Class 10: Structural Bioinformatics Pt. 1

Katelyn Wei (PID: A16682595)

Intro to the PDB Database

Download a CSV file from the PDB site (accessible from "Analyze" > "PDB Statistics" > "by Experimental Method and Molecular Type". Move this CSV file into your RStudio project:

https://tinyurl.com/statspdb

```
stats <- read.csv("PDBstats.csv", row.names = 1)
stats
```

	X.ray	EM	NMR	Multiple.methods	Neutron	Other
Protein (only)	158,844	11,759	12,296	197	73	32
Protein/Oligosaccharide	9,260	2,054	34	8	1	0
Protein/NA	8,307	3,667	284	7	0	0
Nucleic acid (only)	2,730	113	1,467	13	3	1
Other	164	9	32	0	0	0
Oligosaccharide (only)	11	0	6	1	0	4

	Total
Protein (only)	183,201
Protein/Oligosaccharide	11,357
Protein/NA	12,265
Nucleic acid (only)	4,327
Other	205
Oligosaccharide (only)	22

The numbers have commas, making them read as characters. You can use `gsub()` and `as.numeric()` together to change this:

```
x <- stats$X.ray
```

```
  as.numeric( gsub(",", "", x) )
```

```
[1] 158844   9260   8307   2730    164     11
```

```
# Turning it into a function
rm.comma <- function(x) {
  as.numeric( gsub(",", "", x) )
}
```

```
# Apply it to stats
pdbstats <- apply(stats, 2, rm.comma)
pdbstats
```

```
     X.ray    EM   NMR Multiple.methods Neutron Other  Total
[1,] 158844 11759 12296              197      73    32 183201
[2,]   9260  2054    34                8       1     0  11357
[3,]   8307  3667   284                7       0     0  12265
[4,]   2730   113  1467               13       3     1   4327
[5,]    164     9    32                0       0     0    205
[6,]     11     0     6                1       0     4     22
```

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

```
# Total number of structures
totals <- apply(pdbstats, 2, sum)

# Calculating the percentage for each
round((totals/totals["Total"]) * 100, 2)
```

```
          X.ray              EM             NMR Multiple.methods
          84.83            8.33            6.68            0.11
        Neutron           Other           Total
           0.04            0.02          100.00
```

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

```
(pdbstats[1, "Total"] /sum(pdbstats[,"Total"])) * 100
```

```
   Total
86.67026
```

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?

211,377 structures.

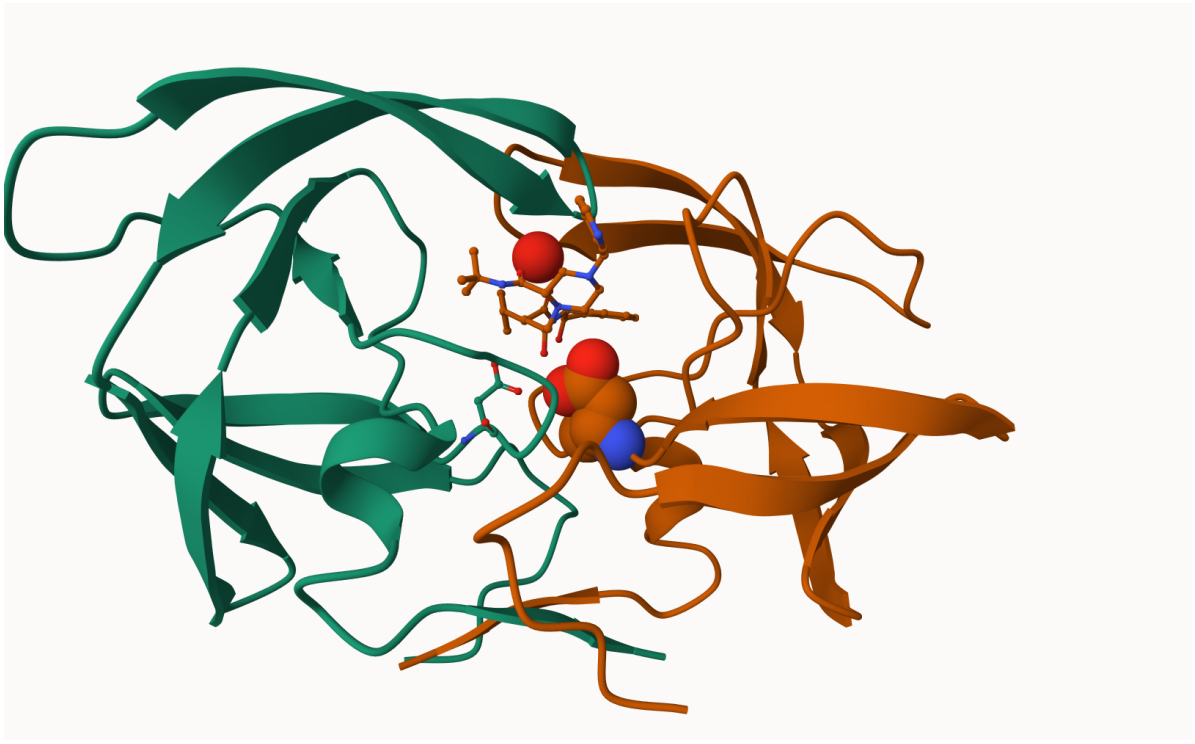
Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure?

