Class 6: Homework

Arshiya Zarmahd (PID: A16247996)

Question 6

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

Original Function

```
# 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
```

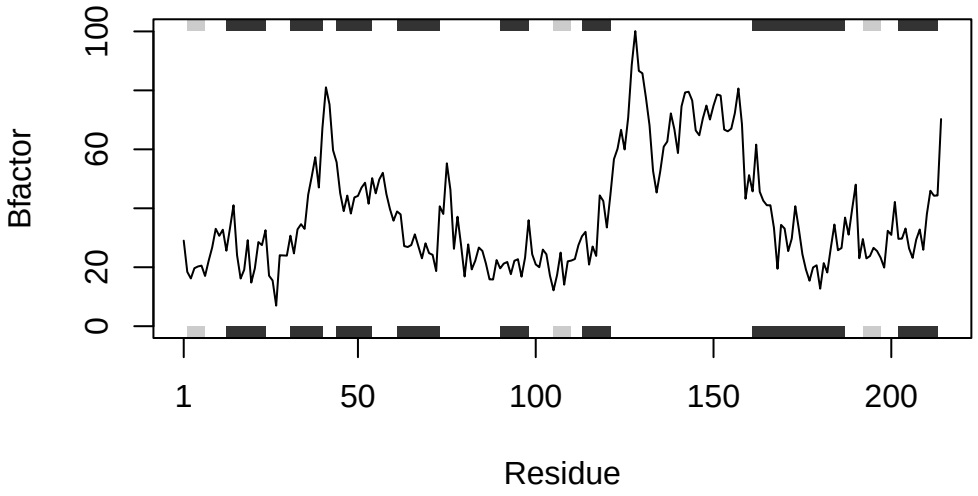
```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```

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

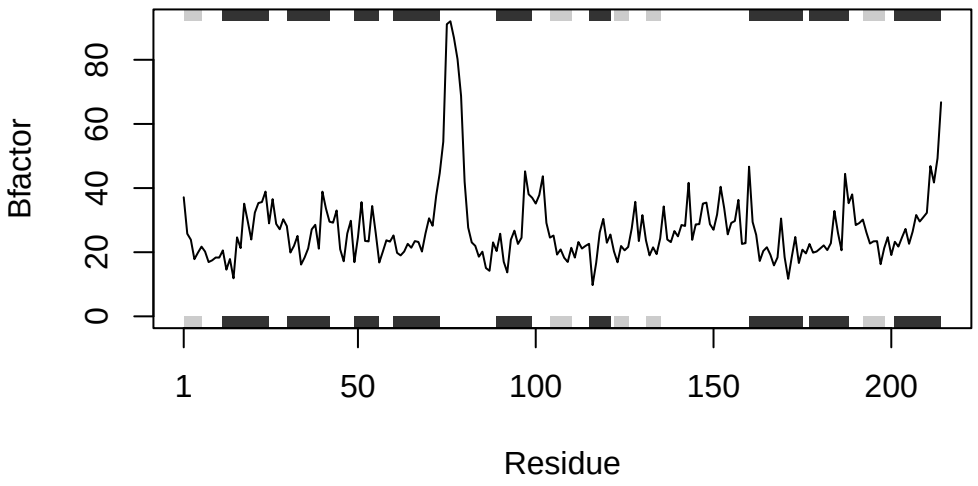
```
Note: Accessing on-line PDB file
```

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

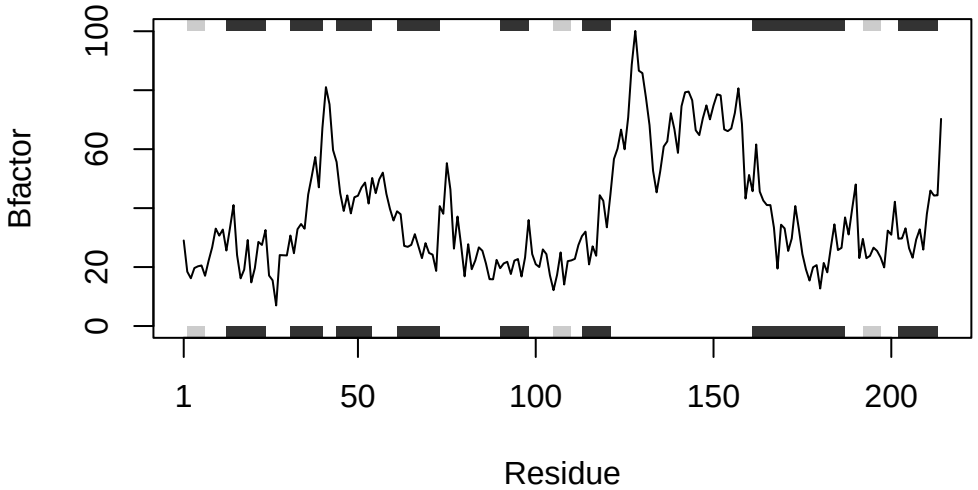
```
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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



