Lab 6 Question 6

Groot (PID: A15485151)

10/19/2021

```
#load bio3d!
library(bio3d)

# use read.pdb() to read the protein PDB file
s1 <- read.pdb("4AKE") # kinase with drug
```

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

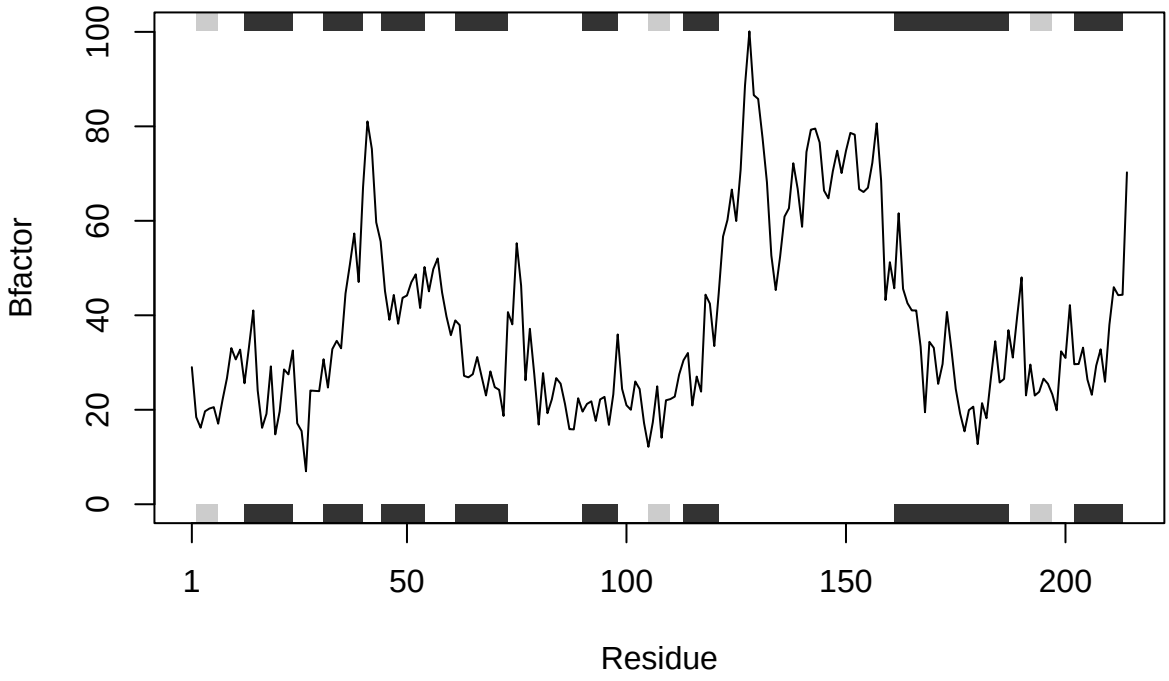
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

