Class06HW

Thisha Thiagarajan A15474979

10/19/2021

Section1 Improving Analysis Code by Writing Functions

a Improve the code below.

The function section1() simplifies the code. Now you can just call the function of df$column and it will perform x <- (x - min(x)) / (max(x) - min(x)).

```
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))

df
```

```
##            a        b         c  d
## 1  0.0000000 1.000000 0.0000000 NA
## 2  0.1111111 1.111111 0.1111111 NA
## 3  0.2222222 1.222222 0.2222222 NA
## 4  0.3333333 1.333333 0.3333333 NA
## 5  0.4444444 1.444444 0.4444444 NA
## 6  0.5555556 1.555556 0.5555556 NA
## 7  0.6666667 1.666667 0.6666667 NA
## 8  0.7777778 1.777778 0.7777778 NA
## 9  0.8888889 1.888889 0.8888889 NA
## 10 1.0000000 2.000000 1.0000000 NA
```

```
section1 <- function(x){
  x <- (x - min(x)) / (max(x) - min(x))
}
section1(df$a)
section1(df$b)
section1(df$c)
section1(df$d)

df
```

```
##            a        b         c  d
## 1  0.0000000 1.000000 0.0000000 NA
```

```
## 2  0.1111111 1.111111 0.1111111 NA
## 3  0.2222222 1.222222 0.2222222 NA
## 4  0.3333333 1.333333 0.3333333 NA
## 5  0.4444444 1.444444 0.4444444 NA
## 6  0.5555556 1.555556 0.5555556 NA
## 7  0.6666667 1.666667 0.6666667 NA
## 8  0.7777778 1.777778 0.7777778 NA
## 9  0.8888889 1.888889 0.8888889 NA
## 10 1.0000000 2.000000 1.0000000 NA
```

b Improve the below example code for the analysis of protein drug interactions by abstracting the main activities in your own new function.

Q1.

The read.pdb() function returns a dataframe object given input (PDB ID).

Q2.

The trim.pdb() function selects a specific subset from the larger dataframe returned by read.pdf().

Q3.

The input parameter sse controls the marginal black and gray rectangles in the plots. Not including it would turn that feature off (ex. plotb3(s1.b, typ="l", ylab="Bfactor")). SSE represents secondary structure element and it basically annotates the positions of major secondary structure elements in the margins of the plot.

Q4.

A better plot to compare different proteins would be to plot all 3 connected scatterplots on the same graph, making it easier to compare them to each other. It would also be beneficial if we were able to obtain a quantifiable value for each of these scatterplots that allowed us to compare the proteins without relying on our subjective graphical analysis.

Q5.