Because hydrogen atoms are too tiny to visualize.

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?

Water 308(the big red circle near MK1 in the photo below).

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.



Discussion Topic: Can you think of a way in which indinavir, or even larger ligands and substrates, could enter the binding site?

The protein probably has other configurations that allow larger ligands to enter the binding site.

The bio3d Package for Structural Bioinformatics

```
library(bio3d)

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
```

```
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>
```

Predicting Functional Motions in a Singule Structure

Let's finish today with a bioinformatics model showing how this protein moves.

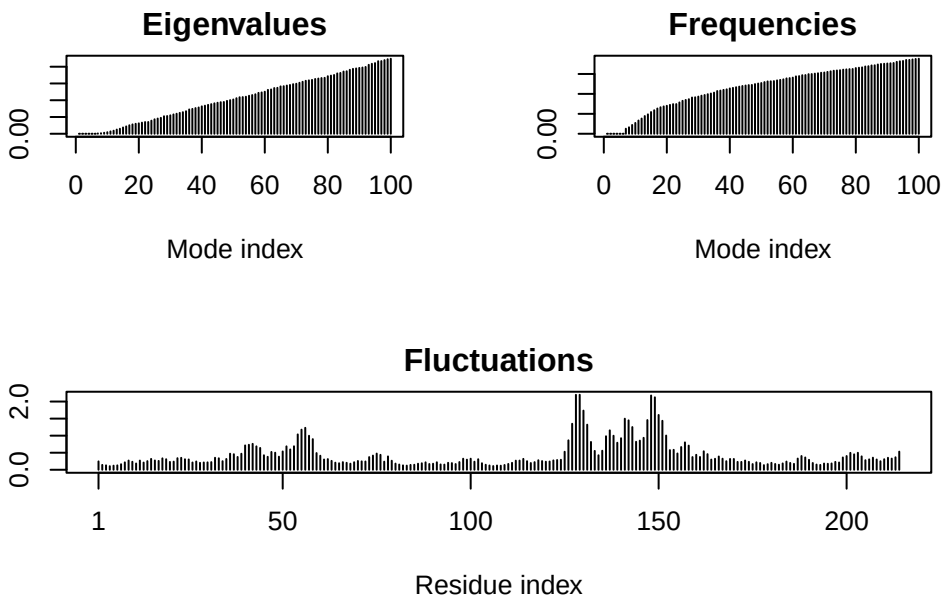
```
adk <- read.pdb("6s36")
```

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

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

```
Building Hessian...        Done in 0.05 seconds.
Diagonalizing Hessian...   Done in 0.523 seconds.
```

```
plot(m)
```




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

Comparative Structure Analysis of Adenylate Kinase (Class 11, 11/7/23)

We need some packages for today's class. These include `bio3d` and `msa`.

the `msa` package is from BioConductor. These packages focus on genomics type work and are managed by the `BiocManager` package.

