Class 6 HW

Jacqueline Cheung(A17085191)

2025-01-23

How can we improve this code?

```
library(bio3d)

s1 <- read.pdb("4AKE") # kinase with drug
```

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

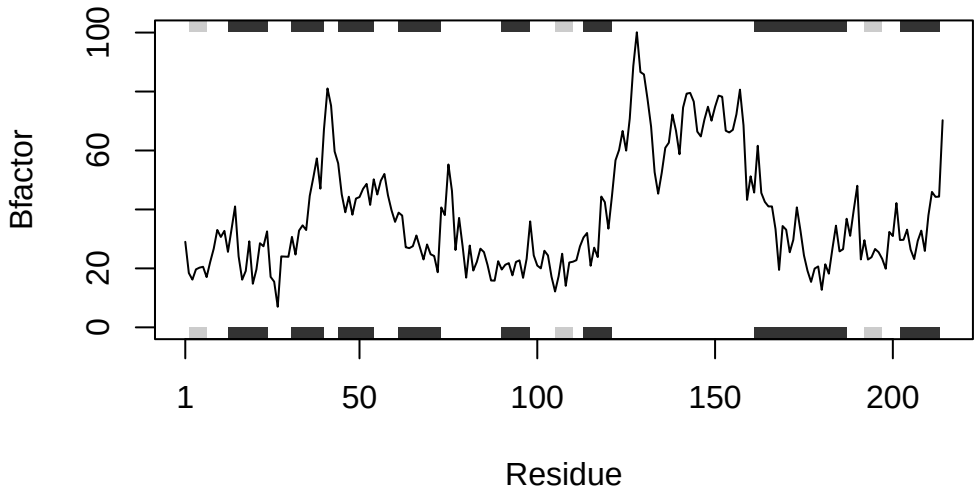
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

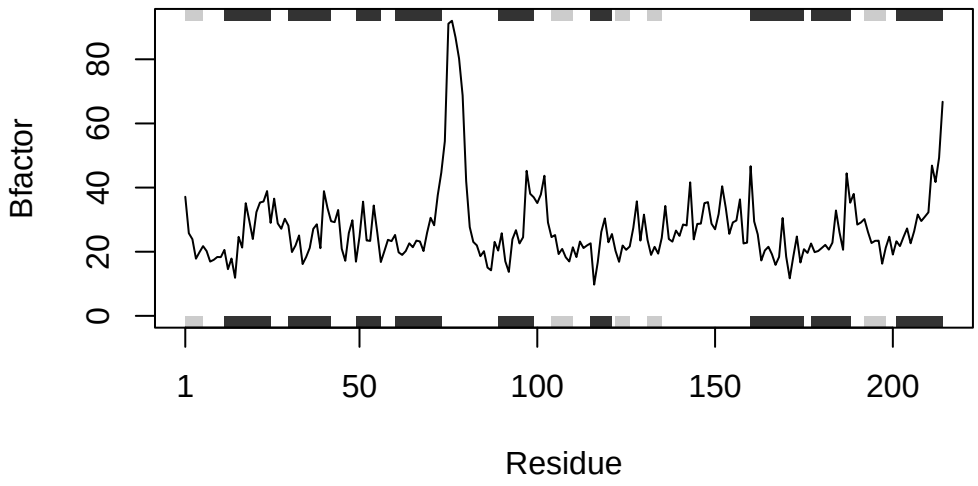
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

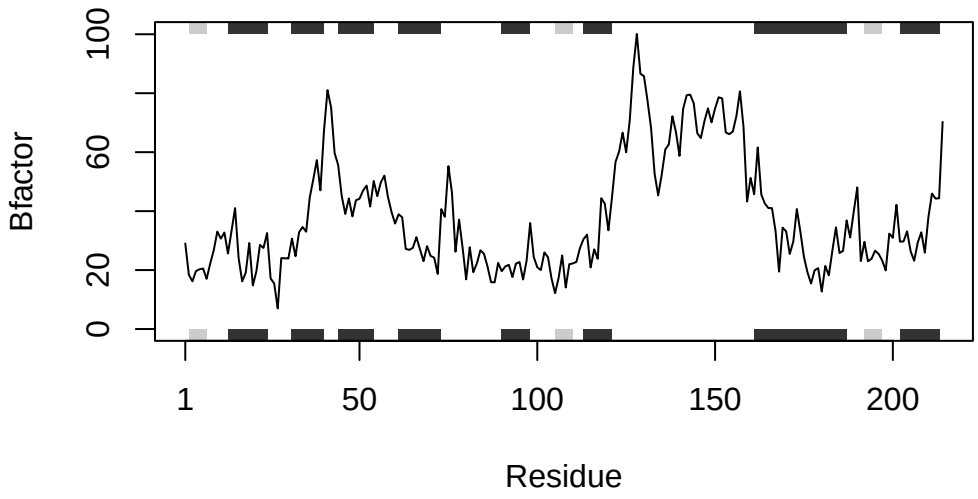
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")

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")
```



Condensing the function

> Q. How would you generalize the original code above to work with any set of input protein structures

I created a function that condenses the original code on order to generically read any PBD file. When a list of the PDB codes is made for the proteins of interest, sapply() can then be used to apply the first generic function to the list of the PDB codes. The output of the code is the graphs of the proteins of interest.