The dendrogram plot allows us to compare all three proteins to each other and quantify the comparison,allowing us to see that protein 2 and protein 3 are more similar to each other in their Bfactor trends.

```
# Can you improve this analysis code?
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

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

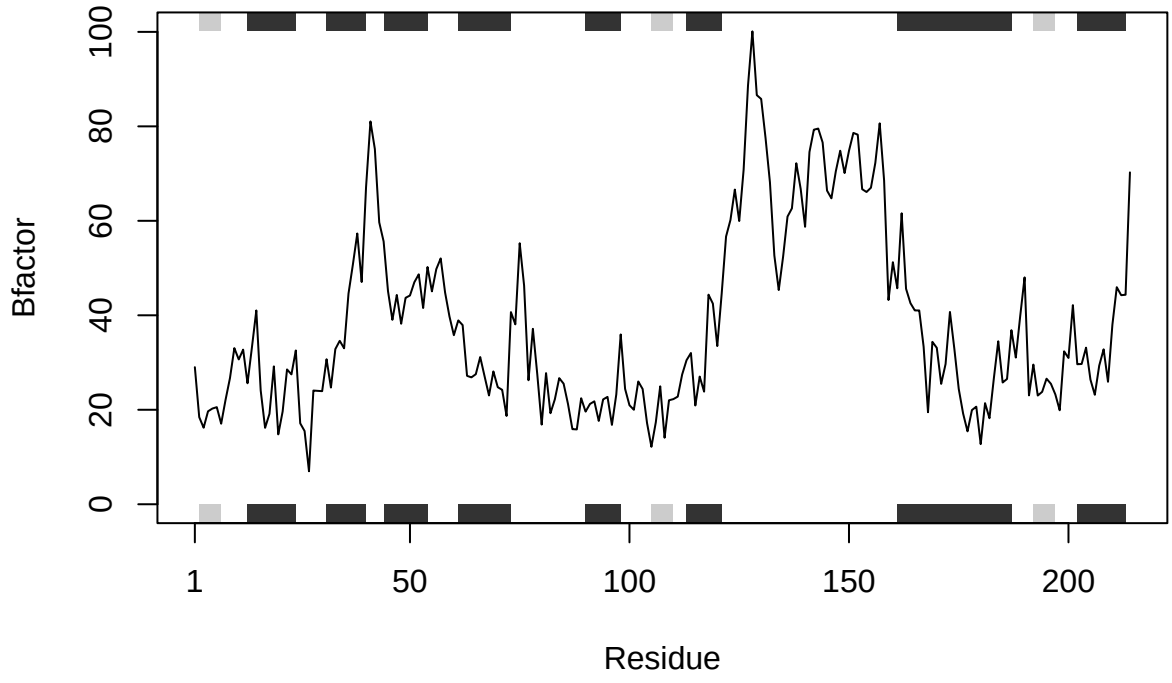
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

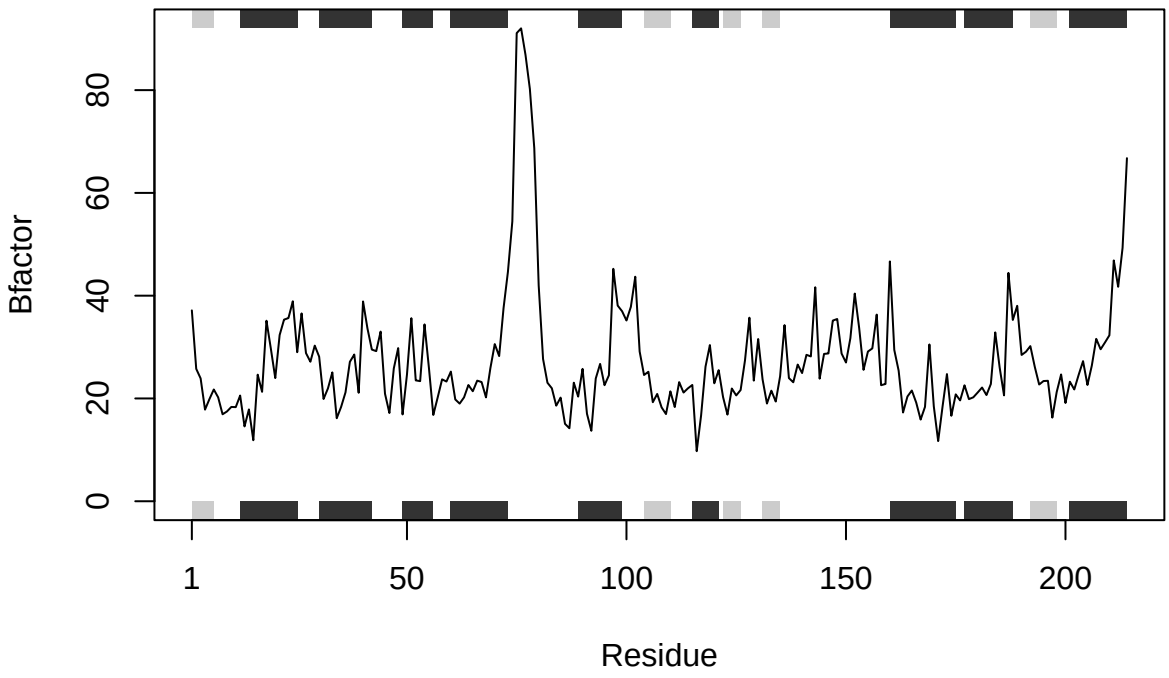
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

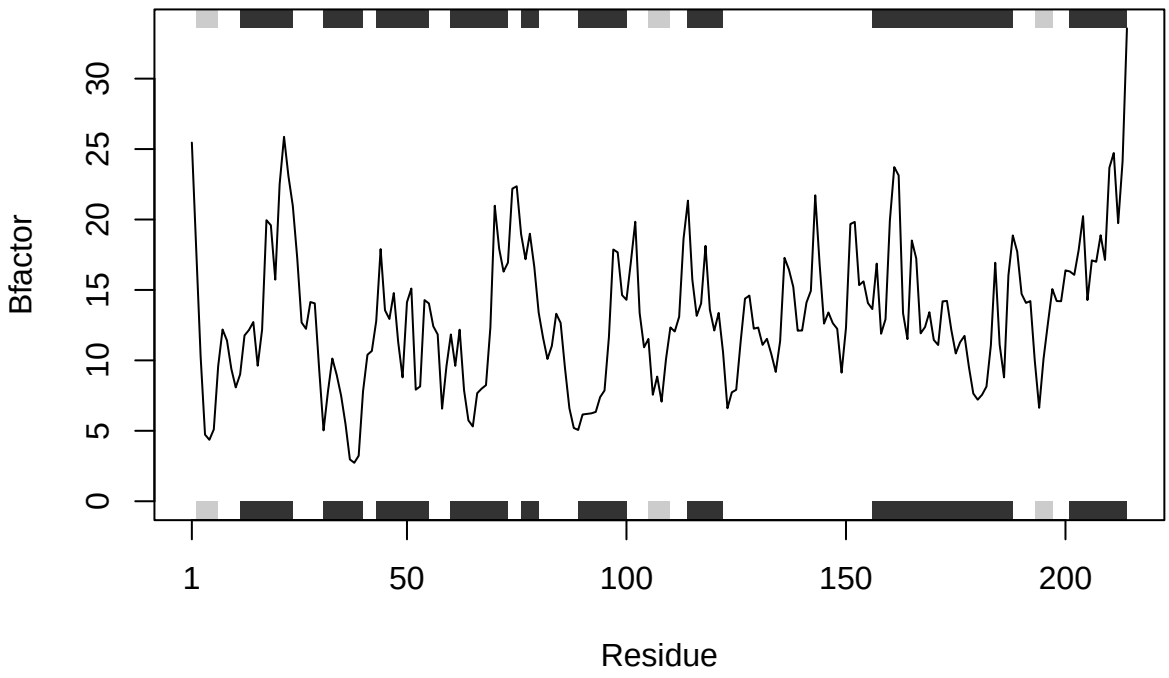
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

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")
```

