HW 6

Stefanie Hodapp (PID: A53300084)

10/20/2021

Make dataframe

```
df <- data.frame(a=1:10, b=seq(200,400,length=10),c=11:20,d=NA)
df
```

```
##     a        b  c  d
## 1   1 200.0000 11 NA
## 2   2 222.2222 12 NA
## 3   3 244.4444 13 NA
## 4   4 266.6667 14 NA
## 5   5 288.8889 15 NA
## 6   6 311.1111 16 NA
## 7   7 333.3333 17 NA
## 8   8 355.5556 18 NA
## 9   9 377.7778 19 NA
## 10 10 400.0000 20 NA
```

Run analysis code

```
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
```

```
##  [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
##  [8] 0.7777778 0.8888889 1.0000000
```

```
df$b
```

```
##  [1] 1.000000 1.111111 1.222222 1.333333 1.444444 1.555556 1.666667 1.777778
##  [9] 1.888889 2.000000
```

```
df$c
```

```
##  [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
##  [8] 0.7777778 0.8888889 1.0000000
```

```
df$d
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA
```

Make new data frame and run the analysis code without errors

```
df_new <- data.frame(a=1:10, b=seq(200,400,length=10),c=11:20,d=NA)
df_new
```

```
##     a        b  c  d
## 1   1 200.0000 11 NA
## 2   2 222.2222 12 NA
## 3   3 244.4444 13 NA
## 4   4 266.6667 14 NA
## 5   5 288.8889 15 NA
## 6   6 311.1111 16 NA
## 7   7 333.3333 17 NA
## 8   8 355.5556 18 NA
## 9   9 377.7778 19 NA
## 10 10 400.0000 20 NA
```

```
df_new$a <- (df_new$a - min(df_new$a)) / (max(df_new$a) - min(df_new$a))
df_new$b <- (df_new$b - min(df_new$b)) / (max(df_new$b) - min(df_new$b))
df_new$c <- (df_new$c - min(df_new$c)) / (max(df_new$c) - min(df_new$c))
df_new$d <- (df_new$d - min(df_new$d)) / (max(df_new$d) - min(df_new$d))

df_new$a
```

```
##  [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
##  [8] 0.7777778 0.8888889 1.0000000
```

```
df_new$b
```

```
##  [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
##  [8] 0.7777778 0.8888889 1.0000000
```

```
df_new$c
```

```
##  [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
##  [8] 0.7777778 0.8888889 1.0000000
```

```
df_new$d
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA
```

Improve the analysis code by writing your own function

```
df_function <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
```

Try out the function on one of the columns of the columb on df_new

```
df_function(df_new$a)
```

```
##  [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
##  [8] 0.7777778 0.8888889 1.0000000
```

```
apply(df, 2, df_function)
```

```
##               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
```

Question 6:

Read in the PDB file. I am comm

```
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
```

