HW Class06 Q6

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

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

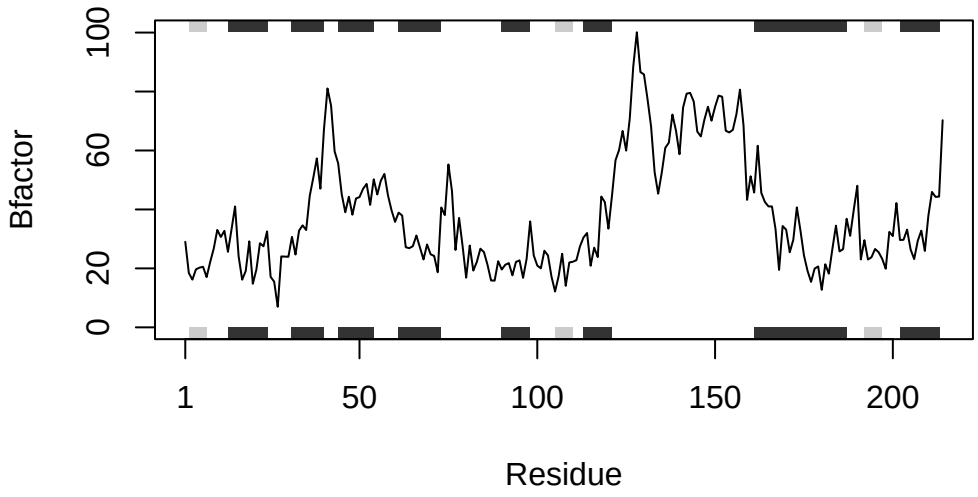
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

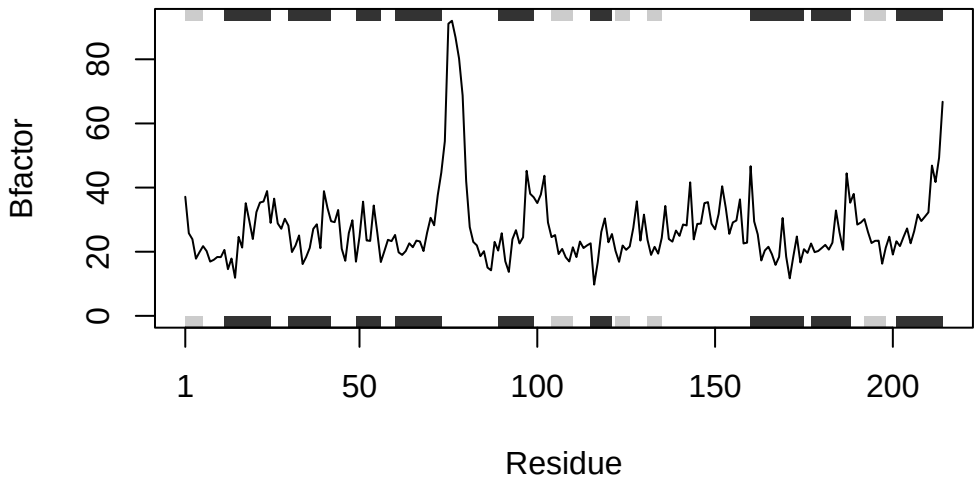
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

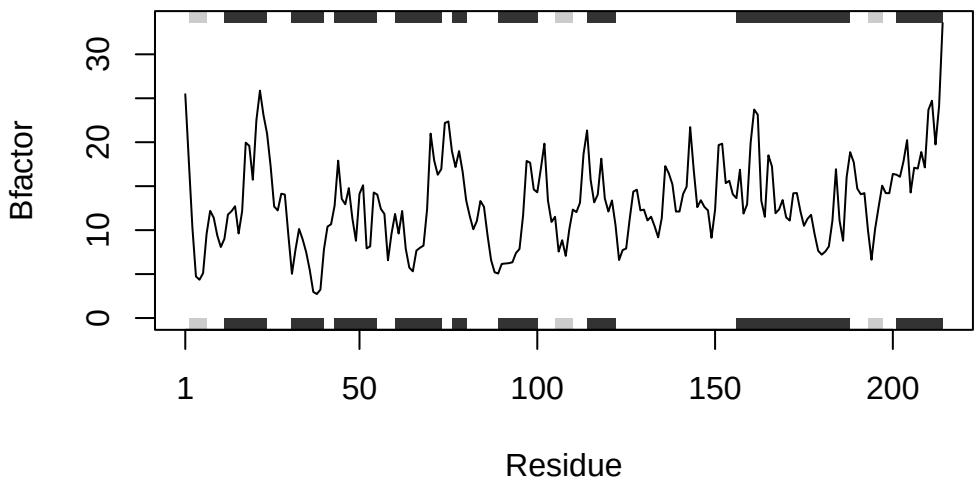
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