Install `install.packages("BiocManager")` and then `BiocManager::install("msa")` in the R console.

We can use bio3d's `get.seq()` function to call up a FASTA sequence.

```
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
```

We can now search the PDB database for related sequences with `blast.pdb()`:

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

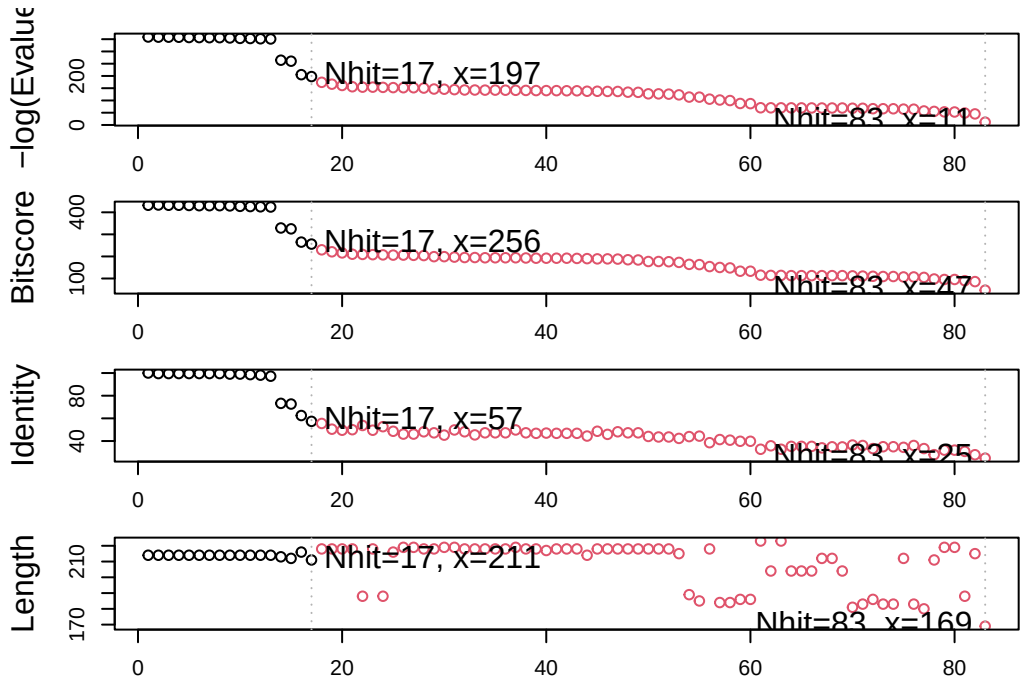
```
 Searching ... please wait (updates every 5 seconds) RID = MN116WT4013
 ..................................
 Reporting 83 hits
```

We can plot b to see our search results:

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

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

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



Our BLAST results are stored in hit.tbl:

```
attributes(b)
```

```
$names
[1] "hit.tbl" "raw"     "url"

$class
[1] "blast"
```

```
b$hit.tbl
```

