class 6 : homework

Jimmi Nguyen

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

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

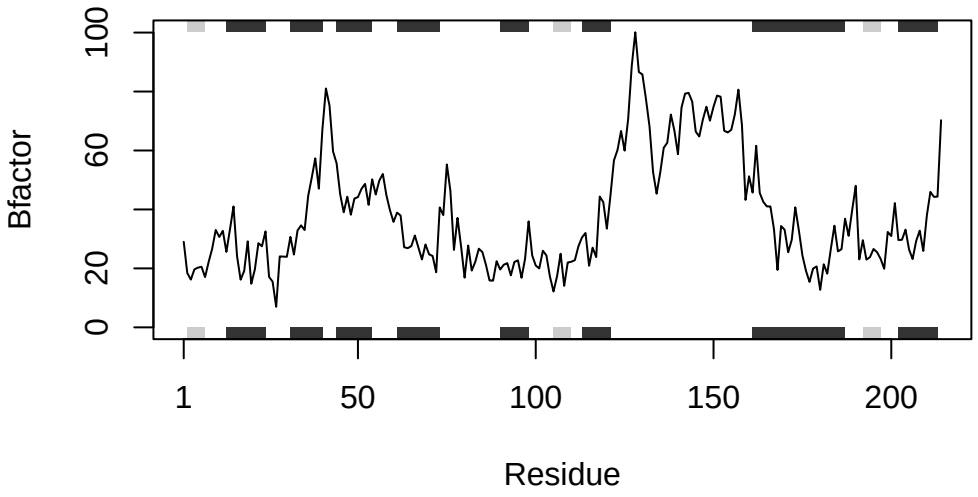
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

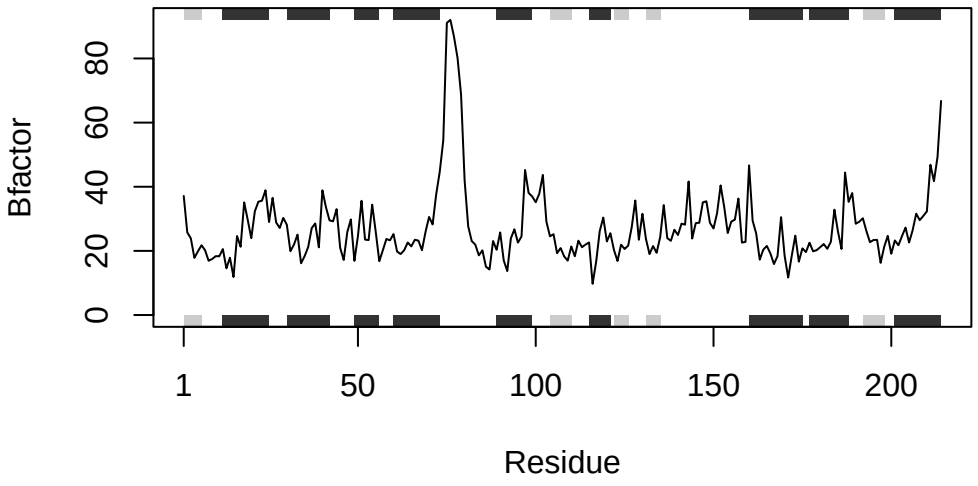
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

