Question 6 Walkthrough

Ashley Martinez (PID: A17891957)

Let's first input the given code.

```
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

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

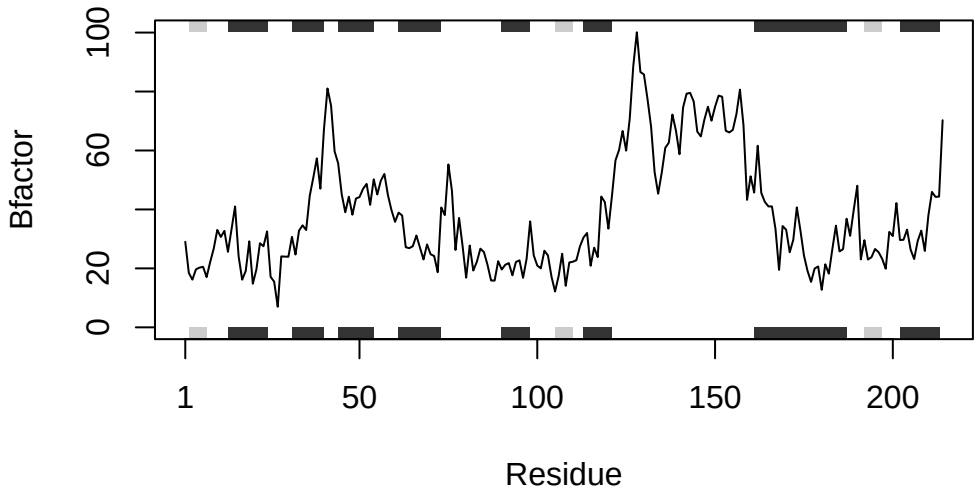
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

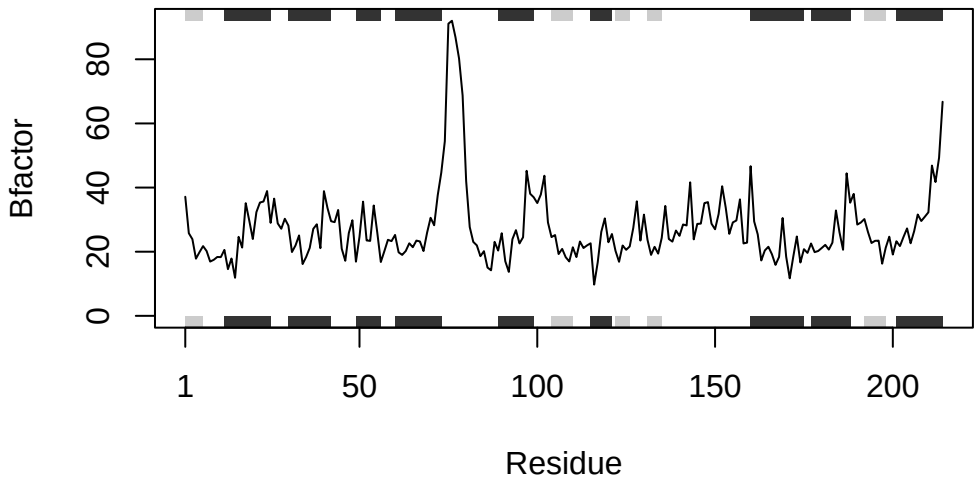
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

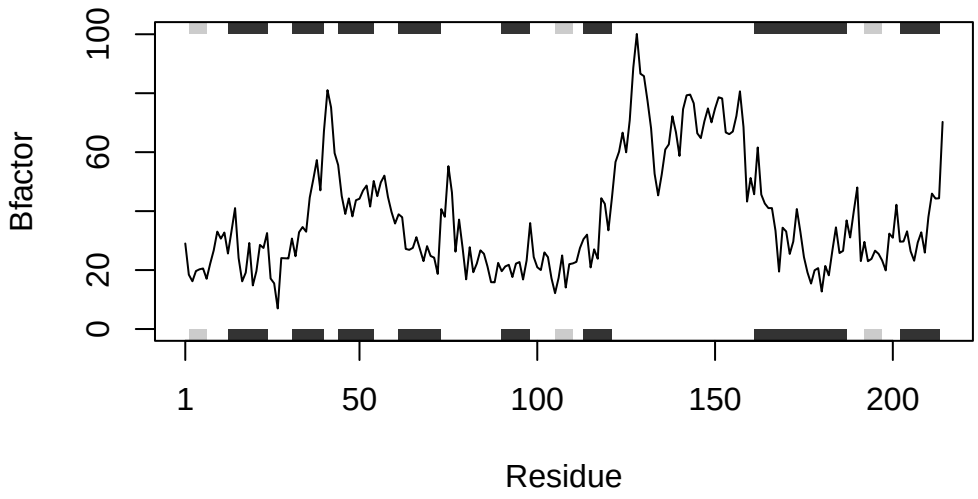
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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

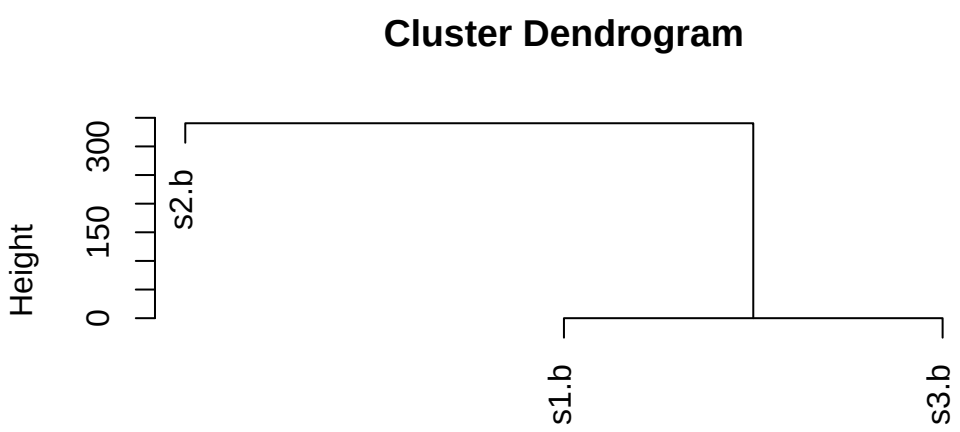


```
plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")
```



Now lets look at the line of code we will be using for the question.

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



The **inputs** appear to be the location of all the amino acid residues in the provided proteins. To **break down** the line of code, the 'hclust()' function groups the amino acid clusters based on how distant or similar they are. The 'dist()' portion calculates how similar or different the given clusters are, 'rbind()' combines all the information from the three protein structures into one matrix, and 'plot()' creates the physical plot for all this information. The **output** is a diagram that shows the distance and relationships between the three sequences.

Now let's generalize the code so any protein sequence can be inputted by the user by using a function.

```
protein_structures <- function(structures) {
  all_structures <- do.call(rbind, structures)
  hc <- hclust(dist(all_structures))
  plot(hc)
  return(hc)
}
```