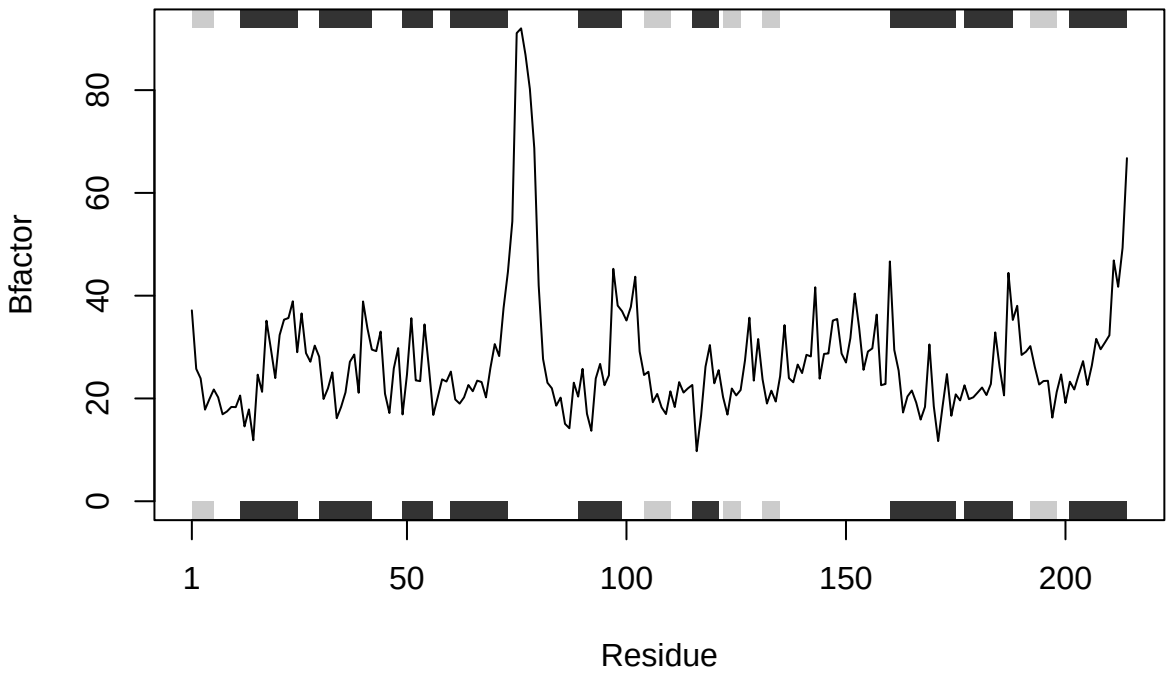
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

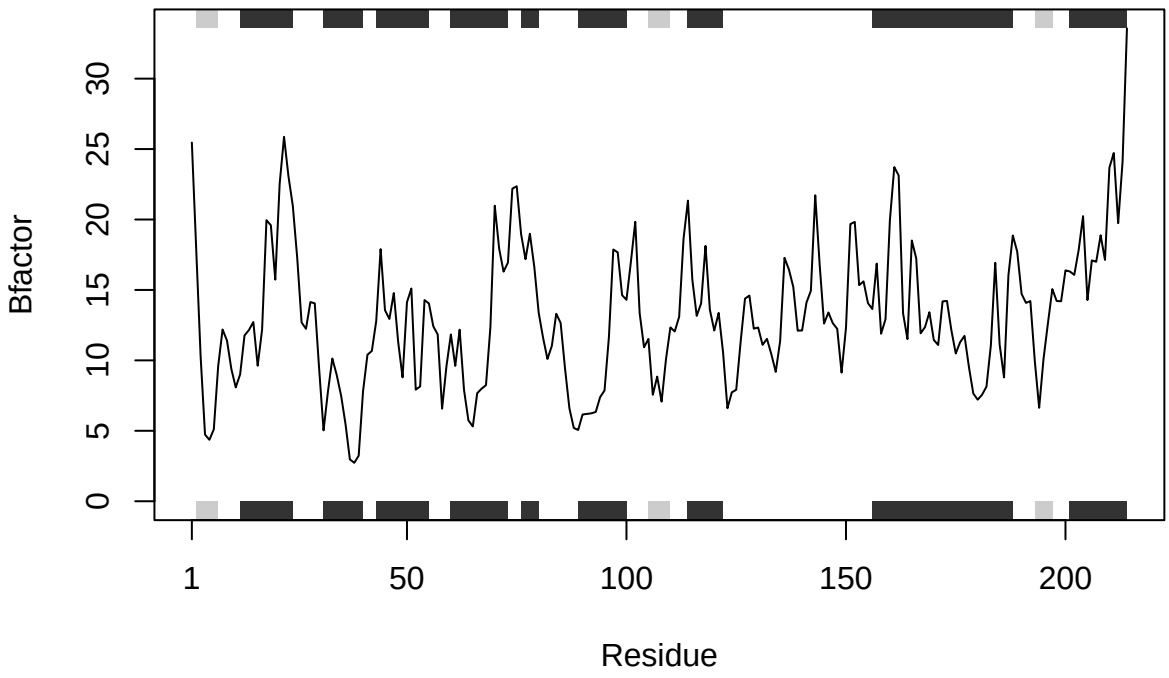
```
# Use trim.pdb() function to trim the protein PDB file to only entail chain A
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

# Use the $ to selectively assign parts of the trimmed protein database file (rows 'atom' and column 'b
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b

# Use plotb3() function with the argument ylab = "Bfactor"(w/ plot type = "l" and sse = trimmed protein
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")
```