	queryid	subjectids	identity	alignmentlength	mismatches	gapopens	q.start
1	Query_102283	1AKE_A	100.000	214	0	0	1
2	Query_102283	8BQF_A	99.533	214	1	0	1
3	Query_102283	4X8M_A	99.533	214	1	0	1
4	Query_102283	6S36_A	99.533	214	1	0	1
5	Query_102283	6RZE_A	99.533	214	1	0	1
6	Query_102283	4X8H_A	99.533	214	1	0	1
7	Query_102283	3HPR_A	99.533	214	1	0	1
8	Query_102283	1E4V_A	99.533	214	1	0	1
9	Query_102283	5EJE_A	99.065	214	2	0	1
10	Query_102283	1E4Y_A	99.065	214	2	0	1
11	Query_102283	3X2S_A	98.598	214	3	0	1
12	Query_102283	6HAP_A	98.131	214	4	0	1
13	Query_102283	6HAM_A	97.196	214	6	0	1
14	Query_102283	4K46_A	73.239	213	57	0	1
15	Query_102283	4NP6_A	72.642	212	58	0	2
16	Query_102283	3GMT_A	62.500	216	75	1	2
17	Query_102283	4PZL_A	57.346	211	86	2	2
18	Query_102283	5G3Y_A	55.505	218	88	2	1
19	Query_102283	5G3Z_A	50.459	218	99	2	1
20	Query_102283	5G40_A	49.541	218	101	2	1
21	Query_102283	5X6J_A	50.000	218	98	3	1
22	Query_102283	2C9Y_A	53.723	188	83	1	1
23	Query_102283	1S3G_A	49.541	218	99	3	1
24	Query_102283	1AK2_A	52.660	188	85	1	1
25	Query_102283	3BE4_A	48.611	216	102	3	2
26	Query_102283	1AKY_A	46.119	219	108	3	1
27	Query_102283	3AKY_A	46.119	219	108	3	1
28	Query_102283	3FB4_A	48.165	218	104	2	1
29	Query_102283	4QBI_A	47.248	218	106	2	1
30	Query_102283	1DVR_A	45.205	219	110	3	1
31	Query_102283	3DKV_A	49.772	219	99	3	1
32	Query_102283	3DL0_A	48.165	218	104	2	1
33	Query_102283	1ZIN_A	45.413	218	110	2	1
34	Query_102283	2P3S_A	47.248	218	106	2	1
35	Query_102283	2EU8_A	47.248	218	106	2	1
36	Query_102283	1P3J_A	47.248	218	106	2	1
37	Query_102283	4QBF_A	49.772	219	99	3	1
38	Query_102283	2ORI_A	47.248	218	106	2	1
39	Query_102283	5X6I_A	46.789	218	107	2	1
40	Query_102283	2QAJ_A	47.005	217	106	2	1
41	Query_102283	2OO7_A	46.789	218	107	2	1
42	Query_102283	2OSB_A	46.789	218	107	2	1

43	Query_102283	4MKF_A	46.789	218	107	2	1
44	Query_102283	3TLX_A	44.393	214	106	3	2
45	Query_102283	4MKH_A	48.624	218	101	3	1
46	Query_102283	4QBH_A	45.872	218	109	2	1
47	Query_102283	4TYQ_A	48.165	218	102	3	1
48	Query_102283	4QBG_B	47.248	218	104	3	1
49	Query_102283	4TYP_A	47.248	218	104	3	1
50	Query_102283	4JKY_A	44.037	218	103	5	1
51	Query_102283	2RGX_A	43.578	218	104	4	1
52	Query_102283	4JLO_A	43.578	218	104	5	1
53	Query_102283	1ZAK_A	42.326	215	112	3	1
54	Query_102283	1ZD8_A	43.915	189	96	3	1
55	Query_102283	2AK3_A	44.324	185	101	2	1
56	Query_102283	4NTZ_A	38.532	218	119	4	1
57	Query_102283	2AR7_A	41.304	184	102	3	1
58	Query_102283	3NDP_A	40.761	184	103	3	1
59	Query_102283	1P4S_A	39.785	186	77	2	1
60	Query_102283	2CDN_A	39.785	186	77	2	1
61	Query_102283	3L0P_A	32.735	223	131	7	1
62	Query_102283	5X6L_A	35.784	204	98	3	3
63	Query_102283	2XB4_A	32.735	223	131	7	1
64	Query_102283	5XRU_A	35.294	204	99	3	3
65	Query_102283	5YCC_A	35.294	204	99	3	3
66	Query_102283	5X6K_A	35.294	204	99	3	3
67	Query_102283	5XZ2_A	33.962	212	107	3	3
68	Query_102283	5YCF_A	34.906	212	105	3	3
69	Query_102283	5YCB_A	34.804	204	100	3	3
70	Query_102283	5YCD_A	36.464	181	87	2	3
71	Query_102283	3ADK_A	36.066	183	89	3	3
72	Query_102283	3UMF_A	33.333	186	92	3	3
73	Query_102283	1Z83_A	34.973	183	91	3	3
74	Query_102283	7X7S_A	34.973	183	91	3	3
75	Query_102283	3CM0_A	34.434	212	106	5	3
76	Query_102283	7DE3_A	36.066	183	89	5	3
77	Query_102283	7N6G_6M	33.333	180	108	4	1
78	Query_102283	1UKY_A	27.962	211	117	5	3
79	Query_102283	1TEV_A	31.963	219	109	7	3
80	Query_102283	7E9V_A	31.963	219	109	7	3
81	Query_102283	2BWJ_A	30.851	188	98	4	3
82	Query_102283	1QF9_A	27.907	215	117	5	3
83	Query_102283	7N6G_6A	24.852	169	84	6	76

