# HW Class 6 Question 6

Emma Bell (A19247017)

## original code

‘s1 <- read.pdb(“4AKE”) s2 <- read.pdb(“1AKE”) s3 <- read.pdb(“1E4Y”)

s1.chainA <- trim.pdb(s1, chain=“A”, elety=“CA”) s2.chainA <- trim.pdb(s2, chain=“A”, elety=“CA”) s3.chainA <- trim.pdb(s3, chain=“A”, elety=“CA”)

s1.b <- s1.chainA*atom*b s2.b <- s2.chainA*atom*b s3.b <- s3.chainA*atom*b

plotb3(s1.b, sse=s1.chainA) plotb3(s2.b, sse=s2.chainA) plotb3(s3.b, sse=s3.chainA)‘

## Why is this a problem?

You are repeating the same steps, you are more likely to make an error. It is also too inefficient if you have a large database

SO we need to rescale!

How I understand it: The code repeats: 1. read a PDB file 2. Trim to chain A and C atoms 3. Claim the B factors 4. Plot

Before I have done everything, I have made sure I have installed all the packages I need, like `bio3d`

## Step 1

instead of using s1, s2 and s3, we store the PDB IDs in a vector, because we can loop these easily after that.

```
pdb_id <- c("4AKE", "1AKE", "1E4Y")
```

## Step 2

I am now using `lapply()` which means do the same thing to all the elements. This will sort out the first bit of the repetitive code.

'paste0(id, ".pdb) creates a filename so that r can read it as a file

```
library(bio3d)
pdbs <- lapply( pdb_id, function(id){
  read.pdb(paste0(id, ".pdb"))
})
```

```
   PDB has ALT records, taking A only, rm.alt=TRUE
```

Now, we have the protein data in R

## Step 3

Now we have tackled the first part woop! Onto the next. We need to apply `trim.pdb()` to each.

So, I am going to make up a function. For each pdb, I need to keep only chain A and the C alpha atoms.

```
chainA_ca <- lapply(pdbs, function(pdb) {
  trim.pdb(pdb, chain = "A", elety = "CA")
})
```

## Step 4

Now, in the repeated steps, we want to take out the B factors. We need to get the atom table from each PDB.

```
b_factors <- lapply(chainA_ca, function(pdb) {
  pdb$atom$b
})
```