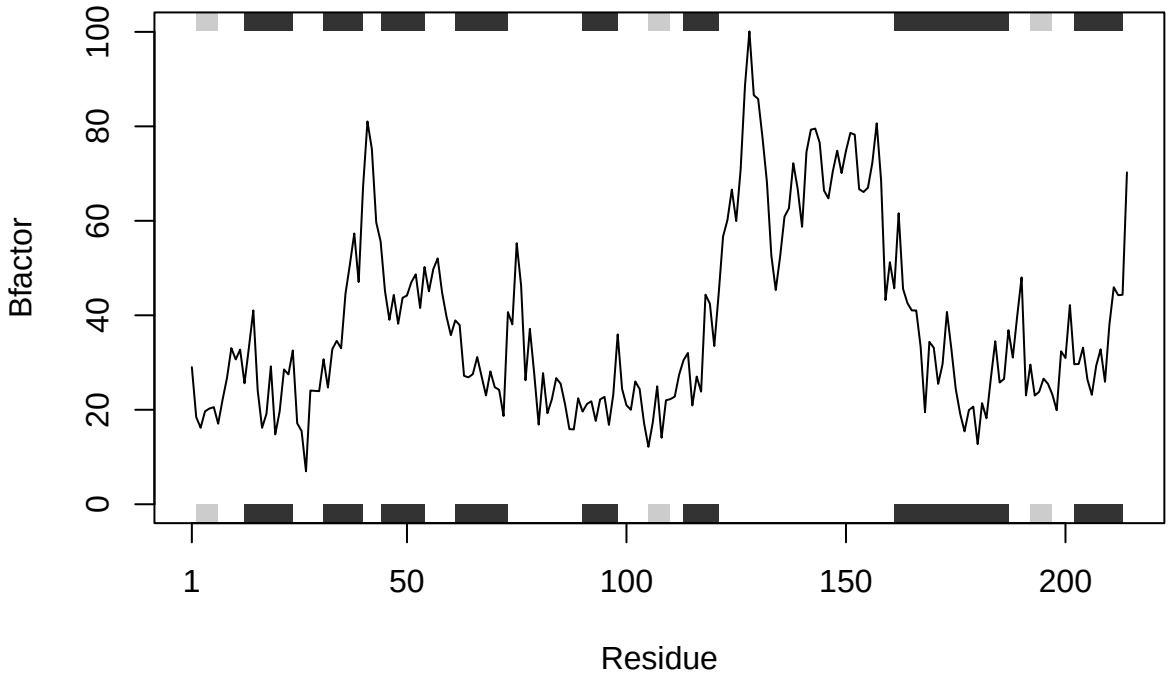



```
plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")
```