	q.end	s.start	s.end	evalue	bitscore	positives	mlog.evalue	pdb.id	acc
1	214	1	214	1.45e-156	432.0	100.00	358.83171	1AKE_A	1AKE_A

2	214	21	234	2.38e-156	433.0	100.00	358.33617	8BQF_A	8BQF_A
3	214	1	214	2.60e-156	432.0	100.00	358.24776	4X8M_A	4X8M_A
4	214	1	214	3.82e-156	432.0	100.00	357.86302	6S36_A	6S36_A
5	214	1	214	1.10e-155	431.0	99.53	356.80538	6RZE_A	6RZE_A
6	214	1	214	1.44e-155	430.0	99.53	356.53605	4X8H_A	4X8H_A
7	214	1	214	2.05e-155	430.0	99.53	356.18285	3HPR_A	3HPR_A
8	214	1	214	2.16e-155	430.0	99.53	356.13058	1E4V_A	1E4V_A
9	214	1	214	6.48e-155	429.0	99.07	355.03197	5EJE_A	5EJE_A
10	214	1	214	3.75e-154	427.0	99.07	353.27635	1E4Y_A	1E4Y_A
11	214	1	214	6.28e-154	426.0	98.60	352.76073	3X2S_A	3X2S_A
12	214	1	214	1.86e-153	425.0	98.60	351.67494	6HAP_A	6HAP_A
13	214	1	214	3.80e-153	424.0	98.60	350.96052	6HAM_A	6HAM_A
14	213	1	213	1.69e-115	329.0	84.98	264.27256	4K46_A	4K46_A
15	213	5	216	9.36e-114	325.0	84.43	260.25826	4NP6_A	4NP6_A
16	211	10	225	7.32e-90	265.0	71.30	205.24205	3GMT_A	3GMT_A
17	209	26	235	1.75e-86	256.0	74.41	197.46270	4PZL_A	4PZL_A
18	214	1	213	2.58e-76	230.0	68.81	174.04868	5G3Y_A	5G3Y_A
19	214	1	213	4.95e-73	221.0	69.27	166.48932	5G3Z_A	5G3Z_A
20	214	1	213	1.01e-70	216.0	68.35	161.17101	5G40_A	5G40_A
21	213	1	212	1.85e-68	210.0	65.60	155.96060	5X6J_A	5X6J_A
22	184	17	204	9.33e-68	209.0	69.68	154.34255	2C9Y_A	2C9Y_A
23	213	1	212	9.60e-68	208.0	65.14	154.31402	1S3G_A	1S3G_A
24	184	17	204	2.41e-67	207.0	70.21	153.39357	1AK2_A	1AK2_A
25	213	7	217	6.47e-67	206.0	68.06	152.40603	3BE4_A	3BE4_A
26	214	5	218	9.15e-67	206.0	65.75	152.05945	1AKY_A	1AKY_A
27	214	5	218	1.11e-66	205.0	65.30	151.86626	3AKY_A	3AKY_A
28	214	1	213	4.09e-66	204.0	65.14	150.56207	3FB4_A	3FB4_A
29	214	1	213	1.98e-64	199.0	65.60	146.68235	4QBI_A	4QBI_A
30	214	5	218	4.27e-64	199.0	64.84	145.91383	1DVR_A	1DVR_A
31	214	1	213	2.30e-63	197.0	66.67	144.22995	3DKV_A	3DKV_A
32	214	1	213	1.01e-62	195.0	66.97	142.75033	3DL0_A	3DL0_A
33	214	1	213	1.13e-62	195.0	65.60	142.63806	1ZIN_A	1ZIN_A
34	214	1	213	1.70e-62	194.0	66.97	142.22965	2P3S_A	2P3S_A
35	214	1	213	1.78e-62	194.0	66.97	142.18366	2EU8_A	2EU8_A
36	214	1	213	2.42e-62	194.0	66.97	141.87651	1P3J_A	1P3J_A
37	214	1	213	3.86e-62	194.0	66.21	141.40961	4QBF_A	4QBF_A
38	214	1	213	6.04e-62	193.0	66.97	140.96187	2ORI_A	2ORI_A
39	214	1	213	1.15e-61	192.0	66.51	140.31793	5X6I_A	5X6I_A
40	213	1	212	1.24e-61	192.0	66.82	140.24258	2QAJ_A	2QAJ_A
41	214	1	213	1.44e-61	192.0	66.51	140.09305	2OO7_A	2OO7_A
42	214	1	213	2.46e-61	192.0	66.51	139.55753	2OSB_A	2OSB_A
43	214	1	213	2.74e-61	191.0	66.51	139.44973	4MKF_A	4MKF_A
44	211	31	235	1.73e-60	190.0	64.95	137.60698	3TLX_A	3TLX_A