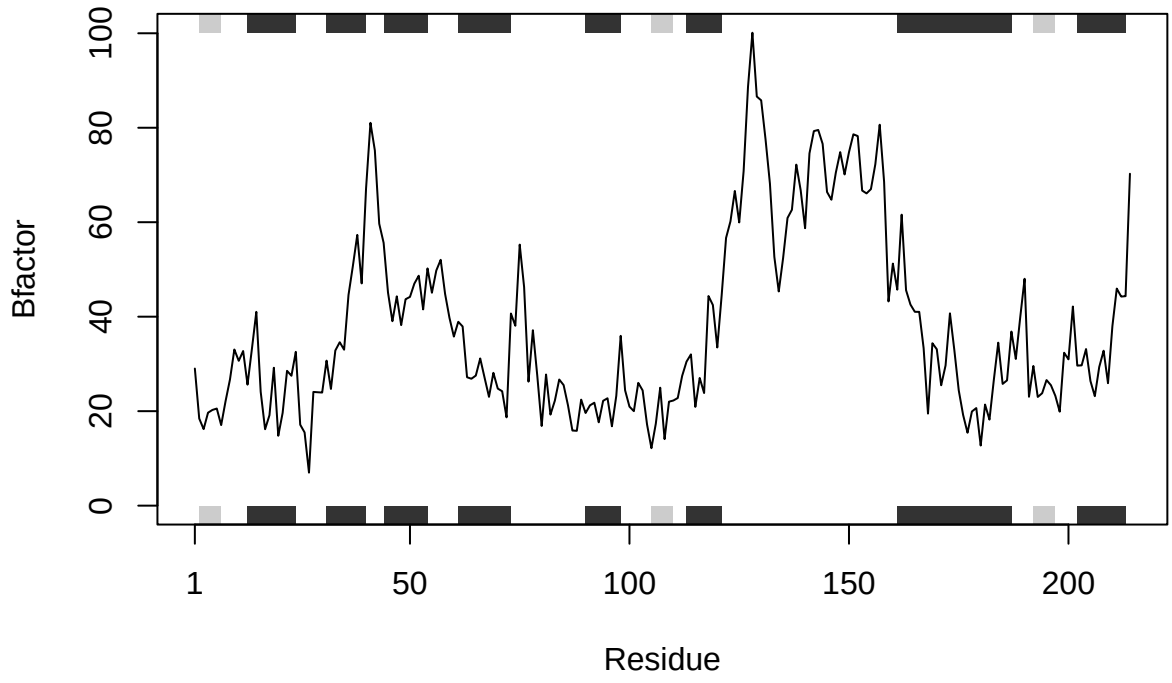
Produce a new smaller PDB object from the original file containing a subset of atoms using 'trim.pdb()'.

```
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")
```

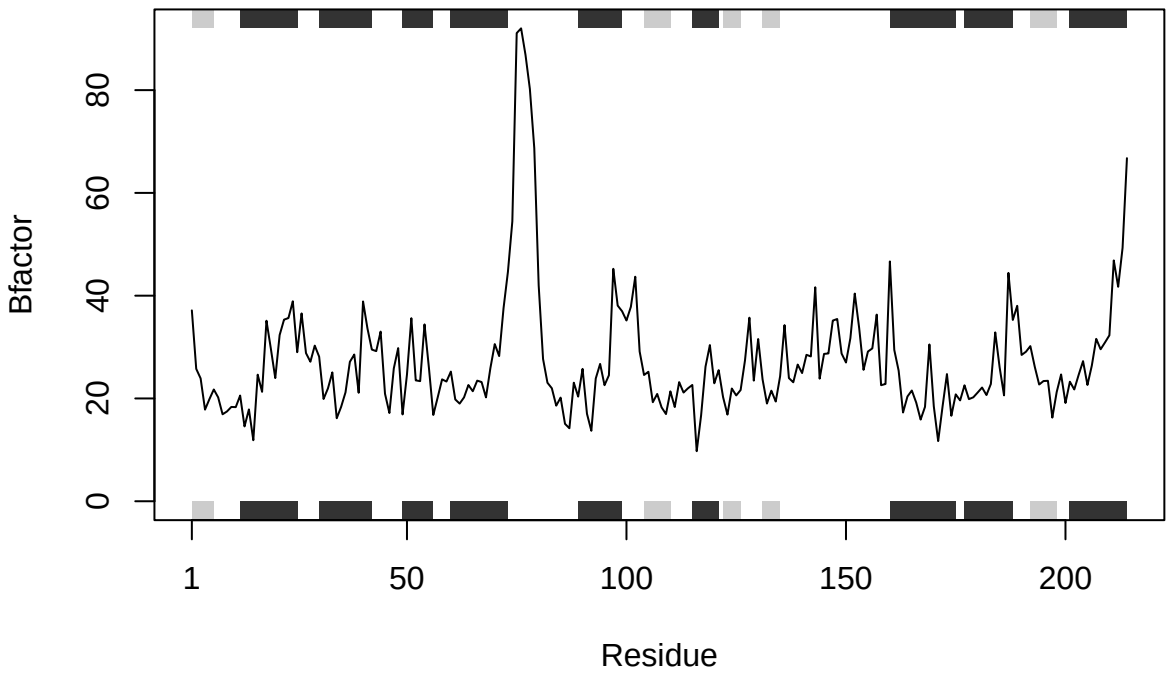
```
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
```

Draw a standard line plot with secondary structure in the marginal regions.

