which does not show hydrogen atoms so the water molecules appear as a single oxygen.

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.

**Answer:** HOH 308

## Bio3D 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/wt/ybcqjt6s597bdw5zz7qg3klm0000gn/T//RtmpaQI6I2/1hsg.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?

**Answer:** 198

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

**Answer:** H2O

Q9: How many protein chains are in this structure?

**Answer:** 2 chains

```
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>

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

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 set up **pak** package and use it to install **bio3dview** from GitHub

```
# 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 39 [4.8s]
```

This chunk of code will render a spinning image however it can not be printed onto a pdf.

```
library(bio3dview) library(NGLVieweR)
```

```
view.pdb(pdb) |> setSpin()
```

```
sele <- atom.select(pdb, resno=25) view.pdb(pdb, cols=c("navy","teal"), highlight = sele,  
highlight.style = "spacefill")
```

## Predict protein flexibility

We can run a bioinformatics 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:

MRILLGAPGAGKGTQAQFIMEKYGIPQISTGDMRLRAAVKSGSELGKQAKDIMDAGKLV  
DELVIALVKERIAQEDCRNGFLDGFRTIPQADAMKEAGINVDYVLEFDVPDELIVDKI  
VGRRVHAPSGRVYHVKFNPVKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG  
YYSKEAEAGNTKYAKVDGTPVAEVRADLEKILG

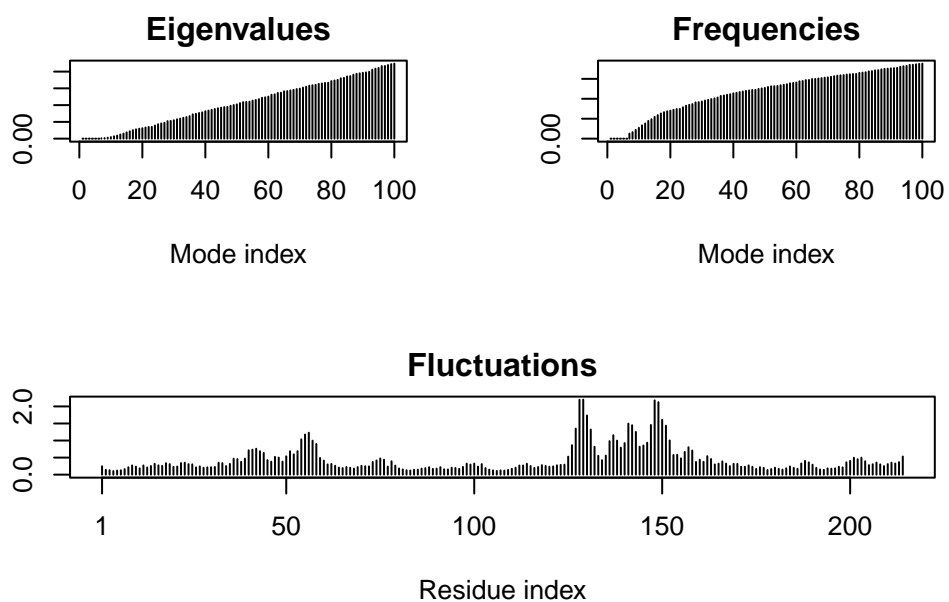
+ attr: atom, xyz, seqres, helix, sheet,  
calpha, remark, call

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

Building Hessian... Done in 0.014 seconds.

Diagonalizing Hessian... Done in 0.287 seconds.

```
plot(m)
```



Generate a "trajectory" of predicted motion

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

## Comparative Structure Analysis

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

**Answer:** msa

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

**Answer:** bio3dview

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

**Answer:** 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  MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMMLRAAVKSGSELGKQAKDIMDAGKLV
      1      .      .      .      .      .      .      60
      61      .      .      .      .      .      .      120
pdb|1AKE|A  DELVIALVKERIAQEDCRNGFLLDGFPRPTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
      61      .      .      .      .      .      .      120
      121      .      .      .      .      .      .      180
pdb|1AKE|A  VGRRVHAPSGRVYHVKNPPKVEGKDDVTGEELTRKDDQEETVRKRLVEYHQMTAPLIG
      121      .      .      .      .      .      .      180
      181      .      .      .      214
pdb|1AKE|A  YYSKEAEAGNTKYAKVDGTPVAEVRADLEKILG
      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?

**Answer:** 214 amino acids

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

