

# HW 6

Elsa Chen (A16632961)

## Table of contents

Section 1: Improving analysis code by writing functions . . . . .	1
A. . . . .	1
B. . . . .	1

## Section 1: Improving analysis code by writing functions

### A.

```
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$b - min(df$a)) / (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$a) - min(df$d))

normalize <- function(x){
  (x - min(x)) / (max(x) - min(x))
}
df <- as.data.frame(apply(df, 2, normalize))
```

### B.

```
library(bio3d)
```

Warning: package 'bio3d' was built under R version 4.3.3

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

Note: Accessing on-line PDB file

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

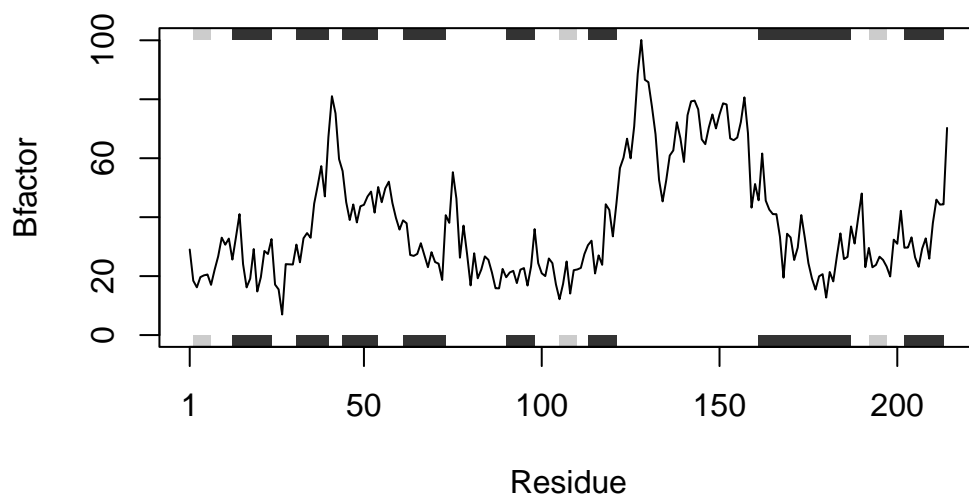
Note: Accessing on-line PDB file

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

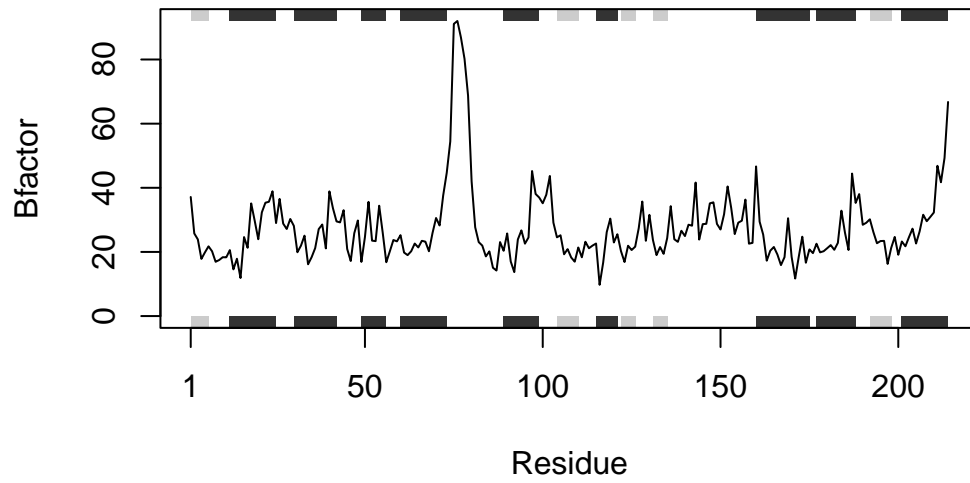
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

Note: Accessing on-line PDB file

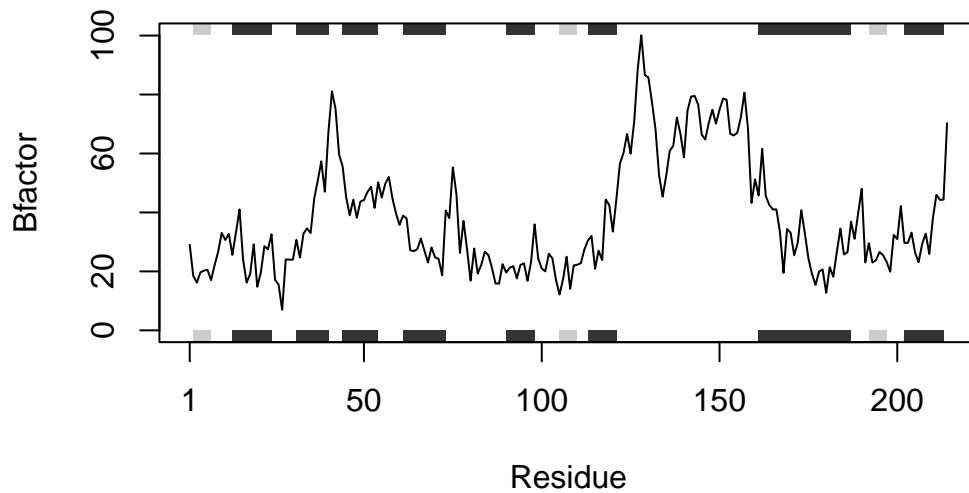
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")  
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")  
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")  
s1.b <- s1.chainA$atom$b  
s2.b <- s2.chainA$atom$b  
s3.b <- s3.chainA$atom$b  
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



```
plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor")
```



