HW 6

Yi-Hung Lee (PID: A16587141)

A. Can you improve this analysis code?

```r
df <- data.frame(a=1:10, b=seq(200,400,length=10),c=11:20,d=NA)
df$a <- (df$a - min(df$a)) / (max(df$a) - min(df$a))
df$b <- (df$b - min(df$a)) / (max(df$b) - min(df$b))
df$c <- (df$c - min(df$c)) / (max(df$c) - min(df$c))
df$d <- (df$d - min(df$d)) / (max(df$a) - min(df$d))

df$a
```

```
[1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
[8] 0.7777778 0.8888889 1.0000000
```

```r
df$b
```

```
[1] 1.000000 1.111111 1.222222 1.333333 1.444444 1.555556 1.666667 1.777778
[9] 1.888889 2.000000
```

```r
df$c
```

```
[1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
[8] 0.7777778 0.8888889 1.0000000
```

```r
df$d
```

```
[1] NA NA NA NA NA NA NA NA NA NA
```

Improved code:

```
analysis <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
```

Use the analysis function:

```
analysis(df$a)
```

```
 [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
 [8] 0.7777778 0.8888889 1.0000000
```

```
analysis(df$b)
```

```
 [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
 [8] 0.7777778 0.8888889 1.0000000
```

```
analysis(df$c)
```

```
 [1] 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
 [8] 0.7777778 0.8888889 1.0000000
```

```
analysis(df$d)
```

```
 [1] NA NA NA NA NA NA NA NA NA NA
```

B. Can you improve this analysis code?

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

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

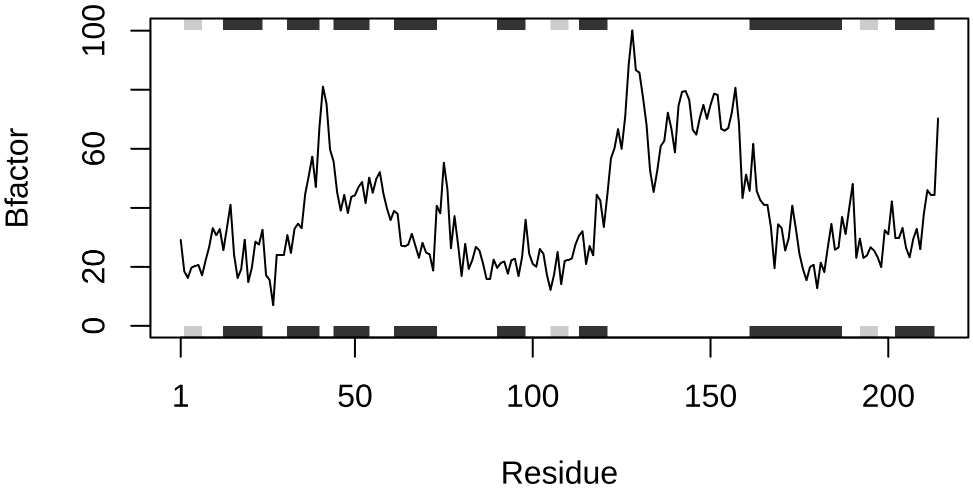
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

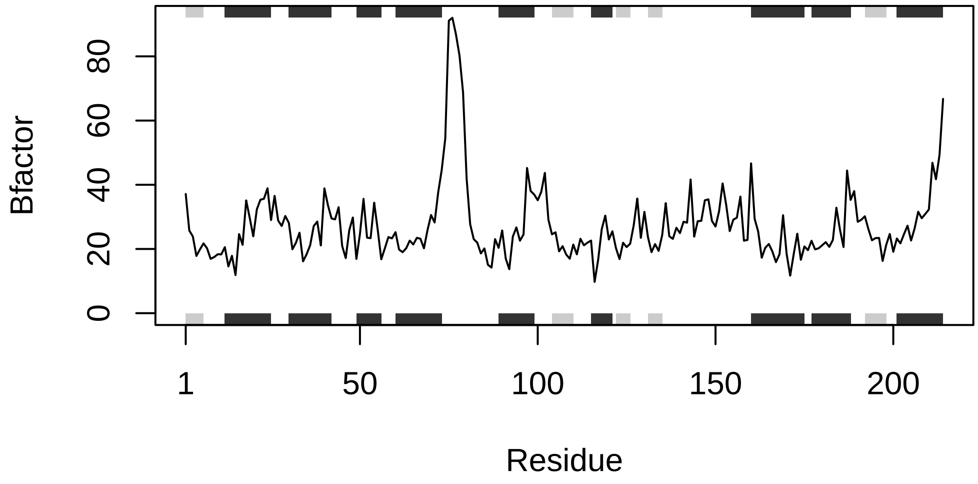
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

