Class 6: Write a Function Q6

Sophia Wang A16838155

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

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

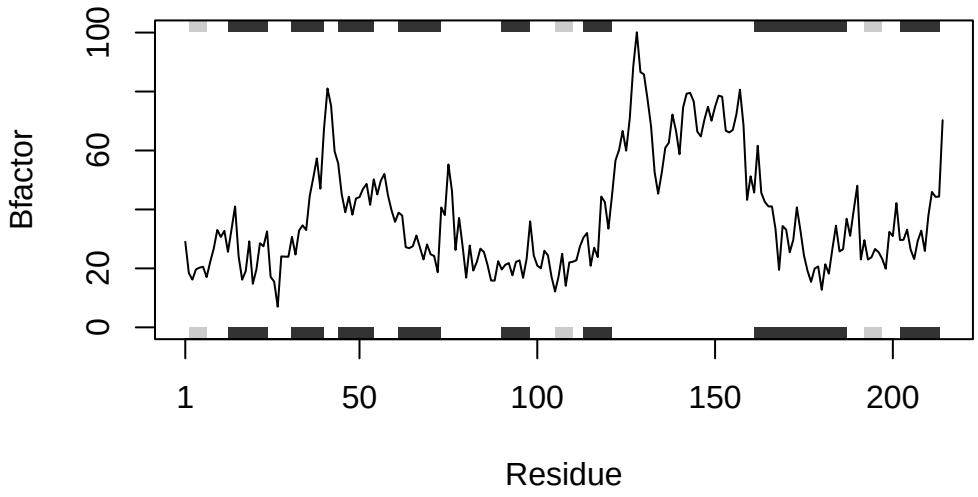
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

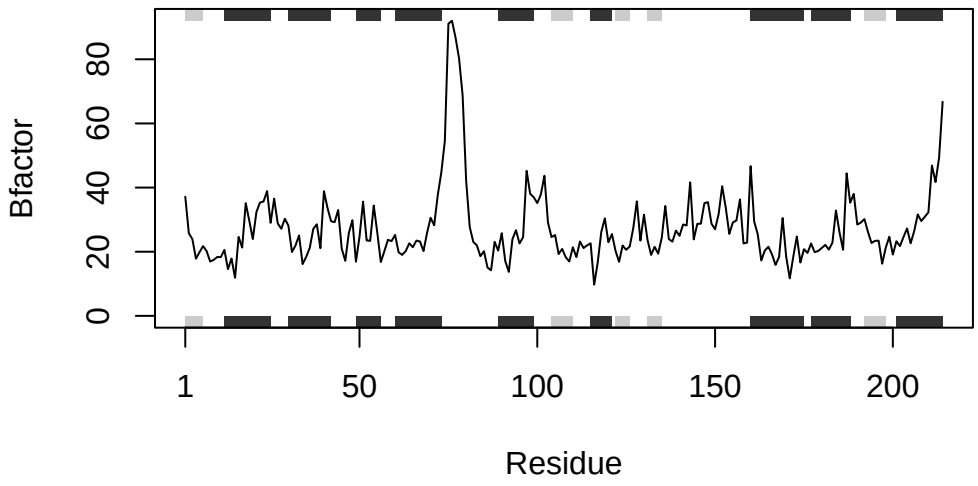
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

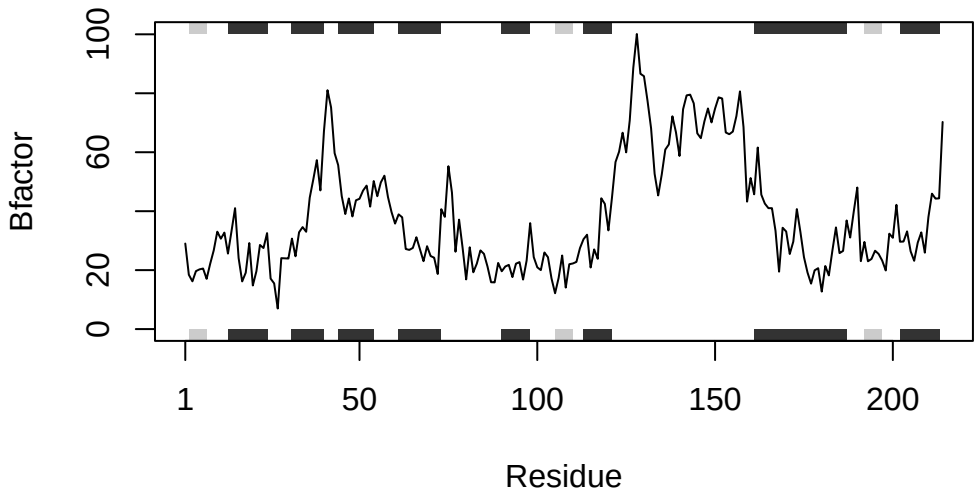
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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



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



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

```
class(s1)
```

```
[1] "pdb" "sse"
```

```
class(s2)
```

```
[1] "pdb" "sse"
```

```
class(s3)
```

```
[1] "pdb" "sse"
```

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

Produce a new smaller PDB object, containing a subset of atoms, from a given larger PDB object.

```
?trim.pdb()
```

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

FALSE sse input turns off the marginal black and grey rectangles in the plots. Secondary structure object.

