class06_hw

Delisa A Ramos, pID:A69026881

##Section 1A

```
# (A. Can you improve this analysis code?
df <- data.frame(a=1:10, b=seq(200,400,length=10),c=11:20,d=NA)
df$a <- (df$a - min(df$a)) / (max(df$a) - min(df$a))
df$b <- (df$b - min(df$b)) / (max(df$b) - min(df$b))
df$c <- (df$c - min(df$c)) / (max(df$c) - min(df$c))
df$d <- (df$d - min(df$d)) / (max(df$d) - min(df$d))

test1 <- function(col){
  #gets the min and max through the range function
  r <- range(col)
  #math equation from above
  (col - r[1]) / (r[2] - r[1])
}

# applies my function across all of the columns of the data frame specified
df <- apply(df, MARGIN=2, FUN=test1)
df
```

```
             a         b         c  d
 [1,] 0.0000000 0.0000000 0.0000000 NA
 [2,] 0.1111111 0.1111111 0.1111111 NA
 [3,] 0.2222222 0.2222222 0.2222222 NA
 [4,] 0.3333333 0.3333333 0.3333333 NA
 [5,] 0.4444444 0.4444444 0.4444444 NA
 [6,] 0.5555556 0.5555556 0.5555556 NA
 [7,] 0.6666667 0.6666667 0.6666667 NA
 [8,] 0.7777778 0.7777778 0.7777778 NA
 [9,] 0.8888889 0.8888889 0.8888889 NA
[10,] 1.0000000 1.0000000 1.0000000 NA
```

##Section 1B

Q1: What type of object is returned from the read.pdb() function? A list containing 8 elements including: atom, xyz, seqres, helix, sheet, calpha, remark, and call.

Q2: What does the trim.pdb() function do? It just trims a PDB to a certain range of atoms within the protein. chain=" " is specifying the atoms you want to include. elety=" " specifies the atom type. so "CA" refers to atoms with an alpha-carbon, i.e. amino acid.

Q3: What input parameter would turn off the marginal black and grey rectangles in the plots and what do they represent in this case? You would set 'top=F' and 'bot=F'. They represent varying levels of secondary structure within the protein.

Q4. What would be a better plot to compare across the different proteins? i don't know.

Q5: Which proteins are more similar to each other in their B-factor trends. How could you quantify this? s1.b (protein PDB:4AKE) and s3.b, (protein PDB:1E4Y) are more similar according to their b-factor trends. The distances between the b-factor scores of the proteins were computed and graphed in a dendrogram plot.

```
#install.packages("bio3d")
library(bio3d)

# Can you improve this analysis code?

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

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

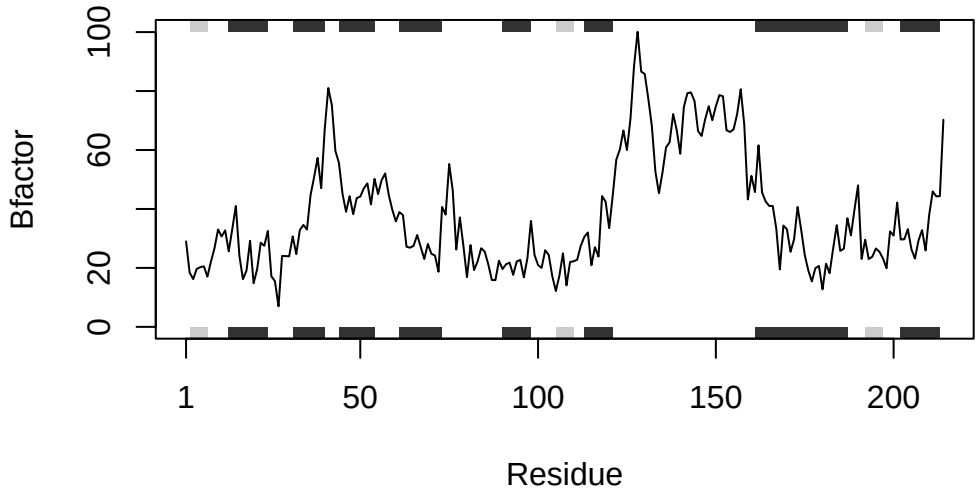
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

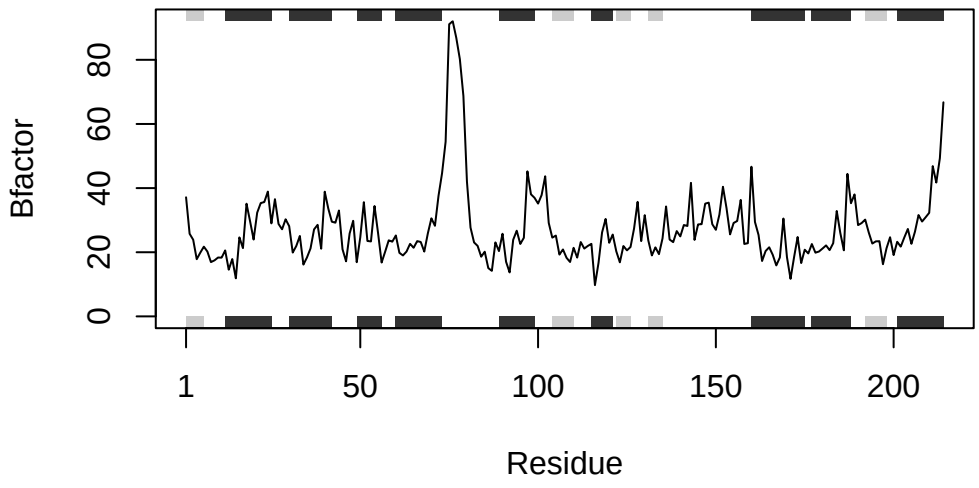
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

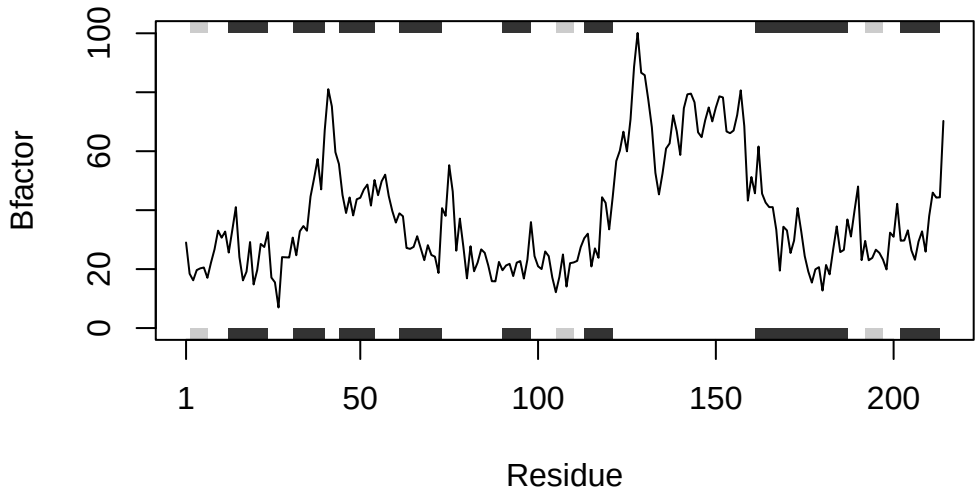
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")

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")
```