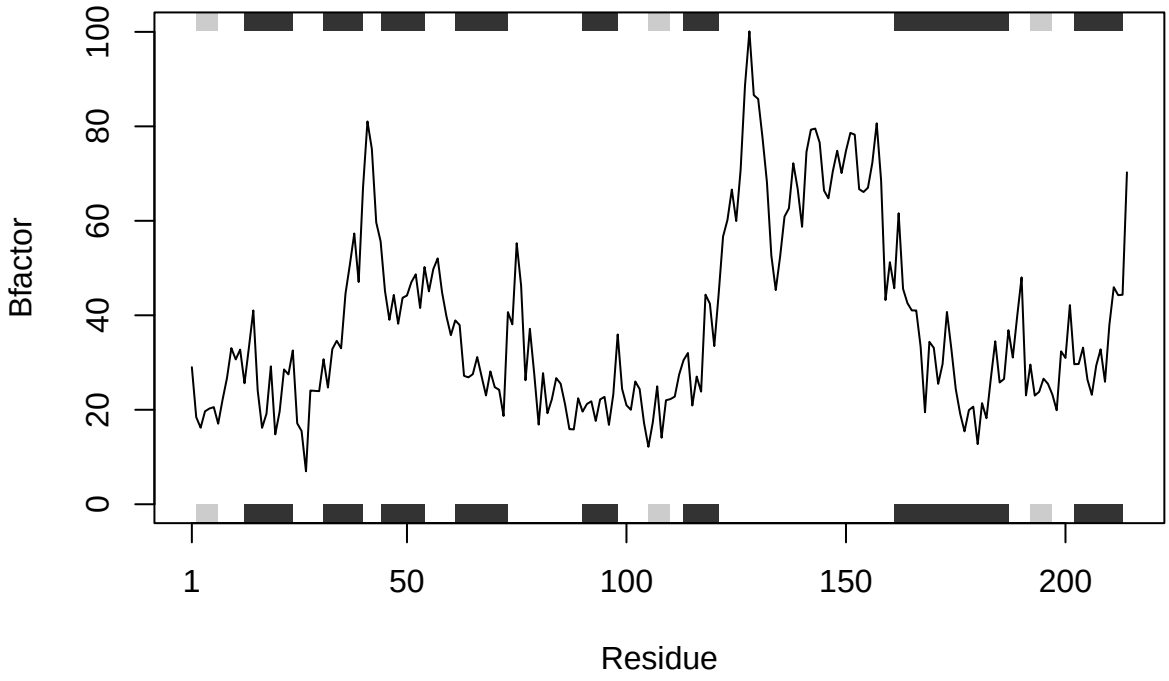


Simplify to avoid calculation **duplication**!

```
# x <- "4AKE" or name of protein PDB file (in "")

s1 <- read.pdb("4AKE")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/jt/
## v6s9jtfd73n038k5kvwrx47w0000gn/T//RtmpSiNXan/4AKE.pdb exists. Skipping download
```

```
# a1 <- s1.chainA
a1 <- trim.pdb(s1, chain="A", elety="CA")

#b1 <- s1.b
b1 <- a1$atom$b
plotb3(b1, sse=a1, typ="l", ylab="Bfactor")
```



Now use these as the body of the function

```
# The inputs of this function is the protein PDB data file (in"")
# To use the function bfplot(), input the names (in "") of the protein PDB data file for x within the p
# Don't forget to load bio3d!!
library(bio3d)
bfplot <- function(x){
  # x <- protein PDB file name (with "")
s1 <- read.pdb(x)

# a1 <- s1.chainA
a1 <- trim.pdb(s1, chain="A", elety="CA")

#b1 <- s1.b
b1 <- a1$atom$b
plotb3(b1, sse=a1, typ="l", ylab="Bfactor")

}
```