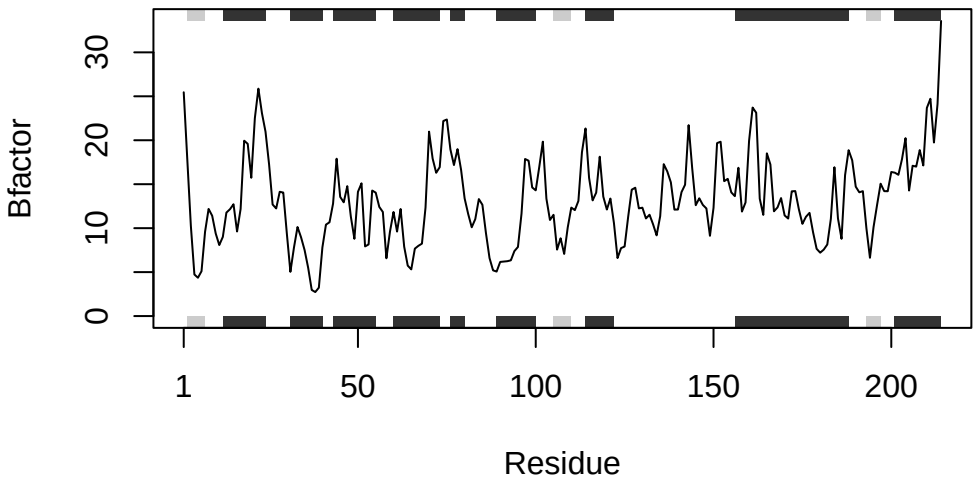
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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



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



Q1. What type of object is returned from the read.pdb() function?

```
typeof(s1)
```

```
[1] "list"
```

`read.pdb()` returns a list object type of class pdb.

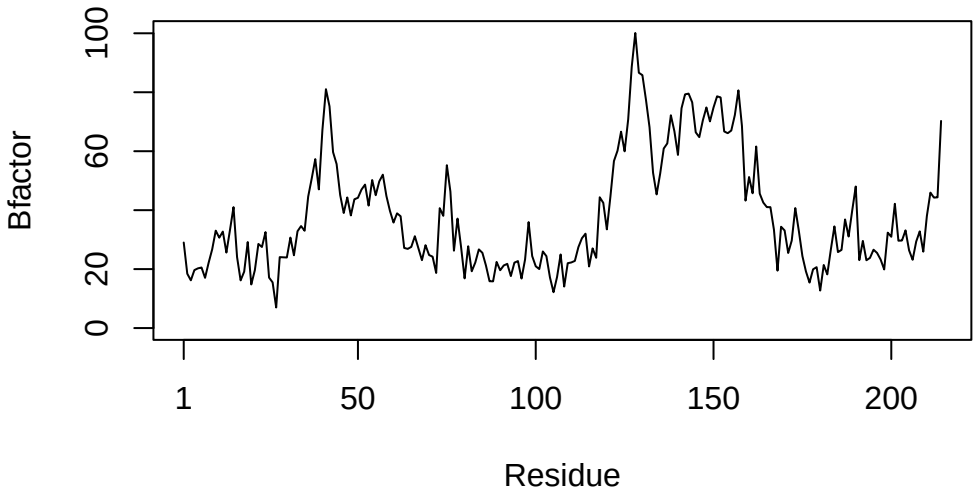
Q2. What does the trim.pdb() function do?

The `trim.pdb()` function creates smaller pdb object containing a subset atoms from the original object.

Q3. What input parameter would turn off the marginal black and grey rectangles in the plots and what do they represent in this case?

The input parameters `top=FALSE, bot=FALSE` in the `plotb3()` function will remove the rectangles.