Re-Write of Function

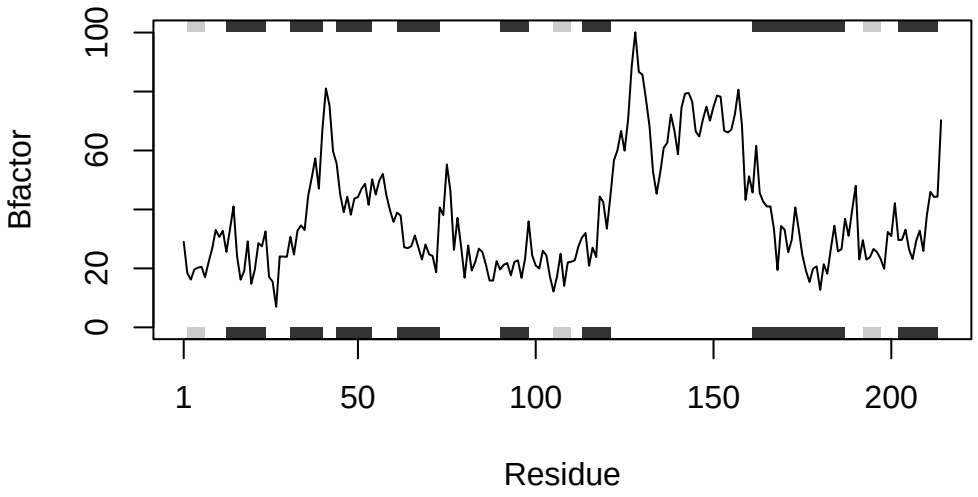
```
library(bio3d)
```

```
# I assigned the function placeholder as plots. Plots will be used to connect
# the code to the PDB file name.
plots <- function(plots) {
# I then read and trimmed the PDB file, under the variable plots. The trim
# included only A chains, as the original code was looking for. Therefore,
# there is no need to specify chainA in the next code, because all the details
# of the code are given to the variable pdb. Read is embedded in trim so that
# the trim code will read the proper pdb file and trim it with the proper
# specifications.
    pdb <- trim.pdb(read.pdb(plots), chain="A", elety="CA")
# This code uses bio3d to plot the b_factors of the above PDB file with the
# specifications of the previous code. The original function took into
# consideration the secondary structure elements in the PDB file, so the same
# was used here, but sse was given to pdb, as pdb holds all the information
# necessary because of the previous line of code. The typ and ylab are specific
# code for the plot itself, rendering a line graph with y-axis labeling.
    plotb3(pdb$atom$b, sse=pdb, typ="l", ylab="Bfactor")
}

plots("4AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/5v/hhjnwx4s629_rjtb6pksdt980000gn/T//RtmpOOZH9v/4AKE.pdb exists.
Skipping download
```

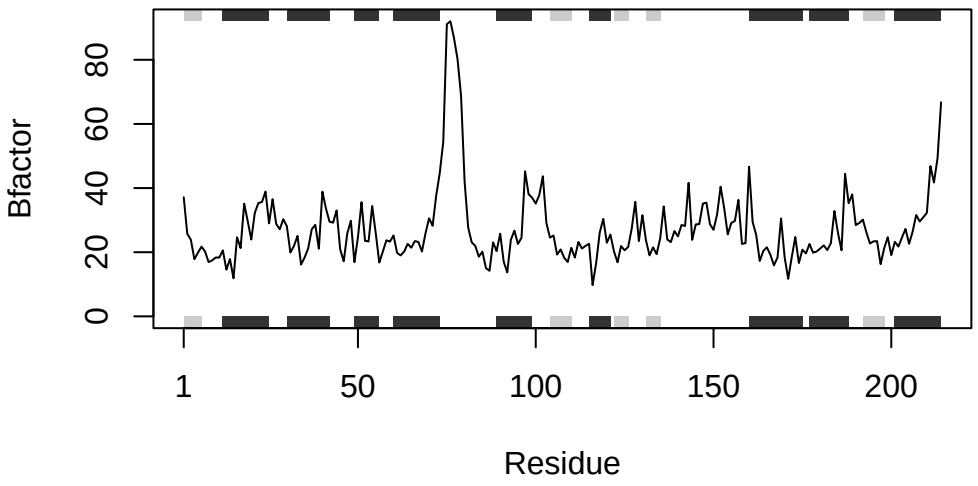



```
plots("1AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/5v/hhjnwx4s629_rjtb6pksdt980000gn/T//RtmpOOZH9v/1AKE.pdb exists.
Skipping download
```

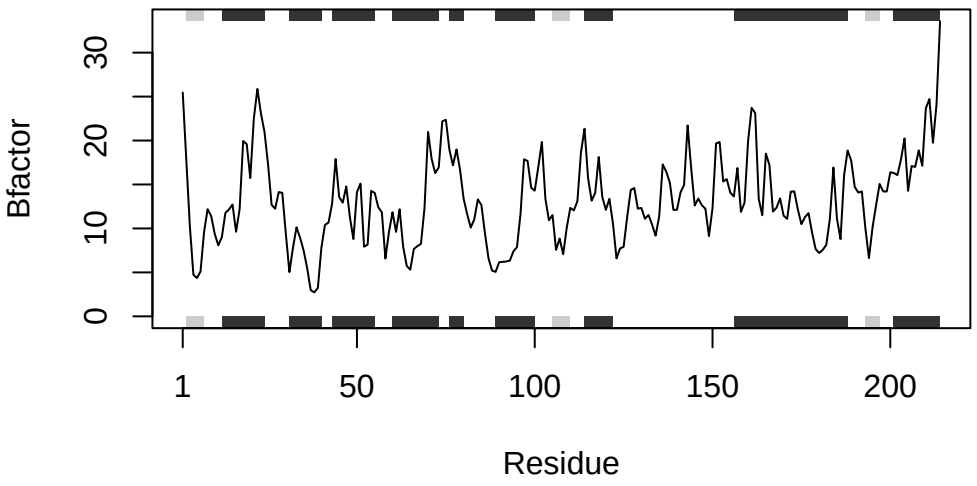
```
   PDB has ALT records, taking A only, rm.alt=TRUE
```




```
plots("1E4Y")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/5v/hhjnwx4s629_rjtb6pksdt980000gn/T//RtmpOOZH9v/1E4Y.pdb exists.
Skipping download
```


Bfactor
0
10
20
30
1
50
100
150
200
Residue