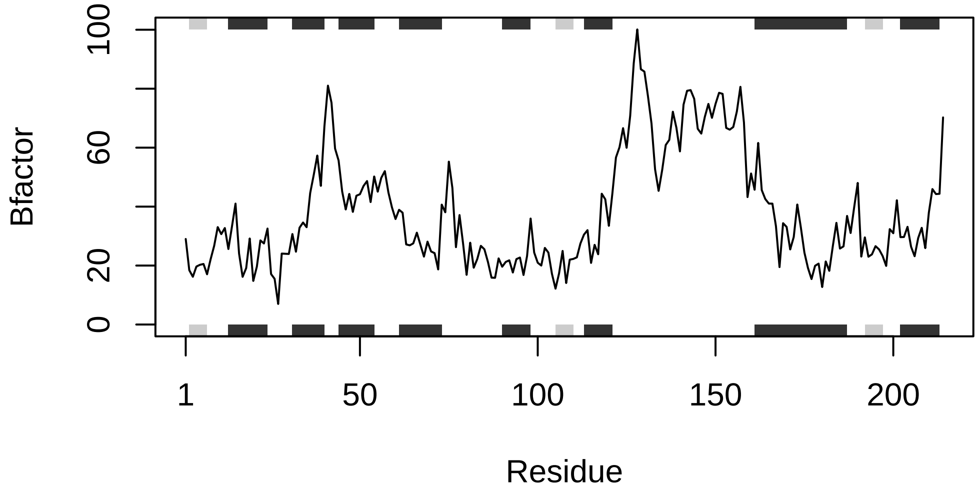
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA,typ="l", ylab="Bfactor")
```



```
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?

read.pdb() function returns a large pdb file that describes a protein structure including atoms, xyz, helix, amino acid sequence, etc.

> 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.

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

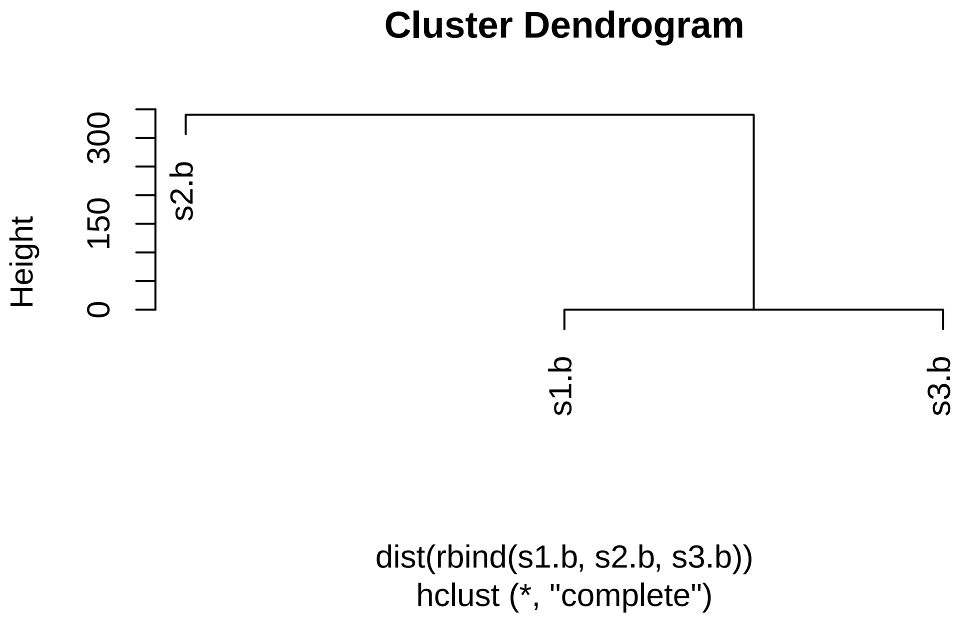
The marginal black and grey rectangles in the plots is controlled by sse parameter. sse parameter is the secondary structure object, in this case is the trim pdb.

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

A better plot to compare across different proteins could be a bar chart across all amino acid residues.

> 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.

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



It shows that s1(4AKE) and s3(1AKE) are colser to each other than s2(1E4Y).

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

Improved code:

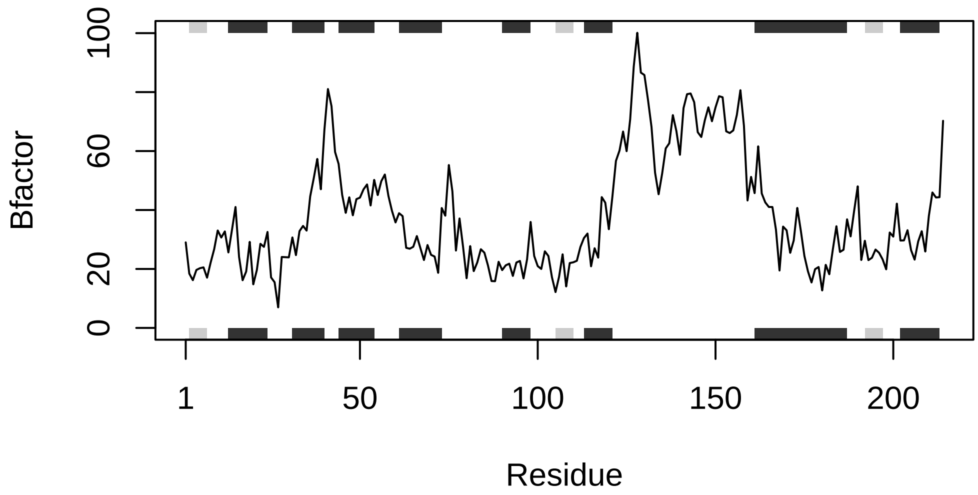
```r
library(bio3d)

plot_pdb <- function(x){
  s <- read.pdb(x)
  s.chainA <- trim.pdb(s, chain="A", elety="CA")
  s.b <- s.chainA$atom$b
  plotb3(s.b, sse=s1.chainA,typ="l", ylab="Bfactor")
}
```

Run improved code:

```r
plot_pdb('4AKE')
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/py/g13_n36s5c559410skx8hb7m0000gn/T//Rtmpq5Zo3M/4AKE.pdb exists.
Skipping download
```

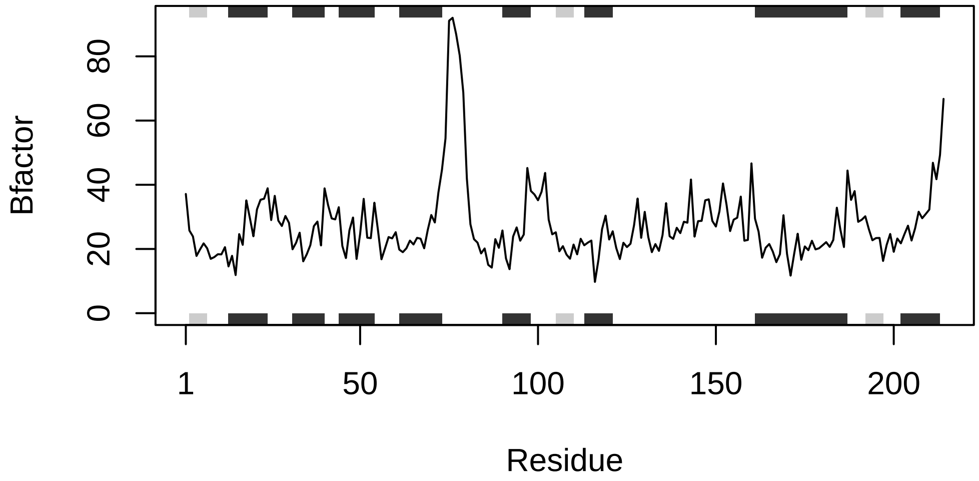


```
plot_pdb('1AKE')
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/py/g13_n36s5c559410skx8hb7m0000gn/T//Rtmpq5Zo3M/1AKE.pdb exists.
Skipping download
```

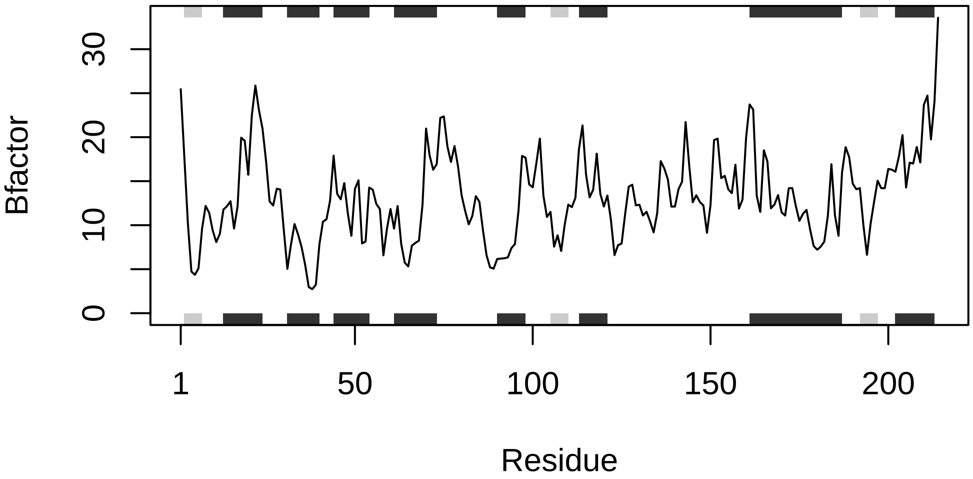
```
   PDB has ALT records, taking A only, rm.alt=TRUE
```



```
plot_pdb('1E4Y')
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/py/g13_n36s5c559410skx8hb7m0000gn/T//Rtmpq5Zo3M/1E4Y.pdb exists.
Skipping download
```


Bfactor
0
10
20
30
Residue
1
50
100
150
200