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?

It returns an object of class pdb, its a list containing different components that describe the structure of a protein or nucleic acids in PDB format.

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

It's used to filter or extract specific parts of the PDB strucutre.

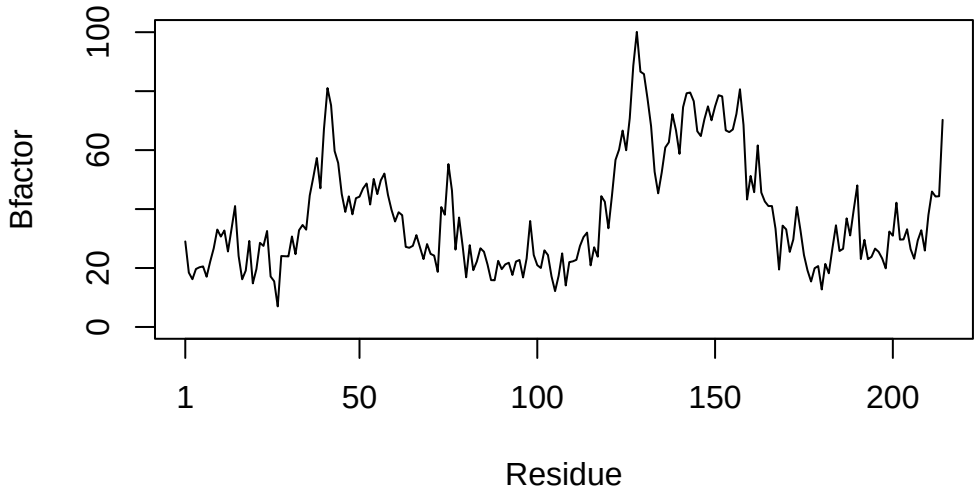
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
```

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

To turn off the marginal black and grey rectangles I would use sse=F

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

```
Warning in plotb3(s1.b, sse = FALSE, typ = "l", ylab = "Bfactor"): Length of
input 'sse' does not equal the length of input 'x'; Ignoring 'sse'
```

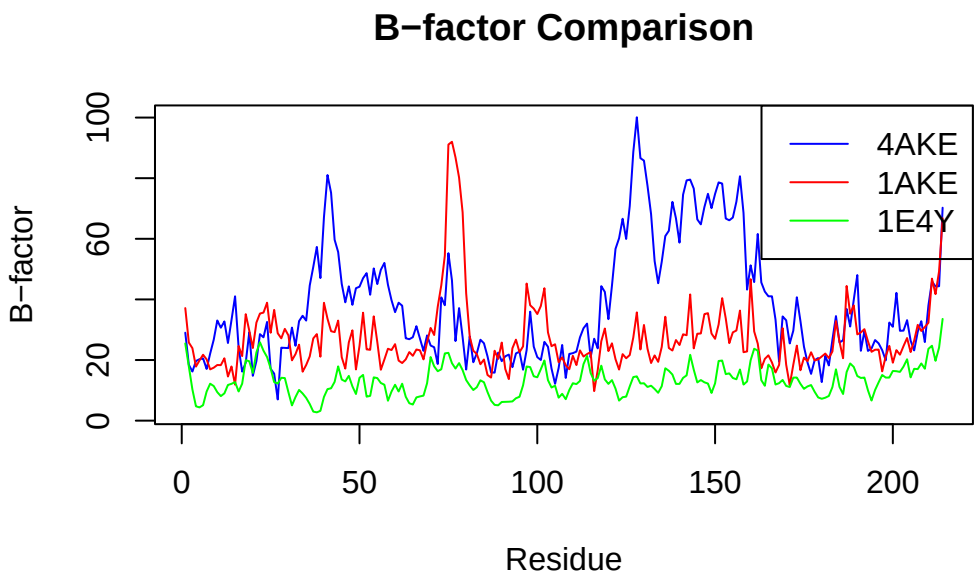


They represent the SSE of the protein. Black rectangles represent alpha helices and grey represents beta strands.

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

To compare across the different proteins I would overlap their plots in a single graph.

