BIMM 143 Homework Class #6

5/2/23

Improving the Code

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

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

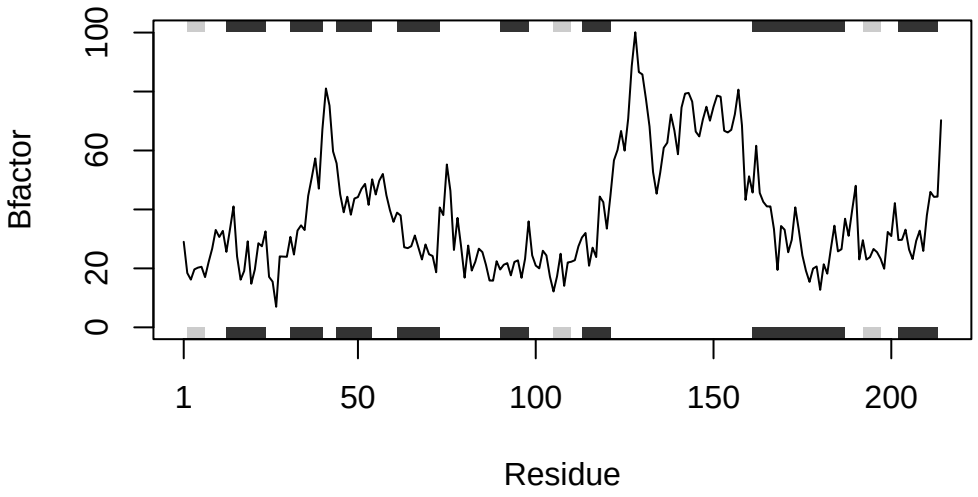
```
s2 <- read.pdb("1AKE")  # kinase no drug
```

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

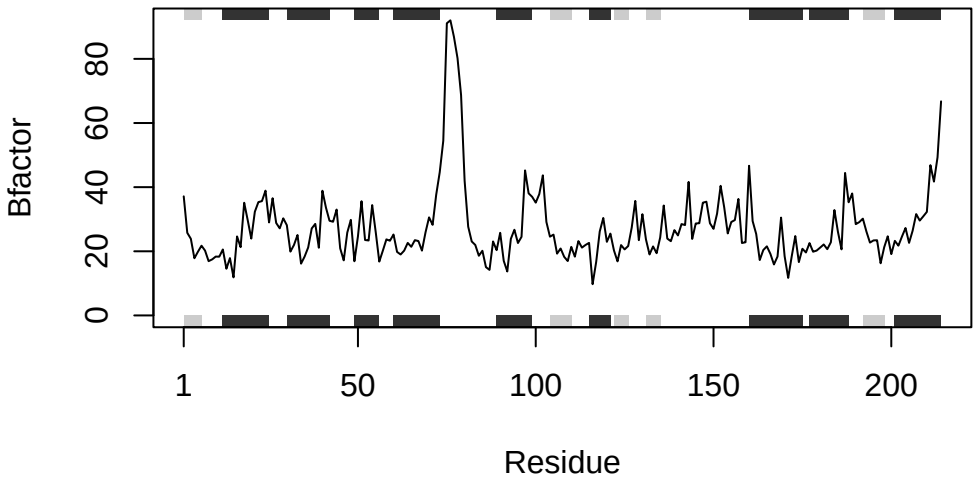
```
s3 <- read.pdb("1E4Y")  # kinase with drug
```

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

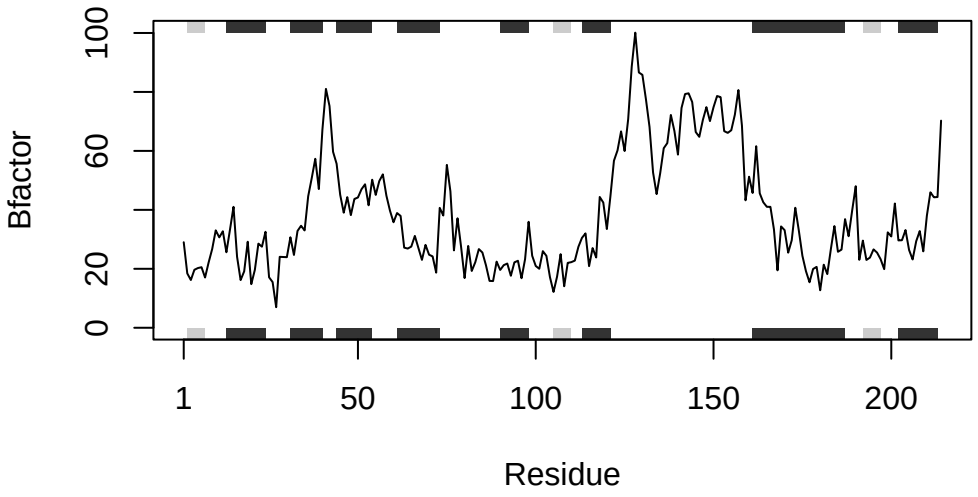
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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



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



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

To simplify the above code, we can create a `function` that combines all of the above steps.

