HW_class6_function

Monika Ramos

Question: improve this code by making it a useful function

Can you improve this analysis code?

Below is the repetitive code that needs **automation**:

#library(bio3d) #s1 <- read.pdb("4AKE") # kinase with drug #s2 <- read.pdb("1AKE") # kinase no drug #s3 <- read.pdb("1E4Y") # kinase with drug #s1.chainA <- trim.pdb(s1, chain="A", elety="CA") #s2.chainA <- trim.pdb(s2, chain="A", elety="CA") #s3.chainA <- trim.pdb(s1, 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, typ="l", ylab="Bfactor") #plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor") #plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")

The code is broken up into 4 steps. 1. reading in the PDB entry by its unique identifier 2. trim and store trimmed pdb entry always by chain A and elety CA 3. store the atom part b under a new object 4. plot the result using function plotb3()

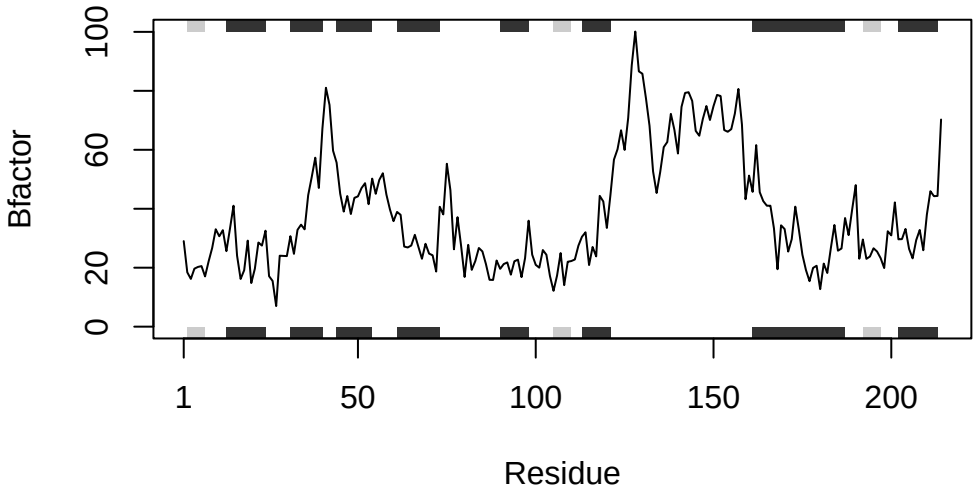
```
library(bio3d)
pdb_function <- function(protein) {
s_x <- read.pdb(protein)
s_x.chainA <- trim.pdb(s_x, chain="A", elety="CA")
s_x.b <- s_x.chainA$atom$b
plotb3(s_x.b, sse=s_x.chainA, typ="l", ylab="Bfactor")
}
```

Now test the function on the three inputs

First, 4AKE:

```
pdb_function("4AKE")
```

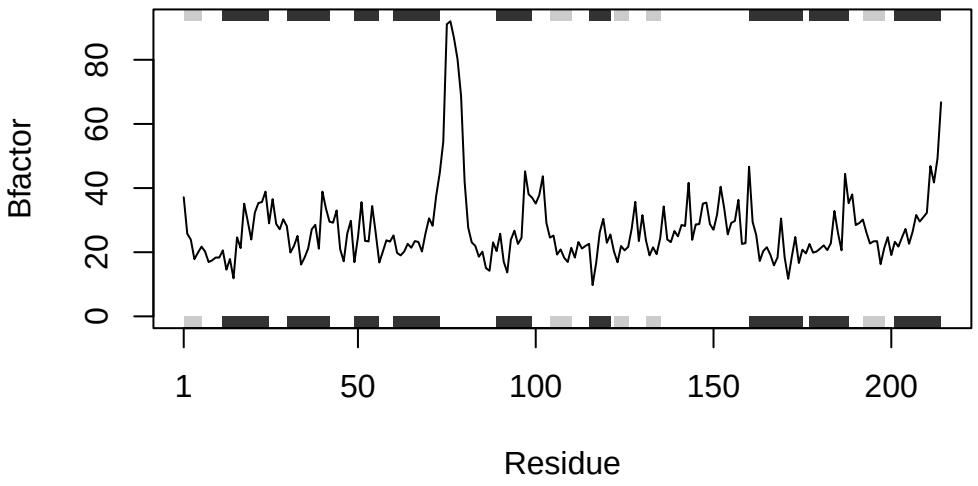
```
Note: Accessing on-line PDB file
```



Then. 1AKE

```
pdb_function("1AKE")
```

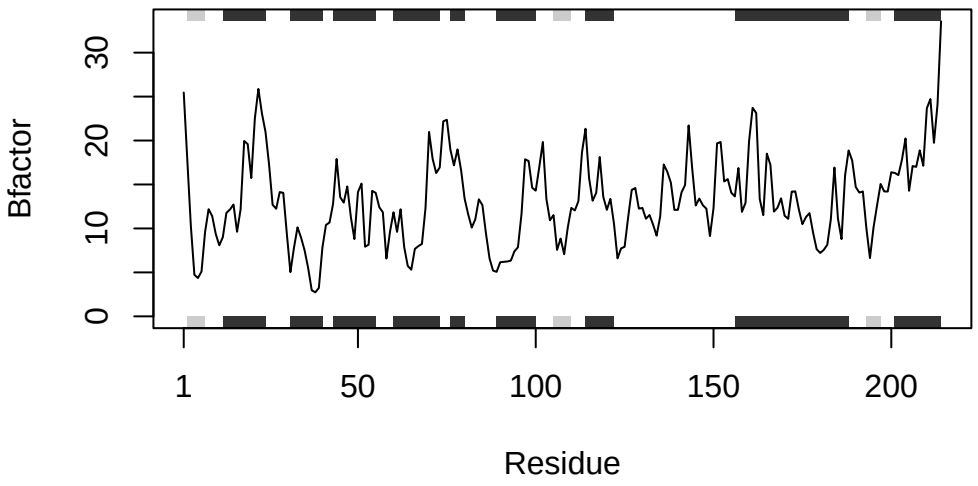
```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```



Lastly, 1E4Y

```
pdb_function("1E4Y")
```

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



The function works!