```
plot(s1.b, type="l", col="blue", ylab="B-factor", xlab="Residue", ylim=range(c(s1.b, s2.b, s
lines(s2.b, col="red")
lines(s3.b, col="green")
legend("topright", legend=c("4AKE", "1AKE", "1E4Y"), col=c("blue", "red", "green"), lty=1)
```



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.

1AKE and 1E4Y

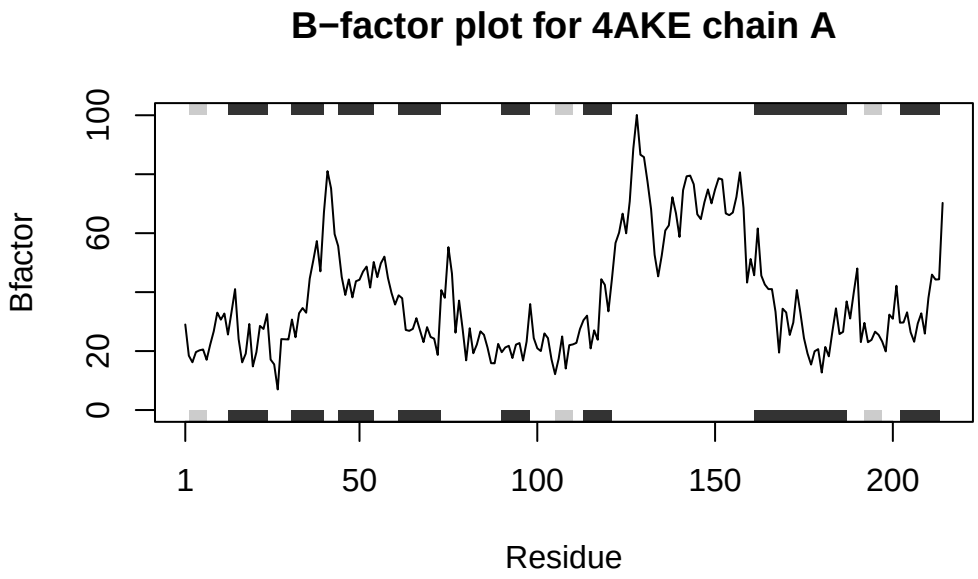
Q6. How would you generalize the original code above to work with any set of input protein structures?

```
library(bio3d)
#Need to define a function to analyze proteins
analyze_protein <- function(pdb_ID, chains)
{
#loop through each PDB ID and chain
  for(i in seq_along(pdb_ID)){
#read the PDB file for the current protein structure
    pdb <- read.pdb(pdb_ID[i])
#need to trim PDB to include only the specified chain and CA atoms
    chain_data <- trim.pdb(pdb, chain=chains[i], elety="CA")
#extract the B-factor values from the trimmed PDB data
    b_factors <- chain_data$atom$b
```

```
#plot the B-factors with SSE
    plotb3(b_factors, sse=chain_data, typ="l", ylab="Bfactor", main=paste("B-factor plot for
}
}