The code generates an analysis of protein drug interactions by plotting the B factor (low B-factor values indicate a part of a structure that is well ordered, while large B-factor values indicate more flexibile parts of the structure) and residue values.

The input for the function is the 4 letter PDB code associated with a particular protein. The function generates a graph of the B-factor and residue values as its output.

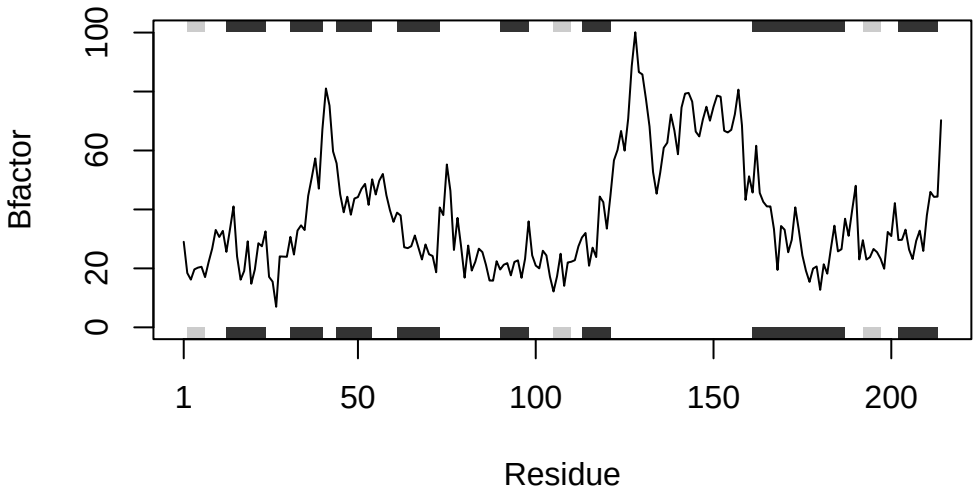
```
#first, the function is assigned to an object,
#"pbdstructure," with the variable 'x'
#denoting whatever 4 letter protein code
#is desired for input
pdbstructure <- function(x){
  #then, 'x' is assigned to the object "pdbcode"
  pdbcode <- x
  #the read.pdb function is used on pdbcode
  #to access the PDB website and read in the data
```

```
  #from a particular protein. This value
  #is assigned to object "pdb"
  pdb <- read.pdb(pdbcode)
  #the values in "pdb" are then trimmed
  #to a subset of atoms to give a smaller
  #object, named "chain"
  chain <- trim.pdb(pdb, chain="A", elety="CA")
  #finally, a scatter plot is generated using
  #the data stored under 'b' in the 'atom'
  #data section in "chain," the sse= plots
  #the secondary structure returned from read.pdb,
  #typ="l" specifies lines to be added to the
  #graph, and ylab= renames the y-axis to "Bfactors
  plotb3(chain$atom$b, sse=chain, typ="l", ylab="Bfactor")
}

#we can now call the function using pdbstructure,
#its assigned object
pdbstructure("4AKE")
```

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

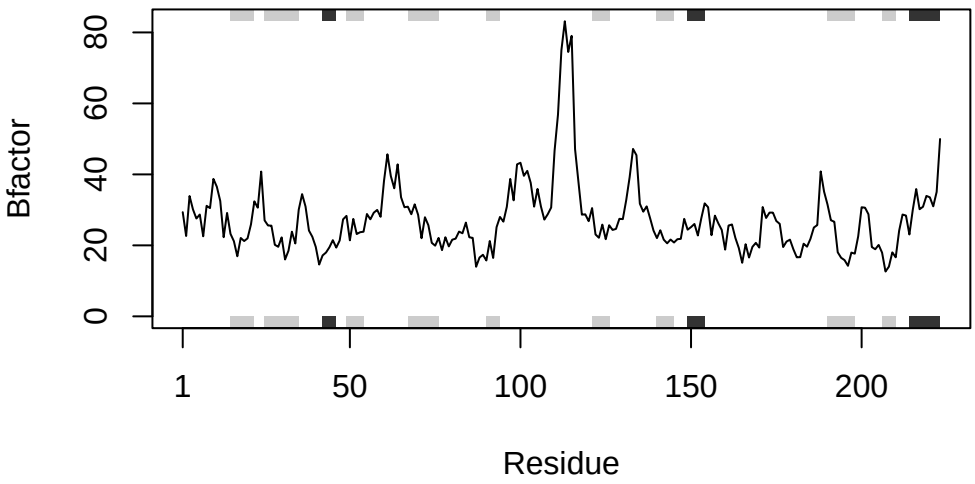
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/_4/yxx9x4dj6vq3dxtydfsf9vnh0000gn/T//RtmpsQPJoh/4AKE.pdb exists.
Skipping download
```



To use the function, one can input any PDB code associated with a protein as the variable of the function `pdbstructure`. The function can be used on any set of protein input structures, satisfying the requirements of the question.

```
pdbstructure("8G25")
```

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


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