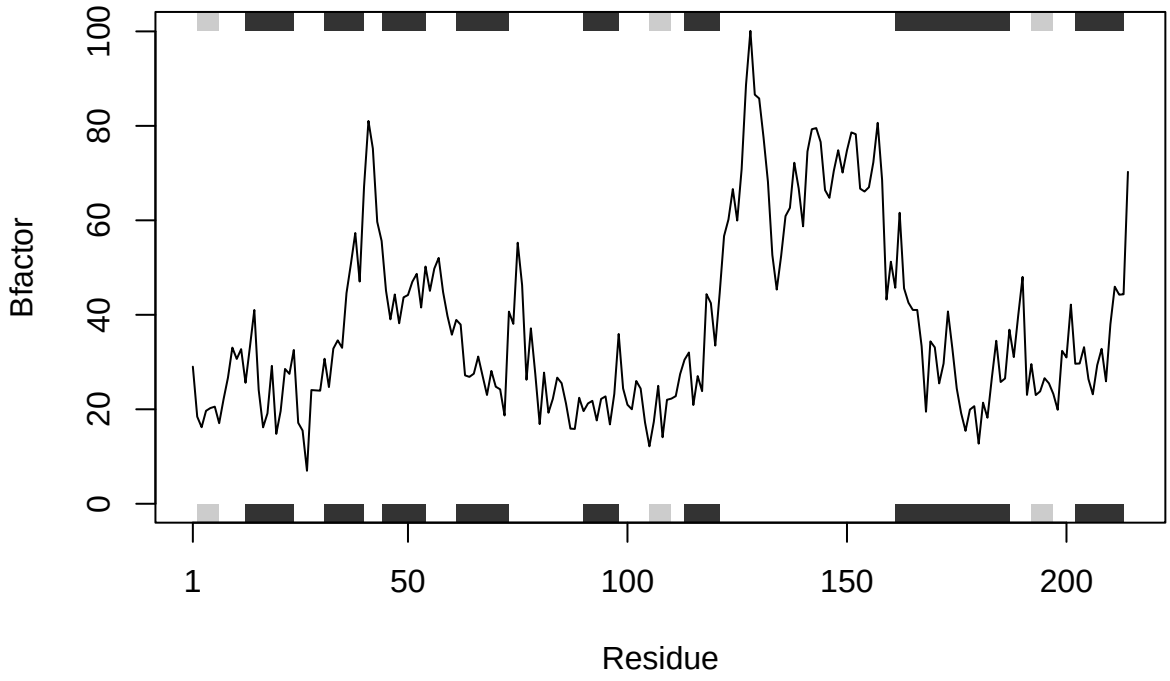
The **output** of this function is the "residue vs. Bfactor" plots of the input protein PDB data file

Check if the function works! :)

```
bfplot("4AKE")
```

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/jt/
## v6s9jtfd73n038k5kvwrx47w0000gn/T//RtmpSiNXan/4AKE.pdb exists. Skipping download
```

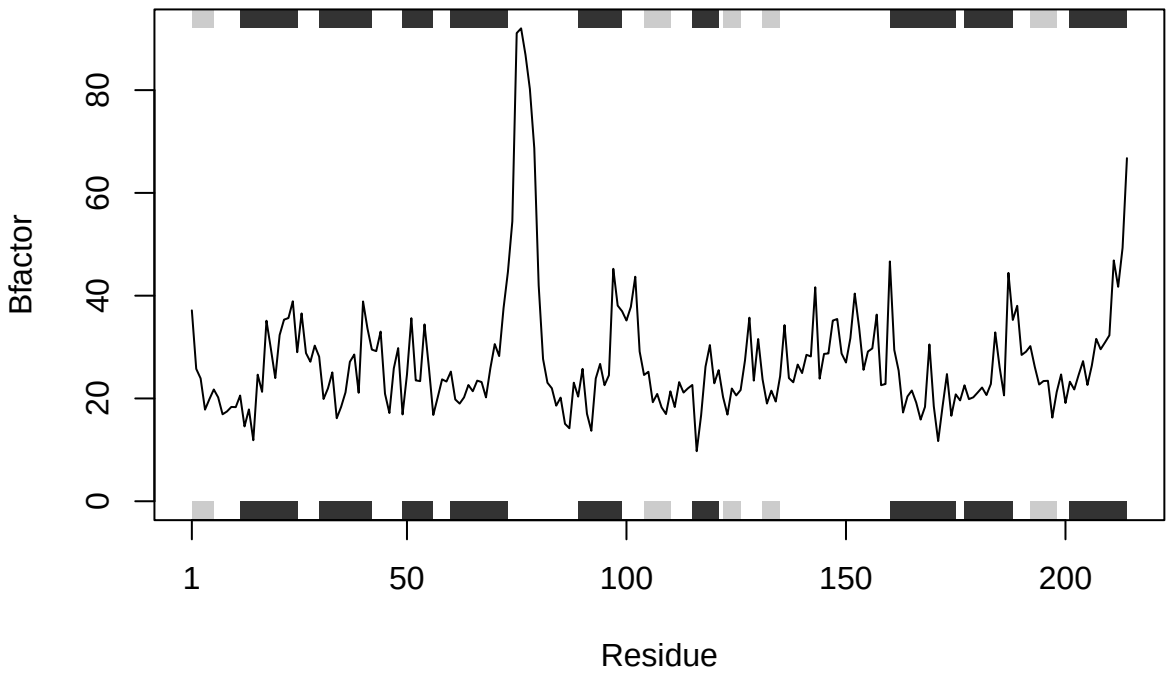