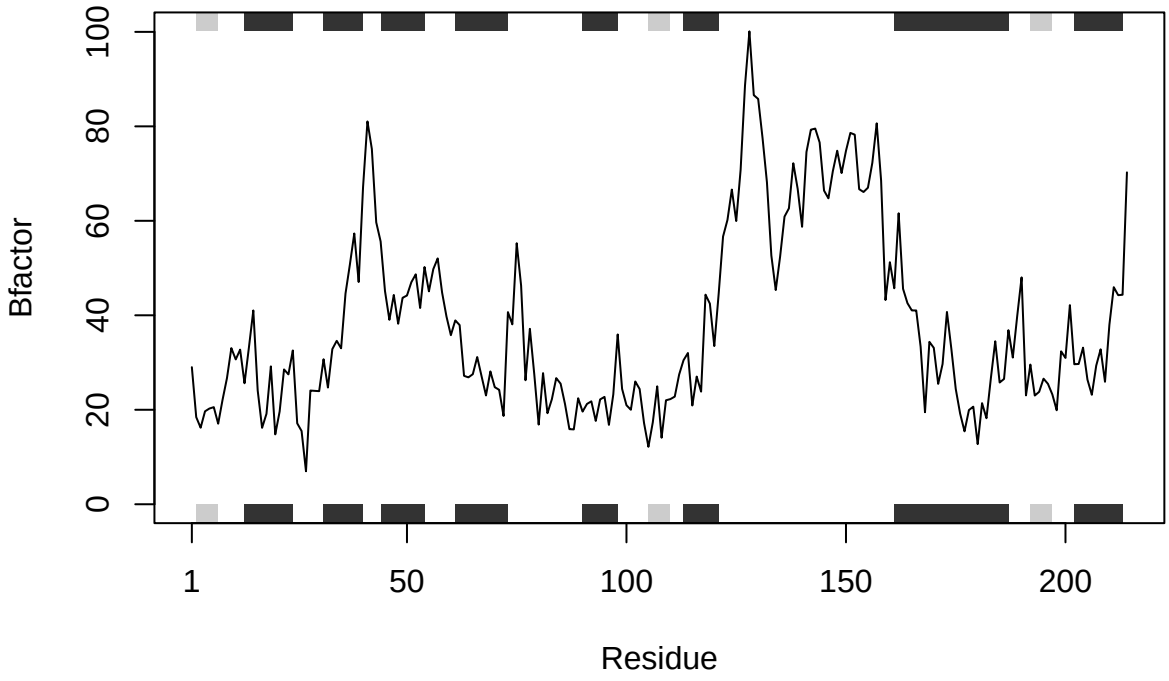
```
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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



Generalize the original code above to work with any set of input protein structures

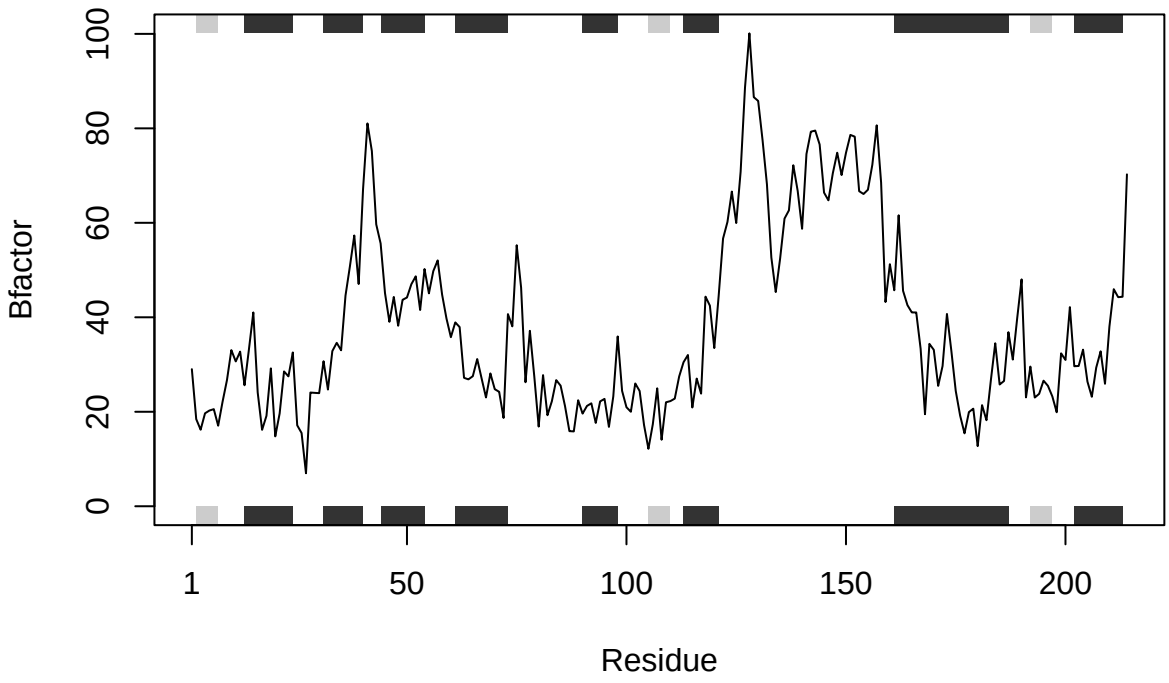
For a given PDB code, the prot_drug_interact function will read-in the PDB file and store this as "protein". The next argument will use "protein" as the input and produce a new smaller PDB object consisting of chain "A", which will be stored as "protein.chainA". Next, the function will pull out temperature factor "b" data from the "protein.chainA" file and store this as "protein.b". Finally, the data in "protein.b" will be plotted as a line plot with secondary structure "protein.chainA".