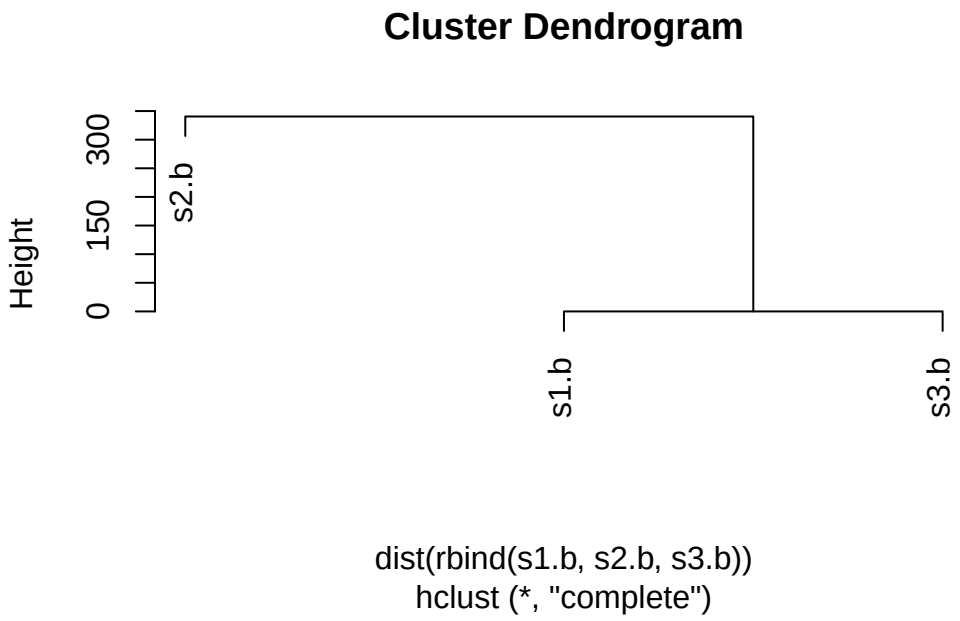
Q4. What would be a better plot to compare across the different proteins?

Heatmap.

Q5. Which proteins are more similar to each other in their B-factor trends. How could you quantify this? HINT: try the rbind(), dist() and hclust() functions together with a resulting dendrogram plot. Look up the documentation to see what each of these functions does.

s1.b (kinase with drug) and s3.b (kinase with drug) are more similar to each other in their B-factor trends. This can be quantified using dist() function which computes and returns the distance matrix between the B-factor profiles of the proteins.

```
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?

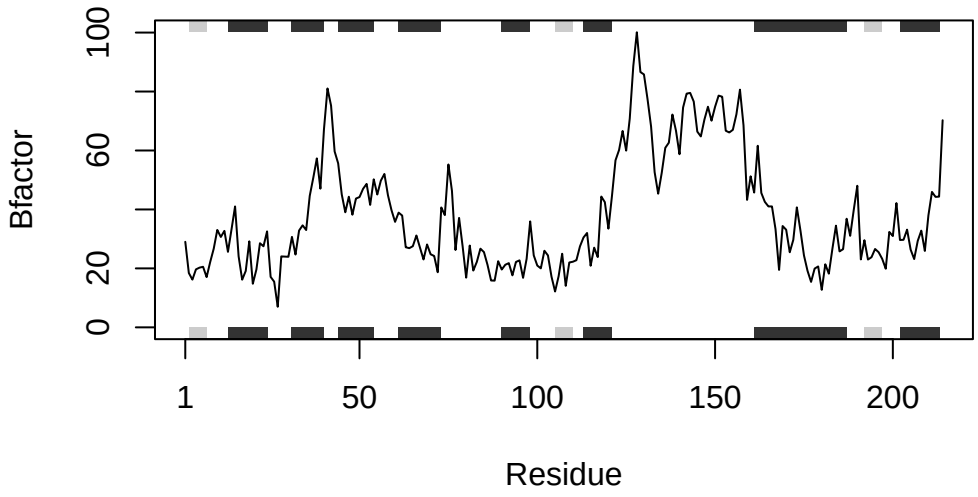
The function takes in three arguments: pdb id (character), the specific chain we are looking at (character), and the atom name (character); presents a plot of the protein's B factor and secondary structure elements. The function reads the PDB file, trims the structure, gets the b factor values, and plots the b factor values of the protein with the secondary structure elements on the margin.

The function outputs a plot of the b factor values of the protein with the secondary structure elements on the margin.

```
protein_bfactor<-function(protein_id,chain,atom){
  #input: pdb id (character), the specific chain we are looking at (character),
  #and the atom name (character)
  file<-read.pdb(protein_id)
  protein_chain <- trim.pdb(file,chain=chain,elety=atom)
  chain <- paste0("chain",chain)
  b_factor <- protein_chain$atom$b
  plotb3(b_factor, sse=protein_chain, typ="l", ylab="Bfactor")
}
protein_bfactor("4AKE","A","CA")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/n3/71d2q4t93flgq8f_vz36vq5c0000gp/T//RtmpzVjEiw/4AKE.pdb exists.
Skipping download
```


Bfactor
0
20
60
100
1
50
100
150
200
Residue