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



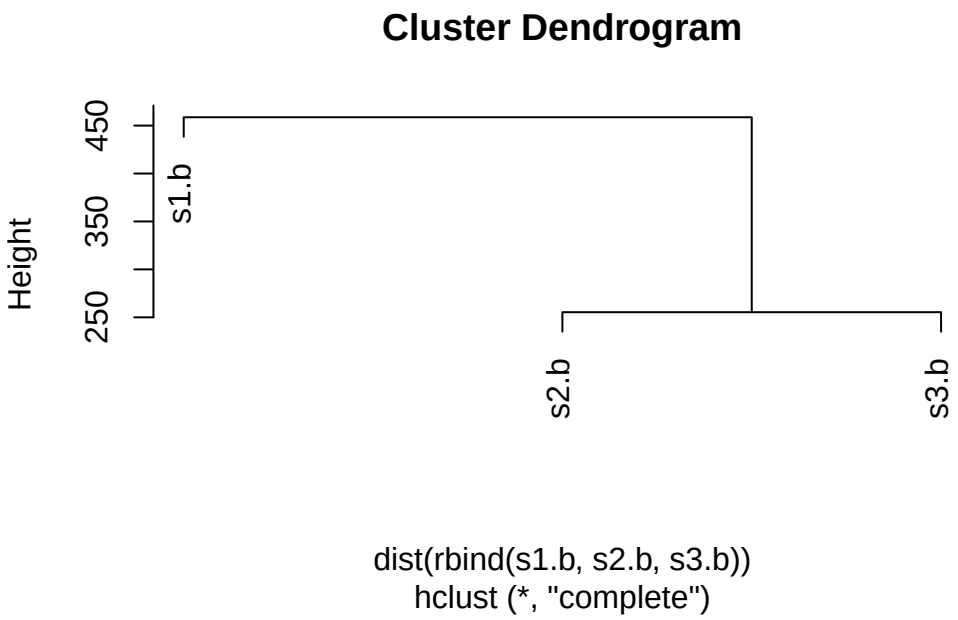
Q4. What would be a better plot to compare across the different proteins?

A better plot to compare would be the use of Dendrograms which would visualize cluster analysis across the different proteins.

Q5. Which proteins are more similar to each other in their B-factor trends. How could you quantify this?

Proteins *1AKE* and *1E4Y* were found to be more similar in B-factor trends. To quantify this, I had created a Dendrogram using the `hclust`,`dist`, and `rbind` functions and ploting their results.

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



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

must load bio3d package using `install.packages("bio3d")` then call it using the `library()` function.

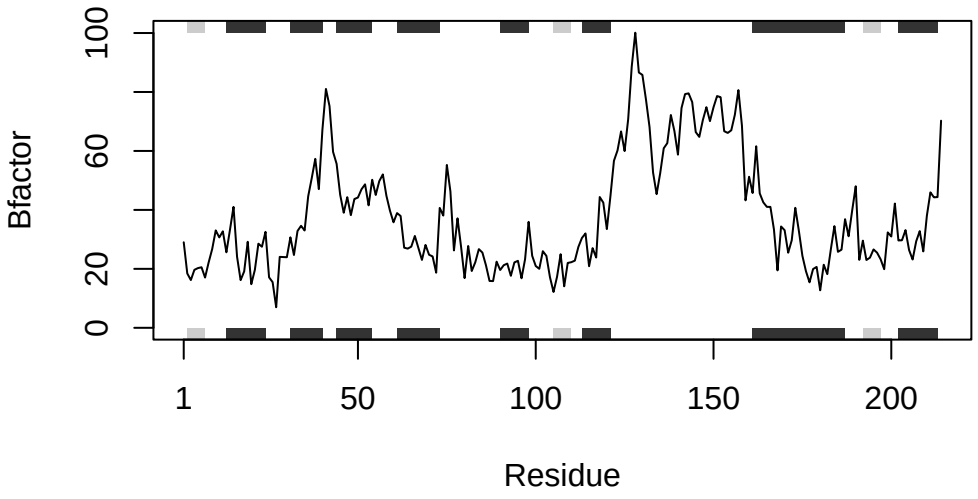
```
library(bio3d)
  #Input to function is the character name of the protein of interest
ProteinPlot = function(protein){
  #will read and obtain protein from pdb database
  x = read.pdb(protein)
  #obtains and stores subset of atoms in Alpha chain from protein
  x.chainA = trim.pdb(x, chain="A", elety="CA")
  #obtains and stores b atoms from protein
  x.b = x.chainA$atom$b
  #plots protein line graphs with residues versus b-factor
  plotb3(x.b, sse=x.chainA, typ="l", ylab="Bfactor")
}
```