```
Searching ... please wait (updates every 5 seconds) RID = GDVBDTXJ014
```

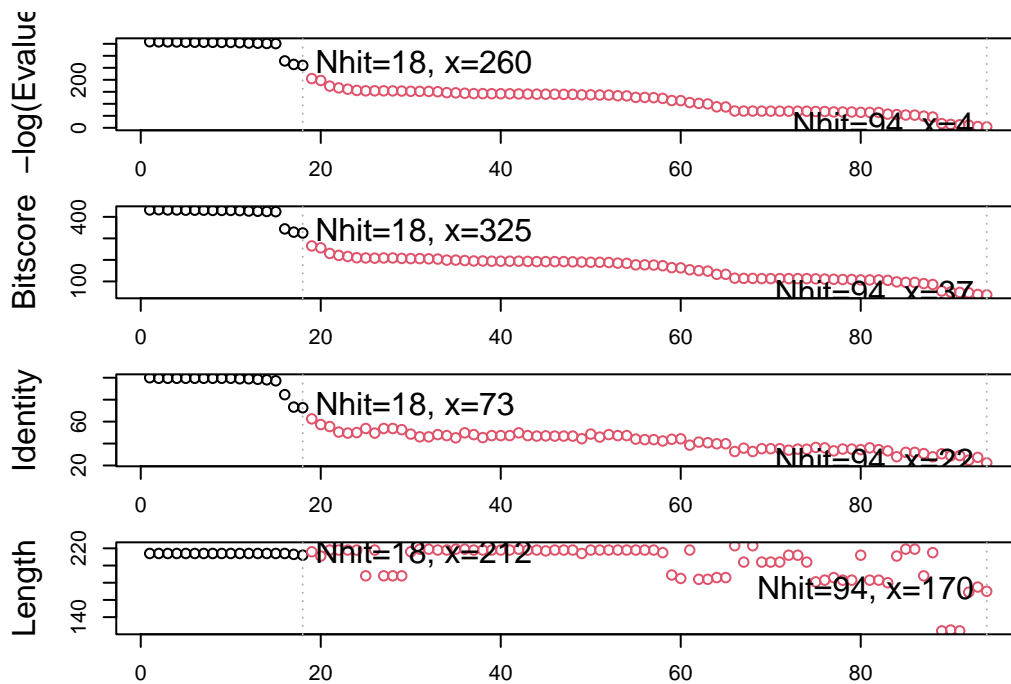
```
.....
```

```
Reporting 94 hits
```

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

```
* Possible cutoff values:    260 3
    Yielding Nhits:         18 94
```

```
* Chosen cutoff value of:    260
    Yielding Nhits:          18
```



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

```
[1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A"
```

```
hits <- NULL
```

```
hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A','6H4I_A')
```

```
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.gz exists. Skipping download
```

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

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

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

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

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

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

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

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

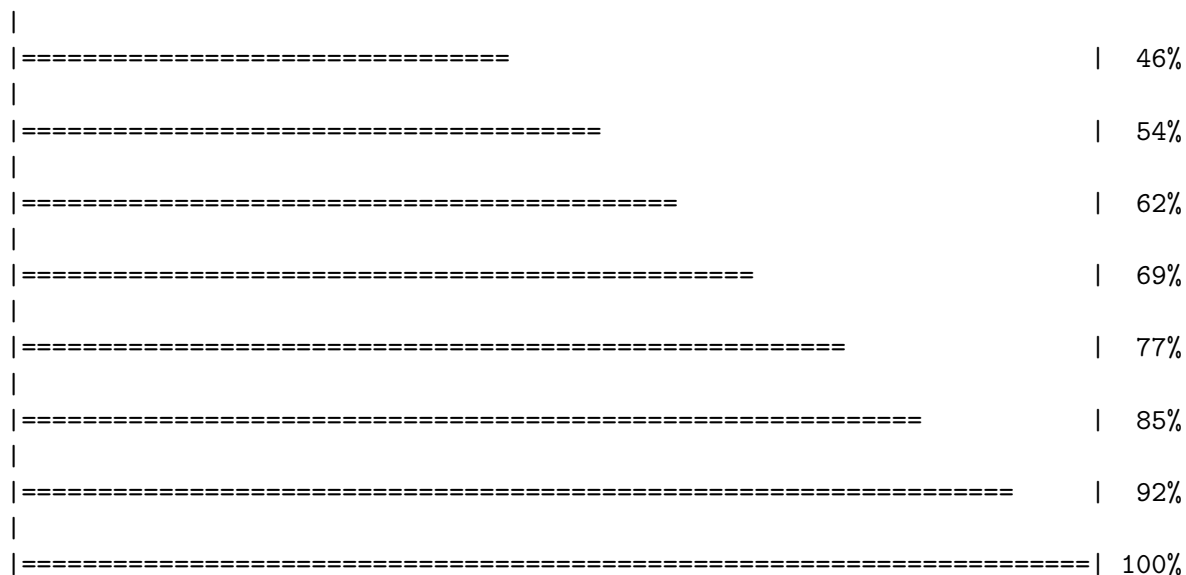
Warning in get.pdb(hits\$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):  
pdbs/6HAM.pdb.gz exists. Skipping download

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

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

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

	0%
=====	8%
=====	15%
=====	23%
=====	31%
=====	38%



This is the code to make a sequence alignment plot from the PDB files. The code cannot be rendered to pdf because the figure margins are too large but it is shown below.

```
pdbbs <- pdbaln(files, fit = TRUE, exe="msa") ids <- basename.pdb(pdbbs$id) plot(pdbbs, labels=ids)
```