45	213	3	214	1.77e-60	189.0	65.60	137.58413	4MKH_A	4MKH_A
46	214	1	213	3.54e-60	189.0	64.68	136.89098	4QBH_A	4QBH_A
47	213	1	212	4.35e-60	188.0	65.60	136.68493	4TYQ_A	4TYQ_A
48	213	1	212	7.62e-59	185.0	65.60	133.82174	4QBG_B	4QBG_B
49	213	1	212	1.43e-58	184.0	65.14	133.19226	4TYP_A	4TYP_A
50	214	1	203	6.58e-56	177.0	66.51	127.06073	4JKY_A	4JKY_A
51	214	1	203	7.53e-56	177.0	65.60	126.92587	2RGX_A	2RGX_A
52	214	1	203	2.25e-55	176.0	66.51	125.83125	4JLO_A	4JLO_A
53	214	6	209	4.52e-54	173.0	63.72	122.83108	1ZAK_A	1ZAK_A
54	185	8	190	3.52e-50	164.0	64.55	113.87079	1ZD8_A	1ZD8_A
55	185	7	189	4.84e-50	163.0	65.41	113.55234	2AK3_A	2AK3_A
56	213	6	213	1.73e-46	154.0	62.39	105.37079	4NTZ_A	4NTZ_A
57	182	28	207	6.09e-45	150.0	64.13	101.80968	2AR7_A	2AR7_A
58	182	6	185	5.09e-44	148.0	63.59	99.68647	3NDP_A	3NDP_A
59	182	1	155	8.46e-39	133.0	56.99	87.66547	1P4S_A	1P4S_A
60	182	21	175	1.52e-38	133.0	56.99	87.07952	2CDN_A	2CDN_A
61	209	1	218	3.25e-31	115.0	54.26	70.20148	3L0P_A	3L0P_A
62	205	13	184	3.97e-31	114.0	52.45	70.00137	5X6L_A	5X6L_A
63	209	1	218	4.29e-31	114.0	54.26	69.92385	2XB4_A	2XB4_A
64	205	11	182	4.77e-31	113.0	52.45	69.81779	5XRU_A	5XRU_A
65	205	11	182	5.60e-31	113.0	52.45	69.65737	5YCC_A	5YCC_A
66	205	13	184	6.64e-31	113.0	52.45	69.48703	5X6K_A	5X6K_A
67	213	13	192	7.84e-31	113.0	52.83	69.32090	5XZ2_A	5XZ2_A
68	213	11	190	7.98e-31	113.0	51.89	69.30320	5YCF_A	5YCF_A
69	205	11	182	8.60e-31	113.0	52.45	69.22838	5YCB_A	5YCB_A
70	182	11	164	2.28e-30	112.0	54.70	68.25338	5YCD_A	5YCD_A
71	184	12	167	2.73e-30	111.0	54.10	68.07325	3ADK_A	3ADK_A
72	185	32	188	1.44e-29	110.0	56.45	66.41032	3UMF_A	3UMF_A
73	184	12	167	1.69e-29	109.0	54.64	66.25024	1Z83_A	1Z83_A
74	184	16	171	2.01e-29	109.0	54.64	66.07683	7X7S_A	7X7S_A
75	214	7	185	8.17e-29	107.0	50.94	64.67450	3CM0_A	3CM0_A
76	184	11	166	1.08e-28	107.0	56.28	64.39542	7DE3_A	7DE3_A
77	168	1239	1418	1.41e-25	105.0	53.33	57.22104	7N6G_6	7N6G_6M
78	210	18	196	8.83e-25	97.8	53.55	55.38647	1UKY_A	1UKY_A
79	213	6	192	6.26e-24	95.5	49.77	53.42786	1TEV_A	1TEV_A
80	213	24	210	9.72e-24	95.5	49.77	52.98786	7E9V_A	7E9V_A
81	187	15	173	6.47e-22	90.1	49.47	48.78970	2BWJ_A	2BWJ_A
82	213	9	189	2.52e-20	85.9	47.44	45.12744	1QF9_A	1QF9_A
83	207	760	922	1.17e-05	47.0	42.01	11.35592	7N6G_6	7N6G_6A