```
bfplot("1AKE")
```

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

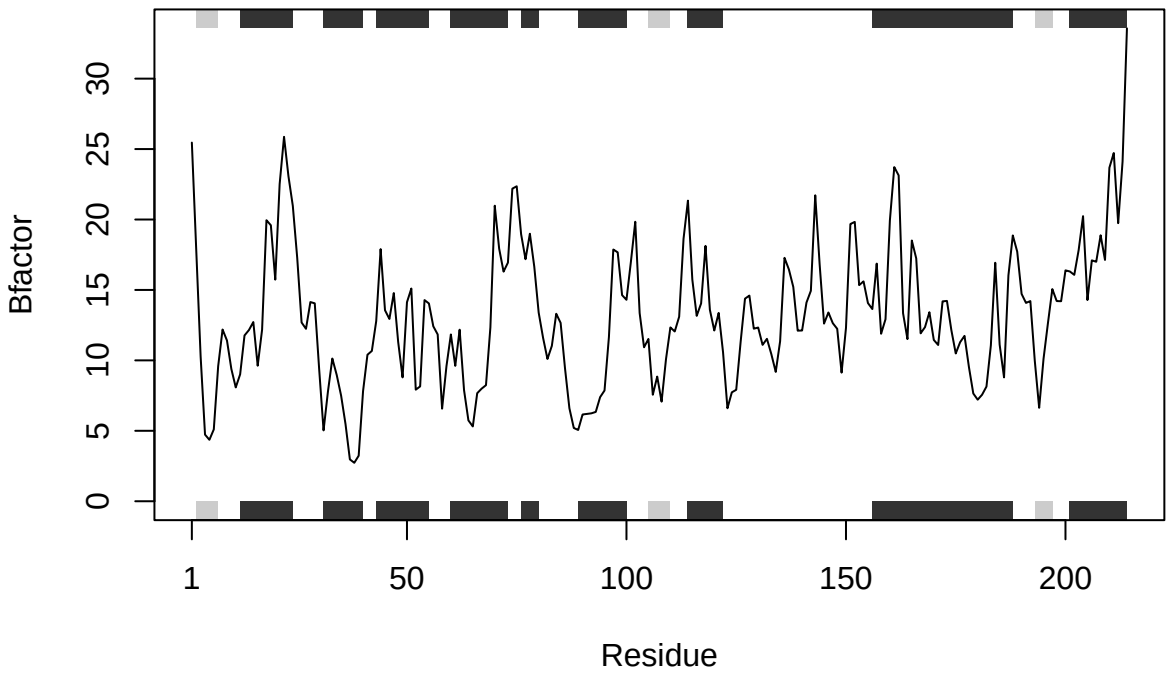
```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/jt/
## v6s9jtfd73n038k5kvwrx47w0000gn/T//RtmpSiNXan/1AKE.pdb exists. Skipping download
```

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




```
bfplot("1E4Y")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/jt/
## v6s9jtfd73n038k5kvwrx47w0000gn/T//RtmpSiNXan/1E4Y.pdb exists. Skipping download
```


Bfactor
0
5
10
15
20
25
30
1
50
100
150
200
Residue