To use function, call it using `ProteinPlot()` and inputing the protein name as the input argument such as **"4AKE"**, **"1AKE"**, and **"1E4Y"**. `ProteinPlot()` will output a line plot of the protein with with residues versus b-factor

```
ProteinPlot("4AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\jimmi\AppData\Local\Temp\RtmpUPwhUm/4AKE.pdb exists. Skipping download
```

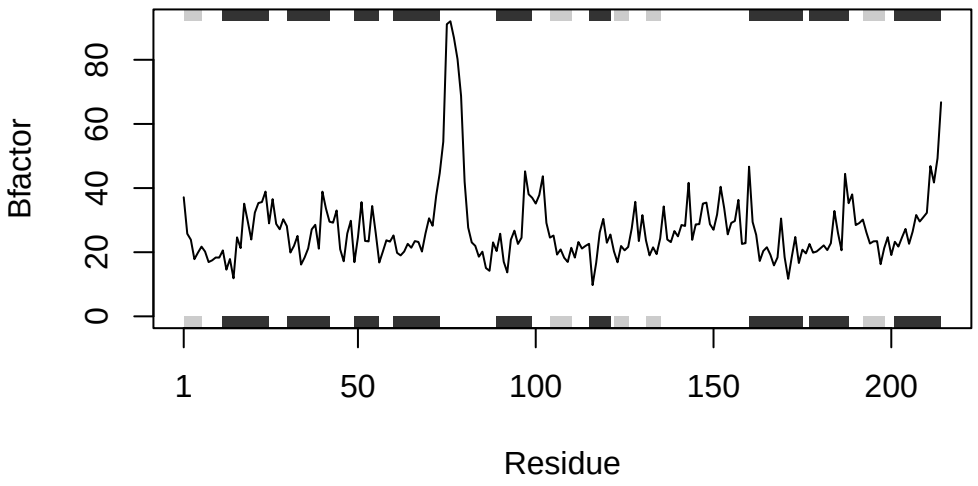



```
ProteinPlot("1AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\jimmi\AppData\Local\Temp\RtmpUPwhUm/1AKE.pdb exists. Skipping download
```

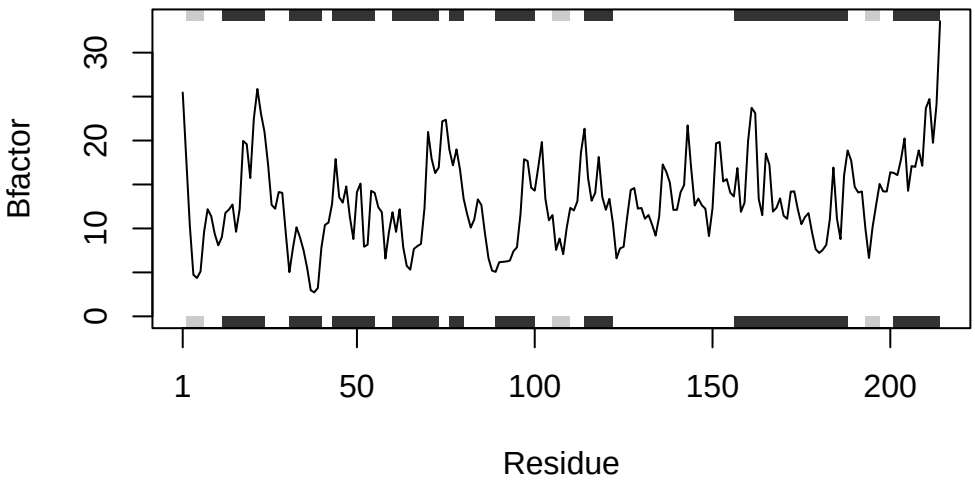
```
   PDB has ALT records, taking A only, rm.alt=TRUE
```




```
ProteinPlot("1E4Y")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\jimmi\AppData\Local\Temp\RtmpUPwhUm/1E4Y.pdb exists. Skipping download
```


Bfactor
0
10
20
30
1
50
100
150
200
Residue