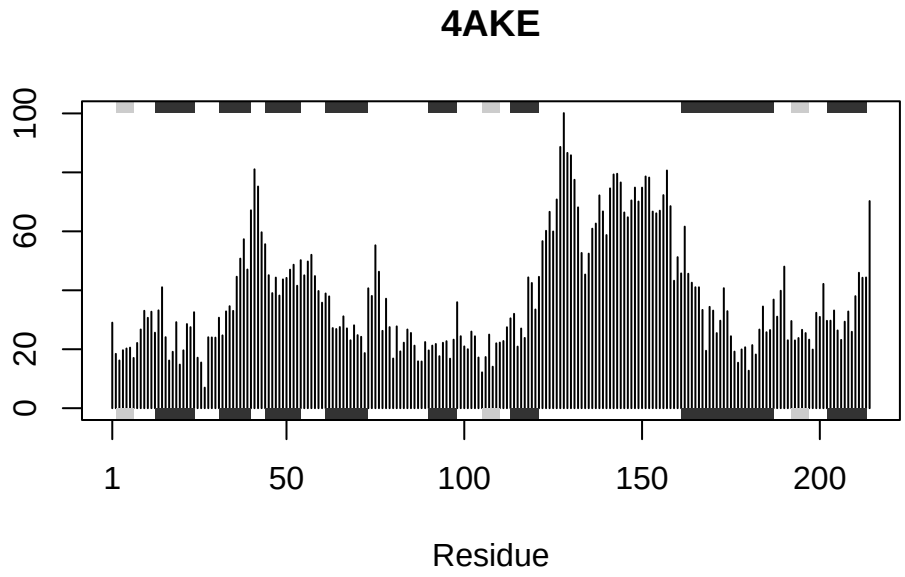
## Step 5

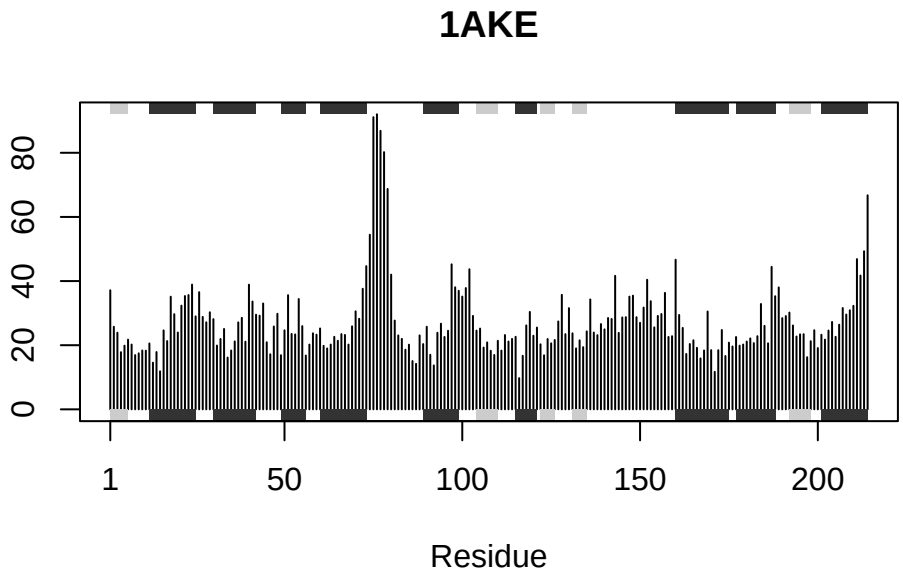
Plotting the steps! Final one!

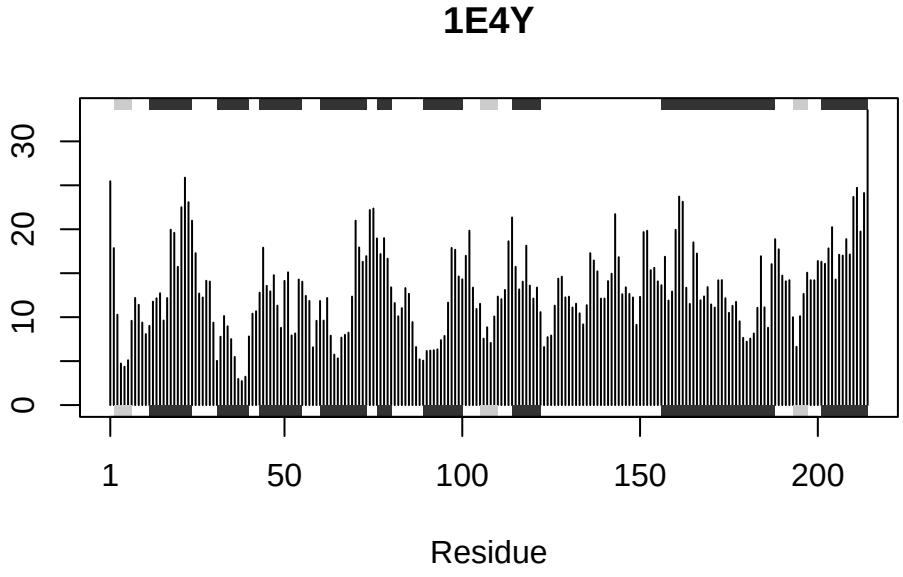
`seq_along()` is a great function, as it selects the B factors, matching structure and generates one plot.

We also are using plotb3() which is in the original code. This plots B factor vs residue index.

```
for (i in seq_along(chainA_ca)) {
  plotb3(
    b_factors[[i]],
    sse = chainA_ca[[i]],
    main = pdb_id[i]
  )
}
```






the `for` ia basically saying for each thing that corresponds to one protein, run the code i is representing the PDB files. so the first loop i-1, which is 4AKE.