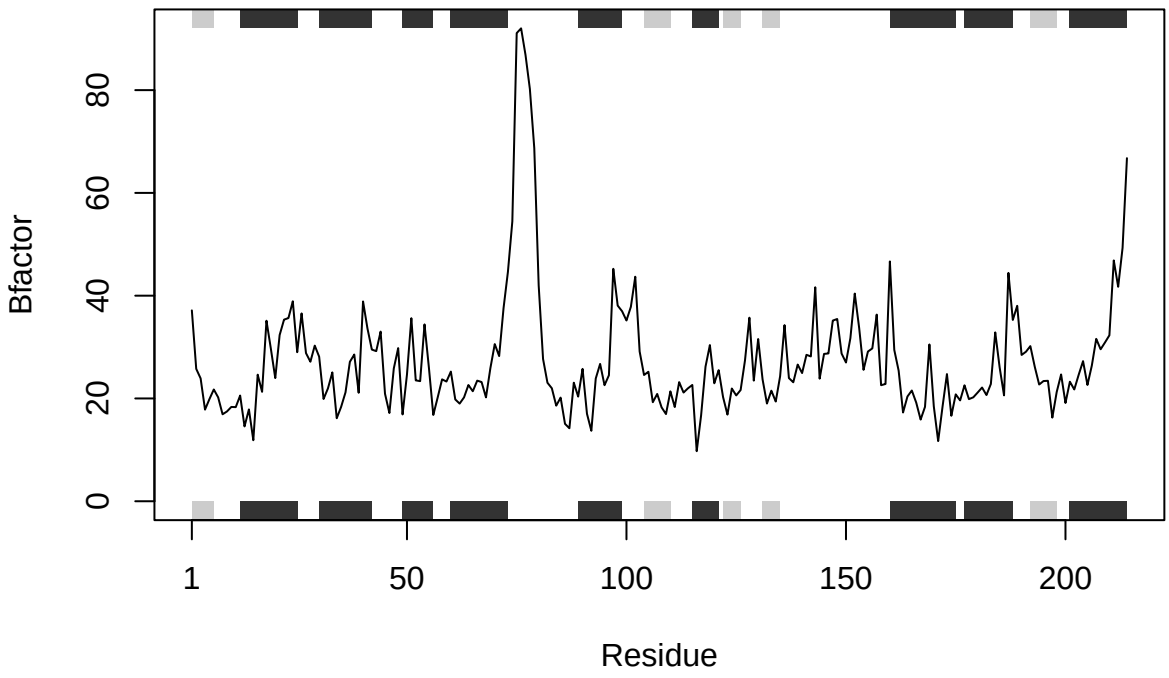

```
#Now create function to simplify

#documentation:
#   input is the specification of file to be read (can be PDB ID)
#   the function creates a plot given a PDB ID to assist in analysis of protein
#       drug interactions (compares residue to Bfactor)
#   output is the plot itself
protein.drug <- function(x){
  #read.pdb
  s <- read.pdb(x)
  #trim.pdb
  s.chainA <- trim.pdb(s, chain="A", elety="CA")
  #set s.b
  s.b <- s.chainA$atom$b
  #plot
  plotb3(s.b, sse=s.chainA, typ="l", ylab="Bfactor")
}

protein.drug("4AKE")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/n5/
## q44f96hn0252pn0ysv1c7rt40000gn/T//RtmpukRU0W/4AKE.pdb exists. Skipping download
```




```
protein.drug("1AKE")
```