```
prot_drug_interact <- function(PDB_code) {
  protein <- read.pdb(PDB_code)
  protein.chainA <- trim.pdb(protein, chain="A", elety="CA")
  protein.b <- protein.chainA$atom$b
  plotb3(protein.b, sse=protein.chainA, typ="l", ylab="Bfactor")
}
```

Test function on protein 4AKE

```
prot_drug_interact("4AKE")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/2f/
## 0n5x4j_j68x9nzk4j938zzsw0000gn/T//Rtmprth7LR/4AKE.pdb exists. Skipping download
```



Apply this function to all proteins in a given list using 'lapply()'. I will make a list of the 3 proteins to test out the lapply function and call this "prot_list".

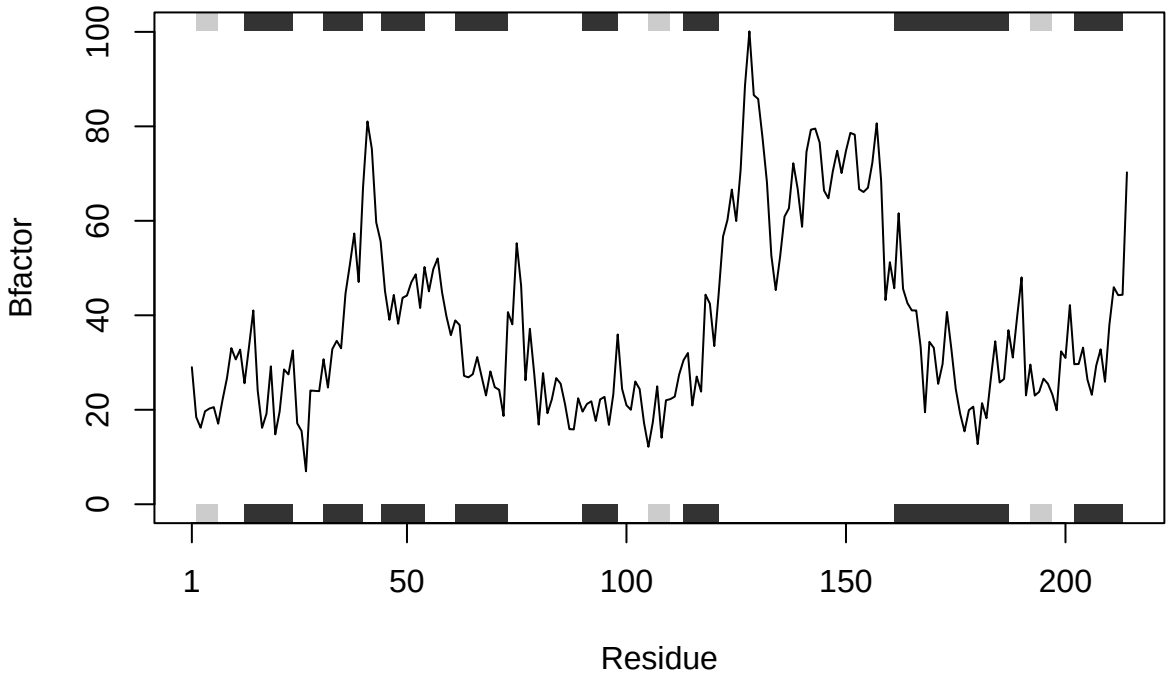
```
prot_list <- c("4AKE", "1AKE", "1E4Y")
print(prot_list)
```