These are the related structures in the PDB database that we found via a BLAST search…

```
hits$pdb.id
```

```
 [1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A" "3HPR_A" "1E4V_A"
 [9] "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "4K46_A" "4NP6_A" "3GMT_A"
[17] "4PZL_A"
```

Sidenote: Let's annotate these structures (in other words find out what they are, what species they're from, stuff about the experiment they were solved in, etc.) so they're not just faceless IDs.

For this we can use the `pdb.annotate()` function:

```
anno <- pdb.annotate(hits$pdb.id)
```

```
#attributes(anno)
head(anno)
```

```
       structureId chainId macromoleculeType chainLength experimentalTechnique
1AKE_A        1AKE       A           Protein         214                 X-ray
8BQF_A        8BQF       A           Protein         234                 X-ray
4X8M_A        4X8M       A           Protein         214                 X-ray
6S36_A        6S36       A           Protein         214                 X-ray
6RZE_A        6RZE       A           Protein         214                 X-ray
4X8H_A        4X8H       A           Protein         214                 X-ray
       resolution       scopDomain                    pfam         ligandId
1AKE_A       2.00 Adenylate kinase Adenylate kinase (ADK)              AP5
8BQF_A       2.05             <NA> Adenylate kinase (ADK)              AP5
4X8M_A       2.60             <NA> Adenylate kinase (ADK)             <NA>
6S36_A       1.60             <NA> Adenylate kinase (ADK) CL (3),NA,MG (2)
6RZE_A       1.69             <NA> Adenylate kinase (ADK)    NA (3),CL (2)
4X8H_A       2.50             <NA> Adenylate kinase (ADK)             <NA>
                                        ligandName           source
1AKE_A              BIS(ADENOSINE)-5'-PENTAPHOSPHATE Escherichia coli
8BQF_A              BIS(ADENOSINE)-5'-PENTAPHOSPHATE Escherichia coli
4X8M_A                                          <NA> Escherichia coli
6S36_A CHLORIDE ION (3),SODIUM ION,MAGNESIUM ION (2) Escherichia coli
6RZE_A               SODIUM ION (3),CHLORIDE ION (2) Escherichia coli
4X8H_A                                          <NA> Escherichia coli

1AKE_A STRUCTURE OF THE COMPLEX BETWEEN ADENYLATE KINASE FROM ESCHERICHIA COLI AND THE INHIBI
```

```
8BQF_A
4X8M_A
6S36_A
6RZE_A
4X8H_A
                                                    citation rObserved   rFree
1AKE_A               Muller, C.W., et al. J Mol Biol (1992)   0.19600      NA
8BQF_A Scheerer, D., et al. Proc Natl Acad Sci U S A (2023)   0.22073 0.25789
4X8M_A              Kovermann, M., et al. Nat Commun (2015)   0.24910 0.30890
6S36_A                Rogne, P., et al. Biochemistry (2019)   0.16320 0.23560
6RZE_A                Rogne, P., et al. Biochemistry (2019)   0.18650 0.23500
4X8H_A              Kovermann, M., et al. Nat Commun (2015)   0.19610 0.28950
         rWork spaceGroup
1AKE_A 0.19600  P 21 2 21
8BQF_A 0.21882  P 2 21 21
4X8M_A 0.24630    C 1 2 1
6S36_A 0.15940    C 1 2 1
6RZE_A 0.18190    C 1 2 1
4X8H_A 0.19140    C 1 2 1
```