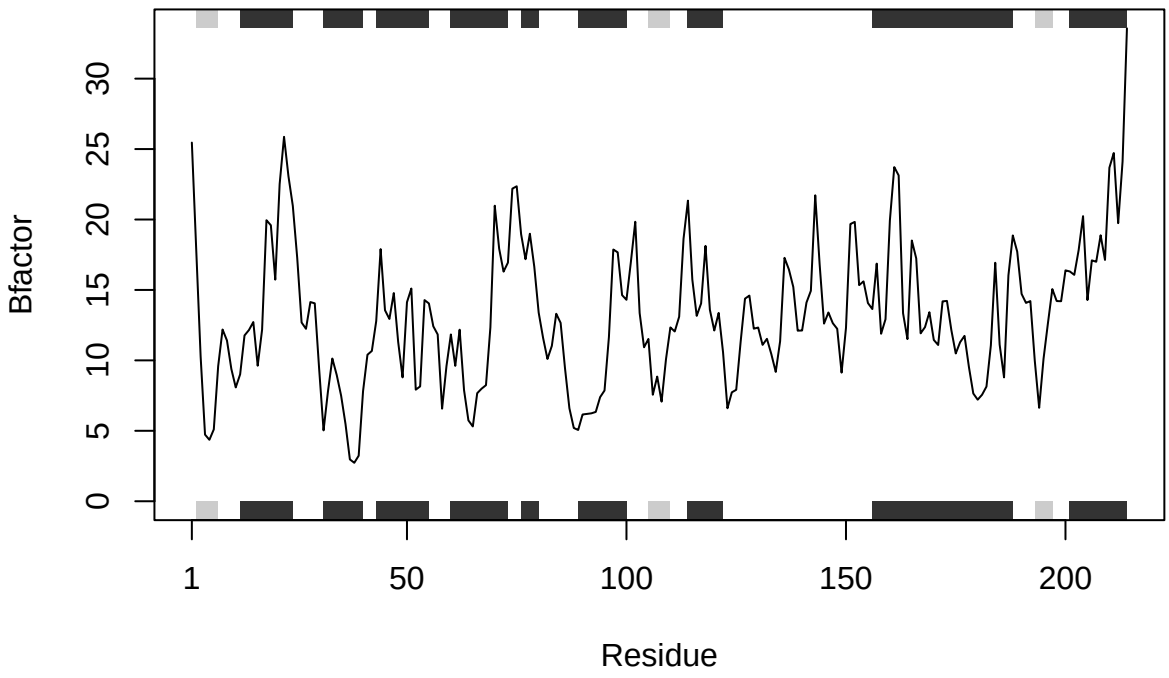
```
##   Note: Accessing on-line PDB file

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/n5/
## q44f96hn0252pn0ysv1c7rt40000gn/T//RtmpukRU0W/1AKE.pdb exists. Skipping download

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



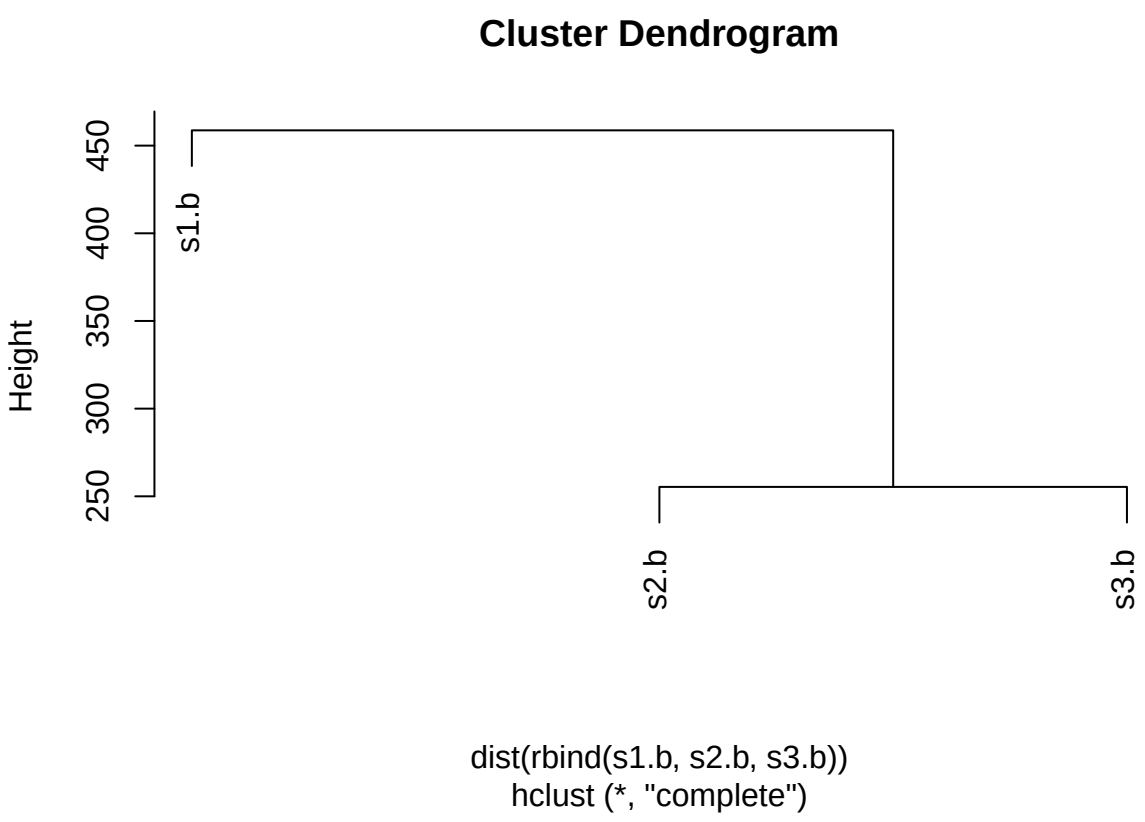

```
protein.drug("1E4Y")
```

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/n5/
## q44f96hn0252pn0ysv1c7rt40000gn/T//RtmpukRU0W/1E4Y.pdb exists. Skipping download
```




```
#It works!

#Q5. creating dendrogram plot

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


Cluster Dendrogram
Height
450
400
350
300
250
s1.b
s2.b
s3.b
dist(rbind(s1.b, s2.b, s3.b))
hclust (*, "complete")

“‘