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



Q1: What type of object is returned from the `read.pdb()` function?

```
class(s1)
```

```
[1] "pdb" "sse"
```

a pdb file (Protein Data Bank)

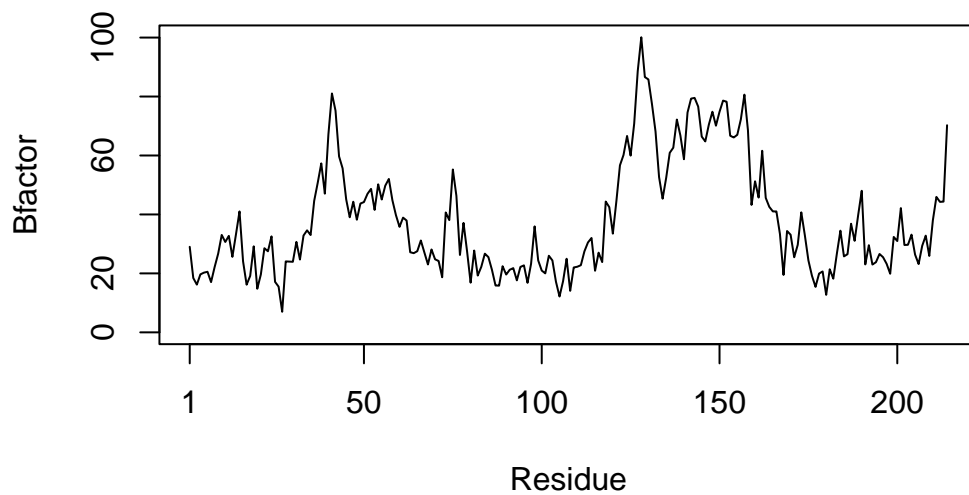
Q2. What does the `trim.pdb()` function do?

“Produce a new smaller PDB object, containing a subset of atoms, from a given larger PDB object”

Q3. What input parameter would turn off the marginal black and grey rectangles in the plots and what do they represent in this case?

The rectangles represent sse (secondary structure element), in this case ‘s1.chainA’

```
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor", top = FALSE, bot = FALSE)
```



Q4. What would be a better plot to compare across the different proteins?

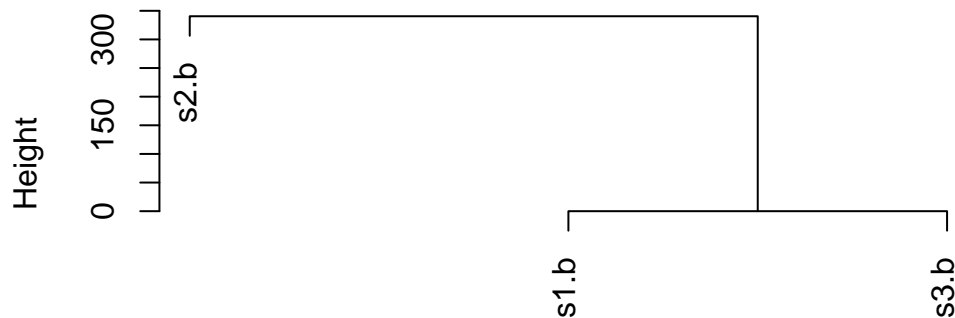
a line plot with all the proteins plotted in different colors

Q5. Which proteins are more similar to each other in their B-factor trends. How could you quantify this?