```
#Create a list of the PDB codes of interest
pdb.files <-c("4AKE", "1AKE", "1E4Y")
```

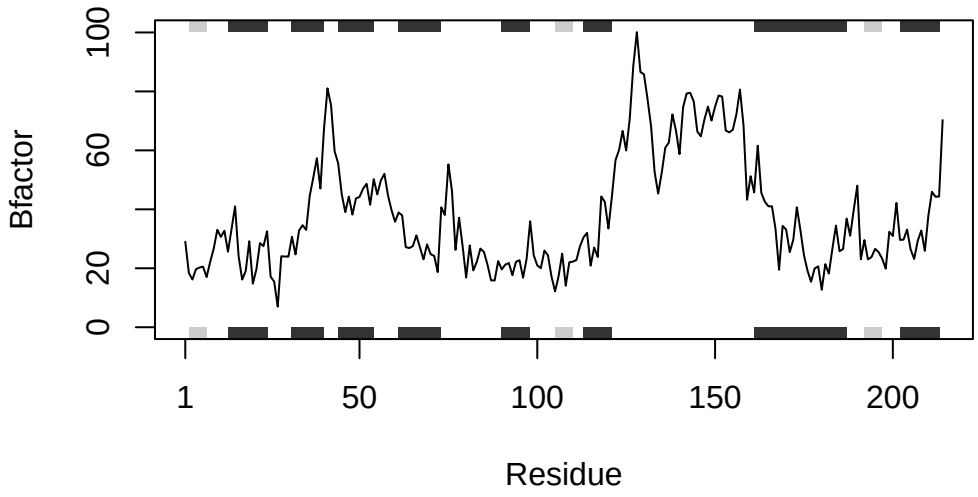
```
library(bio3d)

#Create a function to read any pdb document
generate_data = function(pdb){
   s <- read.pdb(pdb) #s uses read.pdb reads the PDB files in bio3d
   s.chain <- trim.pdb(s, chain="A", elety="CA") #s.chain uses trim.pdb to
```

```
   #produce a smaller PDB containing the chain A and the atom type CA
   s.b <- s.chain$atom$b #s.b extracts an atom vector from s.chain
   plotb3(s.b, sse=s.chain, typ="l", ylab="Bfactor") #s.b is ploted as a line
   #graph, with s.chain as the sum of squared errors, and the label of the
   #y-axis being "Bfactor"
    }
```

```
#ans is created by applying the function generate_data to the list of the
#pdb.files, creating plots for all the proteins of interest
ans <- sapply(pdb.files,generate_data)
```

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

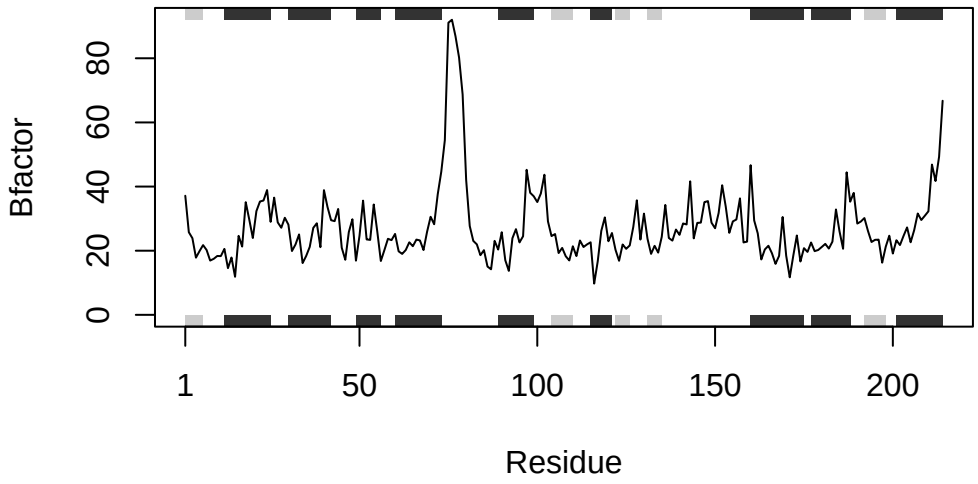
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\bluer\AppData\Local\Temp\RtmpmMuTaF/4AKE.pdb exists. Skipping download
```



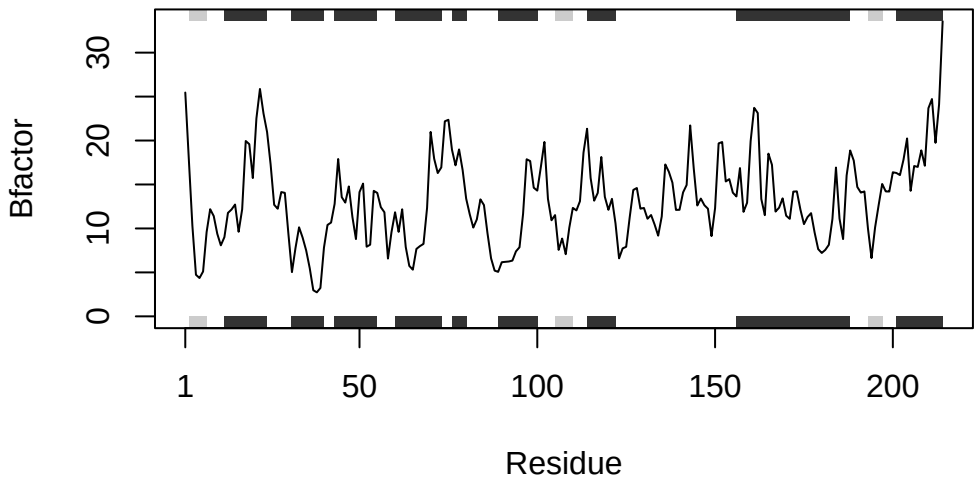

```
  Note: Accessing on-line PDB file
```

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\bluer\AppData\Local\Temp\RtmpmMuTaF/1AKE.pdb exists. Skipping download
```

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




```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\bluer\AppData\Local\Temp\RtmpmMuTaF/1E4Y.pdb exists. Skipping download
```


Bfactor
0
10
20
30
1
50
100
150
200
Residue