```
## [1] "4AKE" "1AKE" "1E4Y"
```

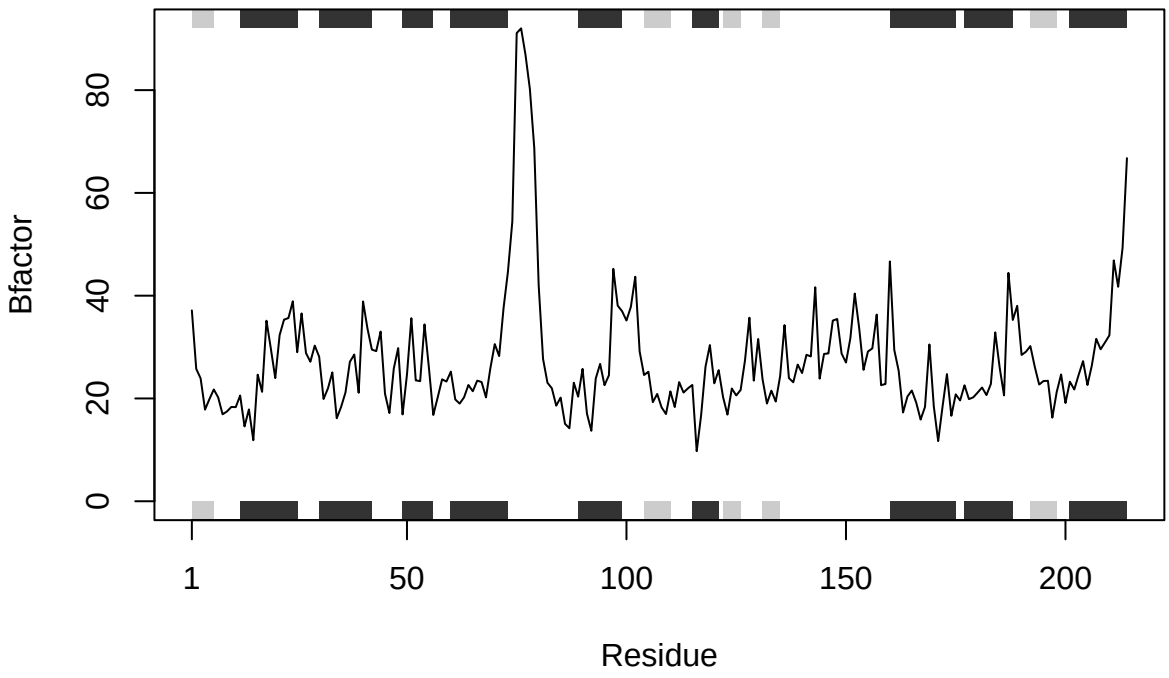
The "lapply()" function specifies the protein list followed by the function to be applied to each element in the list.

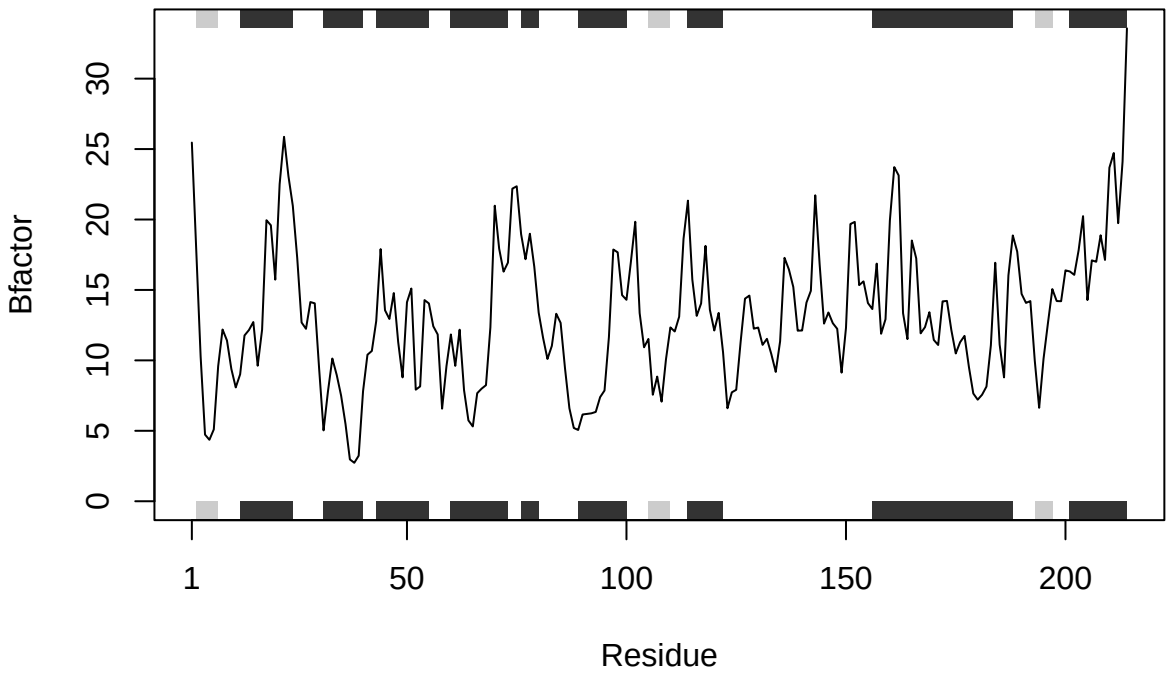
```
lapply(prot_list, prot_drug_interact)
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/2f/
## 0n5x4j_j68x9nzk4j938zzsw0000gn/T//Rtmprth7LR/4AKE.pdb exists. Skipping download

##   Note: Accessing on-line PDB file

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/2f/
## 0n5x4j_j68x9nzk4j938zzsw0000gn/T//Rtmprth7LR/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): /var/folders/2f/
## 0n5x4j_j68x9nzk4j938zzsw0000gn/T//Rtmprth7LR/1E4Y.pdb exists. Skipping download
```


Bfactor
0
20
40
60
80
1
50
100
150
200
Residue




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```