s1.b and s3.b are more similar to each other

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

## Cluster Dendrogram



```
dist(rbind(s1.b, s2.b, s3.b))  
hclust (*, "complete")
```

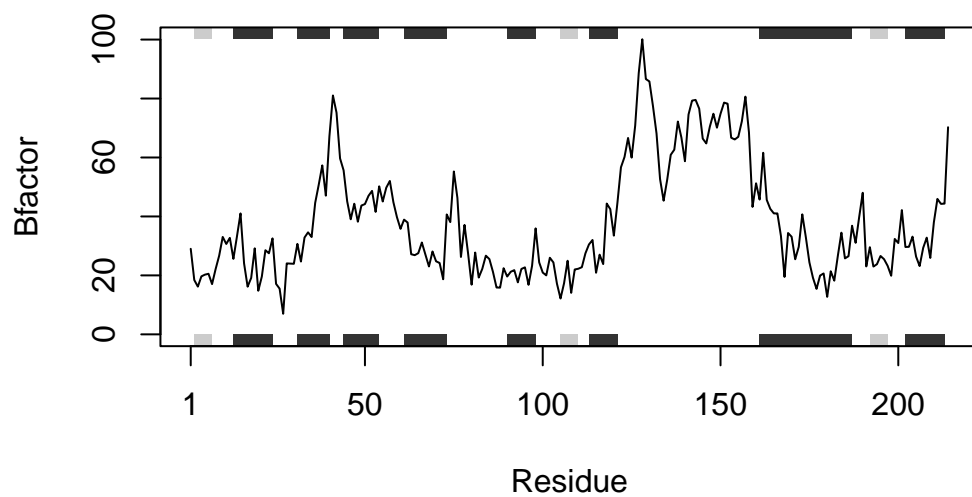
Q6. How would you generalize the original code above to work with any set of input protein structures?

improving the code

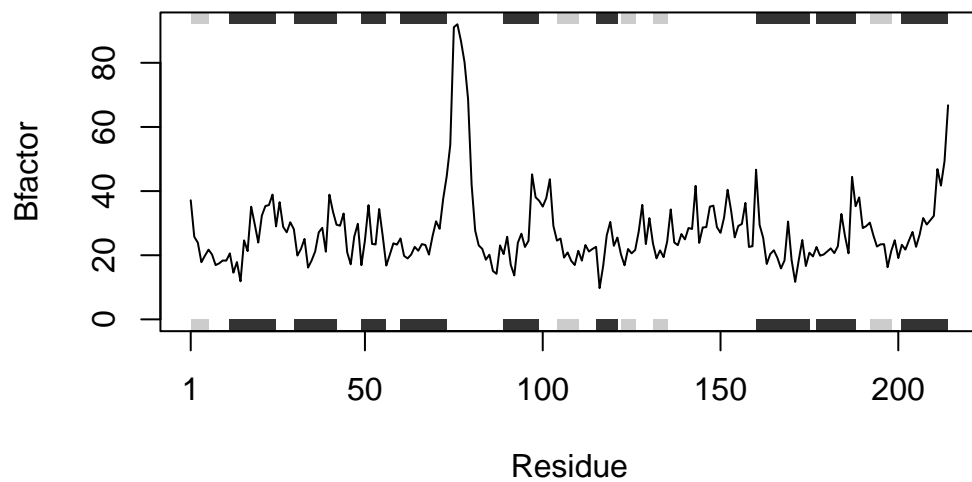
```
plotChainA <- function(x){# x would be a pdb file with the protein we want  
  x.chainA <- trim.pdb(x, chain = "A", elety = "CA") # trim the pdb to a smaller subset  
  x.b <- x.chainA$atom$b # get the desired item from the subset  
  plotb3(x.b, sse = x.chainA, typ = "l", ylab = "Bfactor") # plot the item  
} # this should return a line plot for the desired protein
```

see if it works

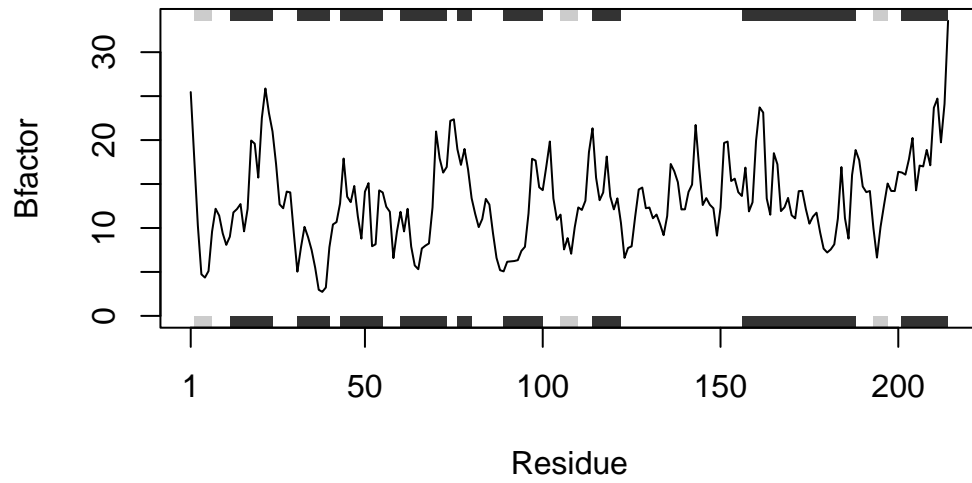
```
plotChainA(s1)
```



```
plotChainA(s2)
```



```
plotChainA(s3)
```



yay it does!