#list of PDB IDs
pdb_ID <- c("4AKE", "1AKE", "1E4Y")
#list corresponding chains for each PDB ID
chains <- c("A", "A", "A")
analyze_protein(pdb_ID, chains)
```

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

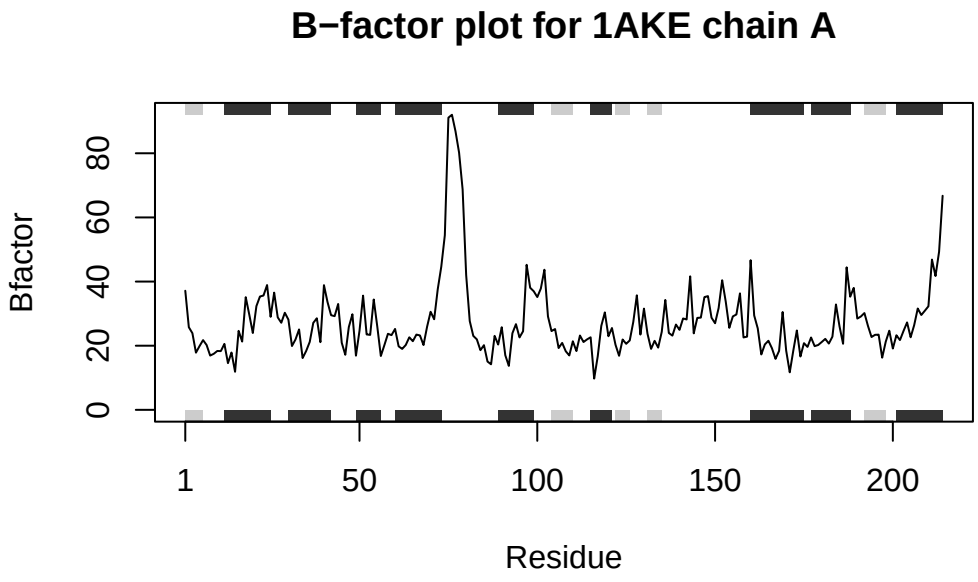
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\leonn\AppData\Local\Temp\Rtmpi2PvPH/4AKE.pdb exists. Skipping download
```



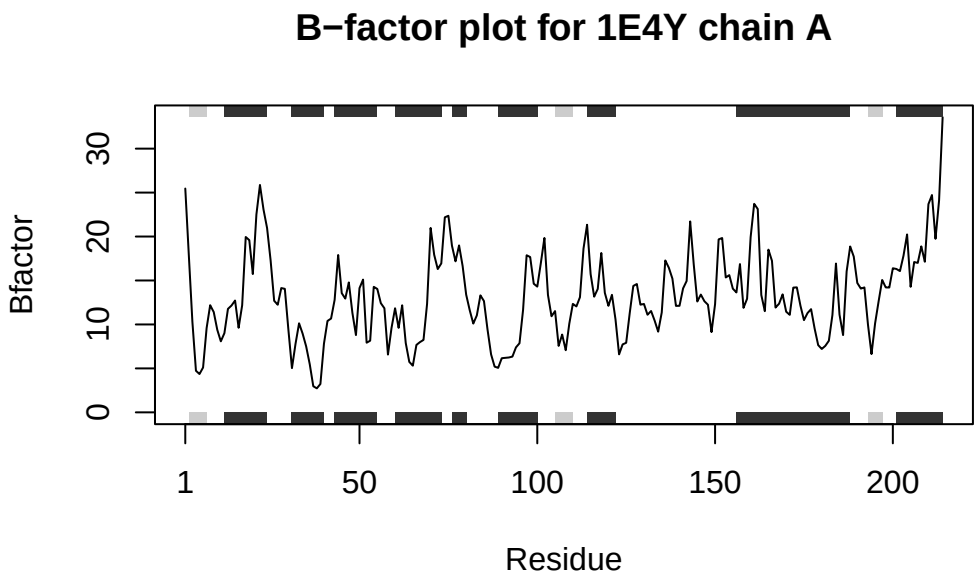

```
  Note: Accessing on-line PDB file
```

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\leonn\AppData\Local\Temp\Rtmpi2PvPH/1AKE.pdb exists. Skipping download
```

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




```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\leonn\AppData\Local\Temp\Rtmpi2PvPH/1E4Y.pdb exists. Skipping download
```


B−factor plot for 1E4Y chain A
Bfactor
0
10
20
30
1
50
100
150
200
Residue