Class 6 HW

Justin Robinson (PID: A16307501)

2024-04-22

##This is the original code

```
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

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

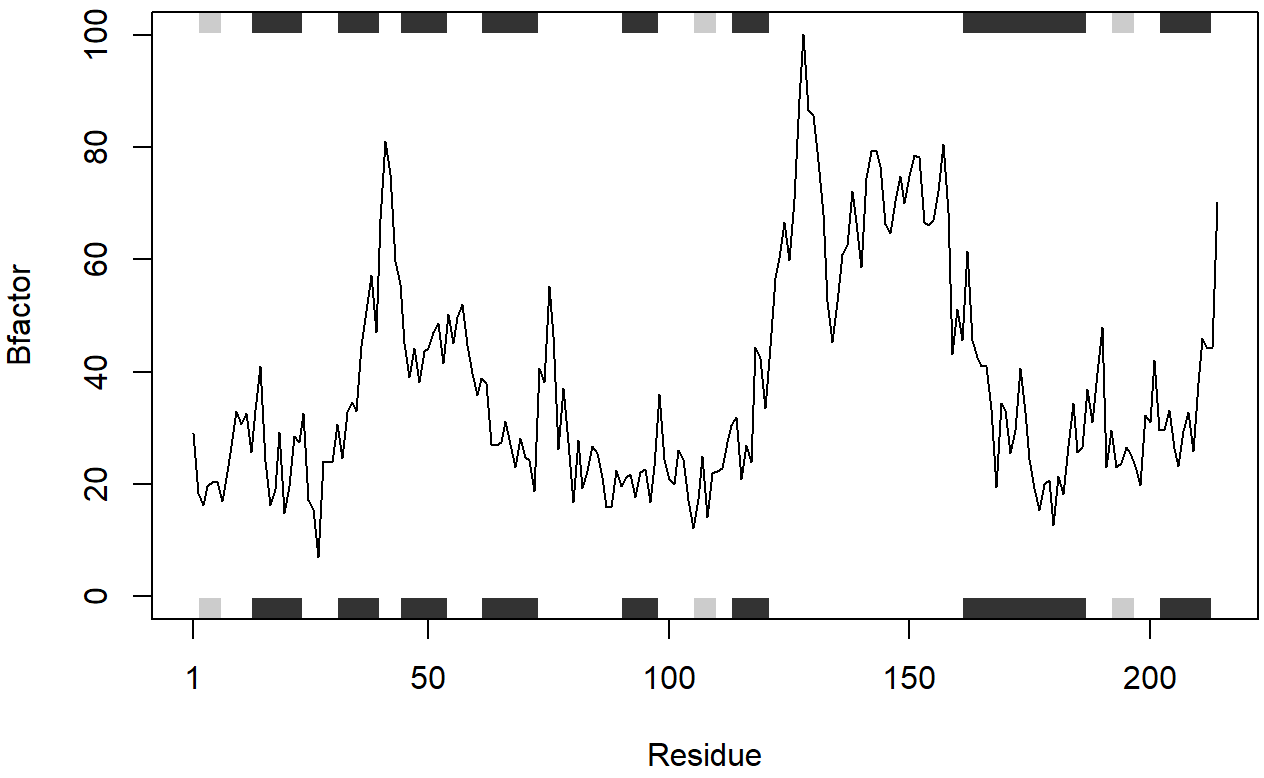
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

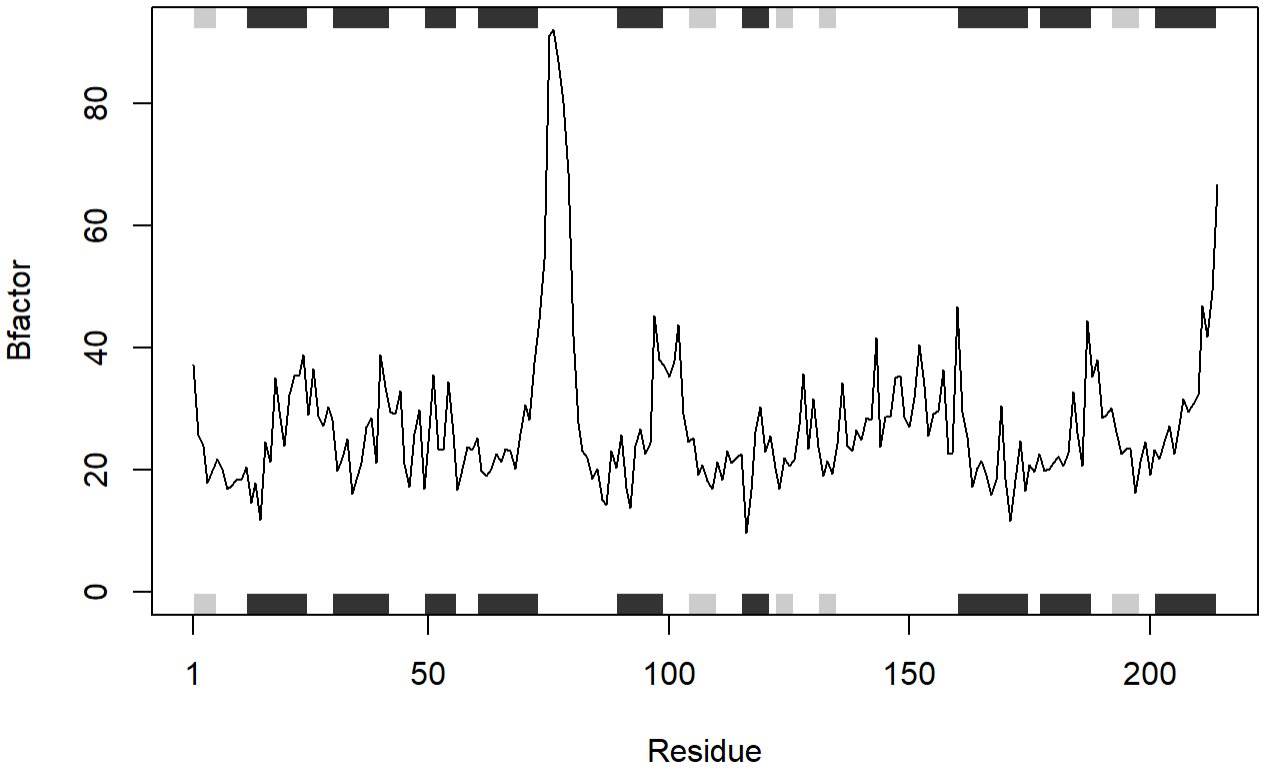
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

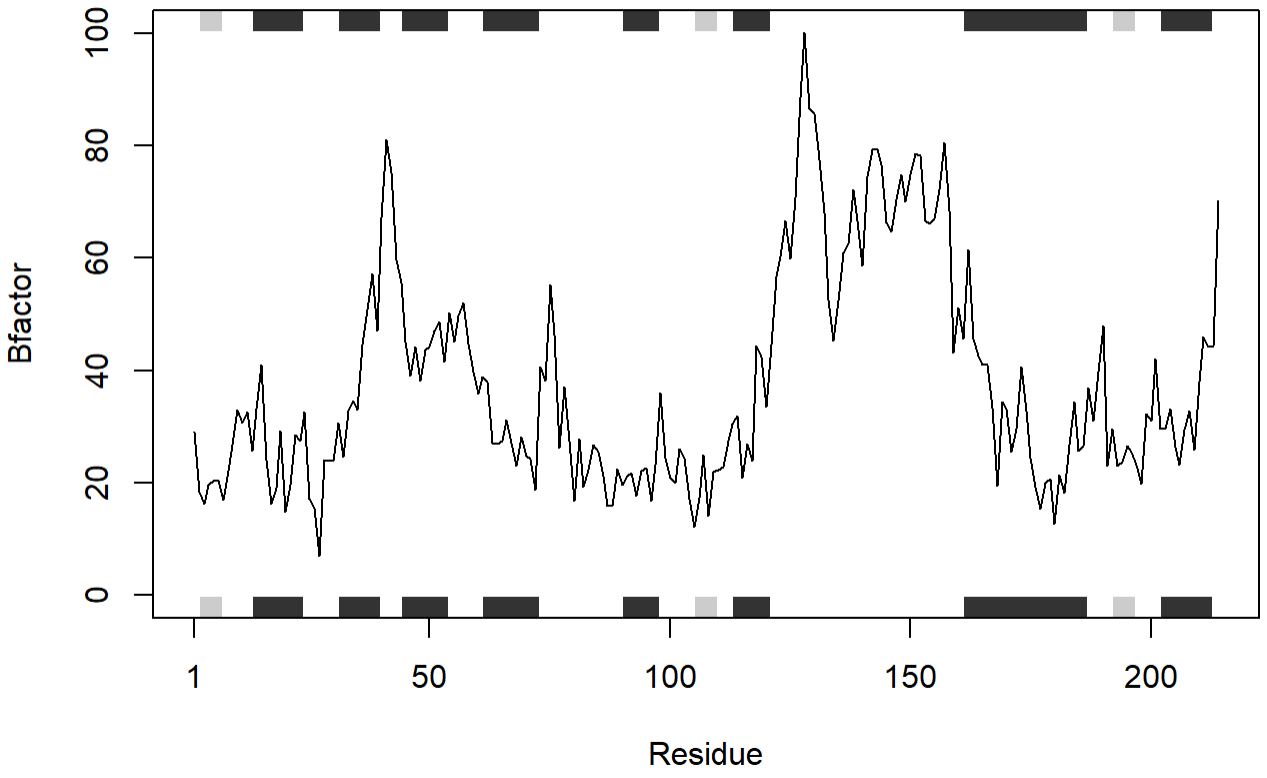
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")

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")
```



##here is my proposed edit:

> The original code was broken down so that the protein of interest can be read through the PDB database, trimmed down, have atom b highlighted, and then be plotted; rather than writing out this entire process for every one of the three proteins

```
library(bio3d)

#enter the x variable as the 4 character pdb code
x <- "4AKE"

plotbfactor <- function(x) {

  pdbentry <- read.pdb(x)

  trimmed <- trim.pdb(pdbentry, chain="A", elety="CA")

  atomb <- trimmed$atom$b

plotb3(atomb, sse=trimmed, typ="1", ylab="Bfactor")
}
```

To do the same for the other two proteins, we will simply do `plotbfactor()` and insert the protein's 4 character PDB code as the variable x