Now we can download all these structures for further analysis with the `get.pdb()` function. The gzip argument compresses the file and the path argument determines where the files will be stored.

```
# Download releated 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.gz exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8BQF.pdb.gz exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8M.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/4X8H.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/4NP6.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%
|
|====                                                                  |   6%
|
|========                                                              |  12%
|
|============                                                          |  18%
|
|================                                                      |  24%
|
|====================                                                  |  29%
|
|========================                                              |  35%
|
|============================                                          |  41%
|
|================================                                      |  47%
|
|====================================                                  |  53%
|
|========================================                              |  59%
|
|============================================                          |  65%
|
|================================================                      |  71%
|
|====================================================                  |  76%
|
|============================================================          |  82%
|
|================================================================      |  88%
|
|====================================================================  |  94%
|
|======================================================================| 100%
```

Now we have all these structures we can align and superpose using the `pdaln()` function.

```
# Align related PDBs
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

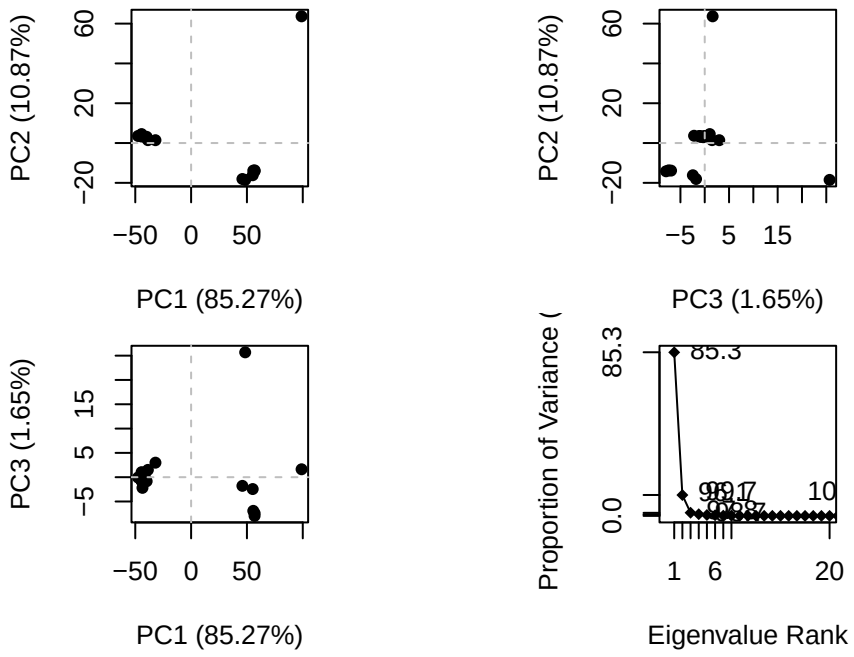
```
Reading PDB files:
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/8BQF_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/4X8H_A.pdb
pdbs/split_chain/3HPR_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/5EJE_A.pdb
pdbs/split_chain/1E4Y_A.pdb
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
.   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/8BQF_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 3   name: pdbs/split_chain/4X8M_A.pdb
pdb/seq: 4   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 6   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 7   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 8   name: pdbs/split_chain/1E4V_A.pdb
```

```
pdb/seq: 9   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 10   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 11   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 12   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 13   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 14   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 15   name: pdbs/split_chain/4NP6_A.pdb
pdb/seq: 16   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: 17   name: pdbs/split_chain/4PZL_A.pdb
```

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




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