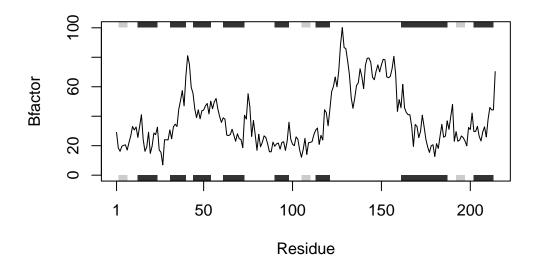
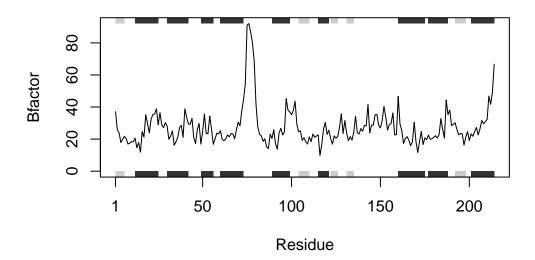
Class06_Hands-on section supplemental information

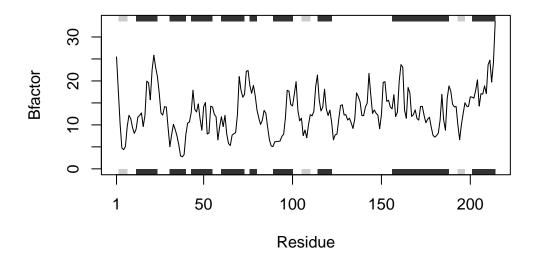
```
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug</pre>
Note: Accessing on-line PDB file
s2 <- read.pdb("1AKE") # kinase no drug
Note: Accessing on-line PDB file
PDB has ALT records, taking A only, rm.alt=TRUE
s3 <- read.pdb("1E4Y") # kinase with drug
Note: Accessing on-line PDB file
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")</pre>
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")</pre>
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")</pre>
s1.b <- s1.chainA$atom$b</pre>
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="1", ylab="Bfactor")
```



plotb3(s2.b, sse=s2.chainA, typ="1", ylab="Bfactor")



plotb3(s3.b, sse=s3.chainA, typ="1", 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"
```

According to "class()" function, the read.pdb() provides "pdb" "sse" object.

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

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

The outcome of the above code: Trim a PDB Object To A Subset of Atoms. Description Produce a new smaller PDB object, containing a subset of atoms, from a given larger PDB 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?

?plotb3()

The above code leads to explanation of each argument: top: logical, if TRUE rectangles for each sse are drawn towards the top of the plotting region. bot: logical, if TRUE rectangles for each sse are drawn towards the bottom of the plotting region.

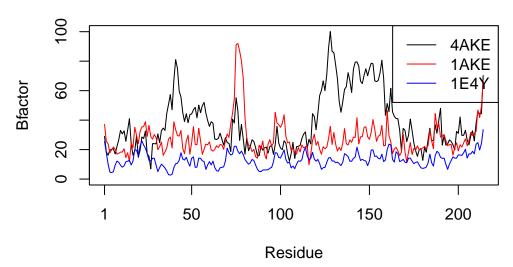
To turn them off, one can specify "FALSE" for each of "top" or "bot" argument.

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

```
# Plot the first dataset
plotb3(s1.b, sse = s1.chainA, typ = "l", ylab = "Bfactor", top=FALSE, bot=FALSE)

# Add lines for the second and third datasets
lines(s2.b, col = "red")
lines(s3.b, col = "blue")

# Add a legend to differentiate the datasets
legend("topright", legend = c("4AKE", "1AKE", "1E4Y"), col = c("black", "red", "blue"), lt
```

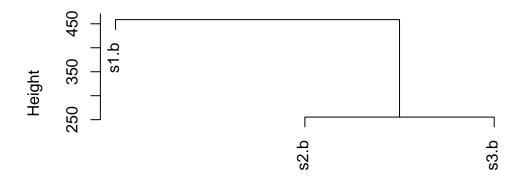


Using the "lines()" function, all three datasets can be plotted on to the same panel. This provides easier comparison between them.

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

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

Cluster Dendrogram



dist(rbind(s1.b, s2.b, s3.b)) hclust (*, "complete")

According to the above plot, s2 and s3 are more similar than s1. Quantification is indicated in the height parameter.

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

```
Protein_analysis <- function(pdb_code){
   s <- read.pdb(pdb_code)
   s.chainA <- trim.pdb(s, chain="A", elety="CA")
   s.b <- s.chainA$atom$b
   plotb3(s.b, sse=s.chainA, typ="l", ylab="Bfactor")
}
Protein_analysis("4AKE")</pre>
```

Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/vq/gq8n1k0n5051267pflhtlmmm0000gn/T//RtmpApPOmn/4AKE.pdb exists.
Skipping download