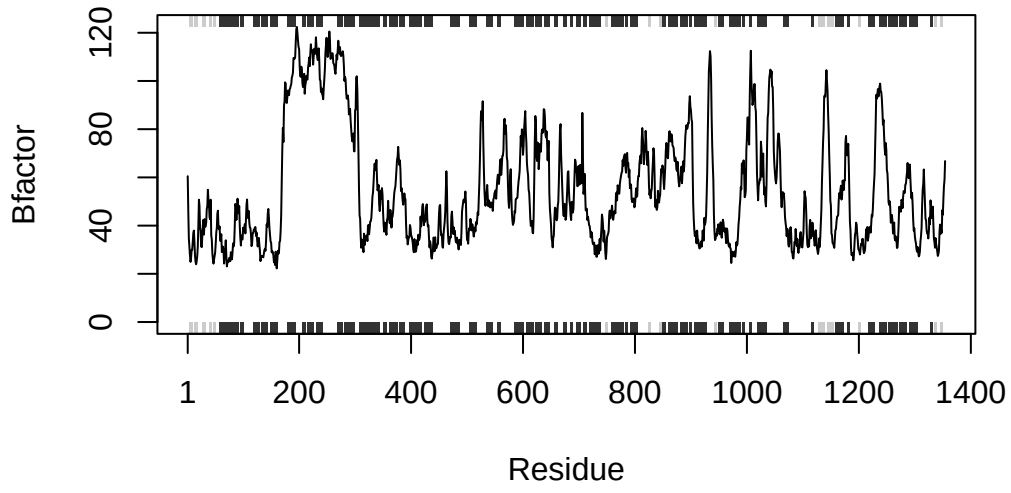


```
plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")
```




```
#my function, variables are pdb and chain
protein <- function(pdb, chain, atom) {
  #reads in the pdb into variable s
  protein <- read.pdb(pdb)
  #trims the pdb down to whatever chain is specified
  chain <- trim.pdb(protein, chain=chain, elety=atom)
  #saves the b factor scores as s.b variable
  b.fac <- chain$atom$b
  #plots the protein chain specified by b-factor
  plotb3(b.fac, sse=chain, typ="l", ylab="Bfactor")
}

#calling the function and specifying the pdb code and chain identifier
protein("7S4X", "A", "CA")
```

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




```
# dendogram plot for Q4
hc <- hclust(dist(rbind(s1.b, s2.b, s3.b) ) )
plot(hc)
```