Class 6 Homework

Clarissa Savko (PID:A69028482)

2023-10-23

Original code

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

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

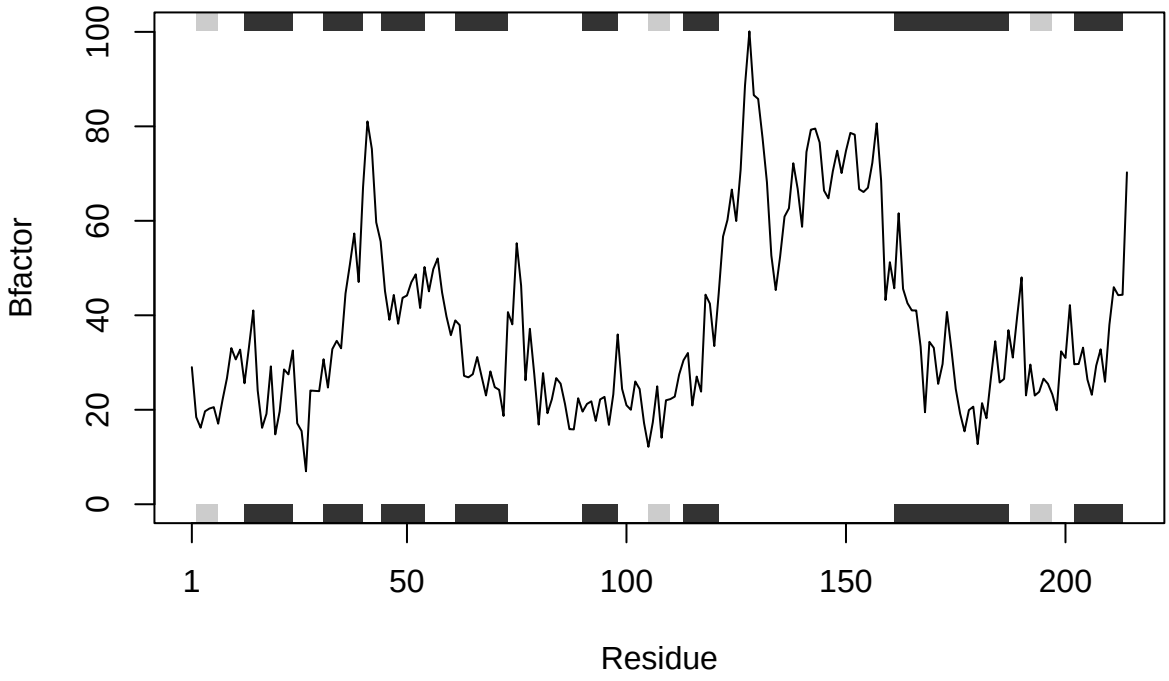
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

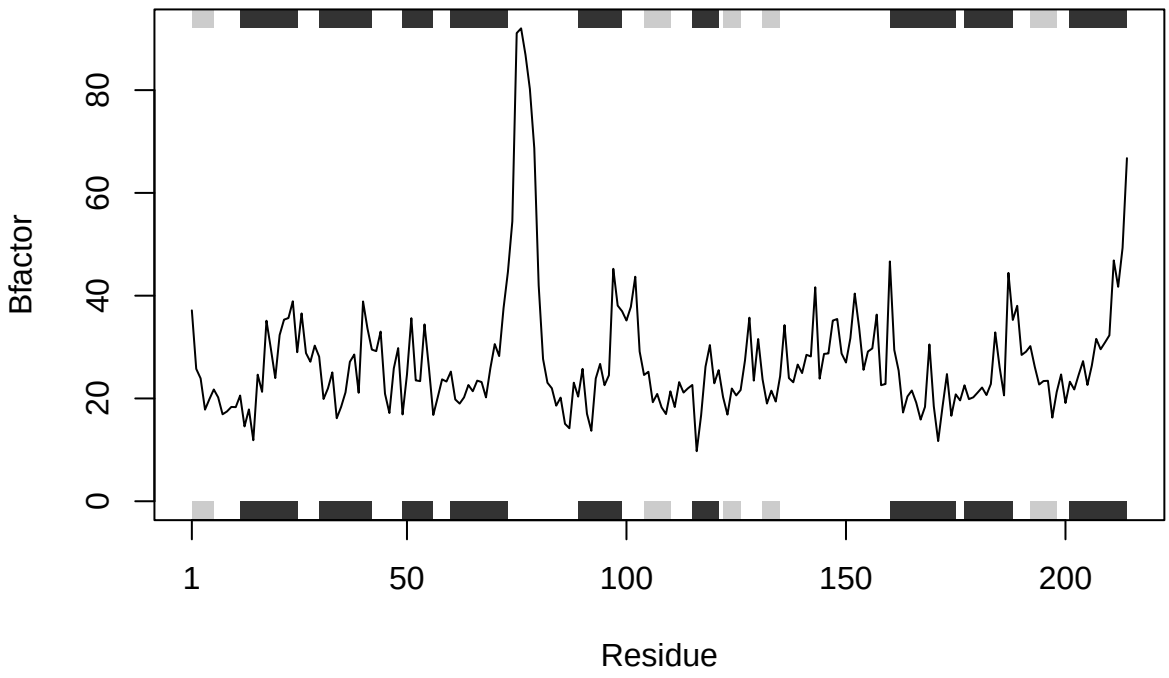
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

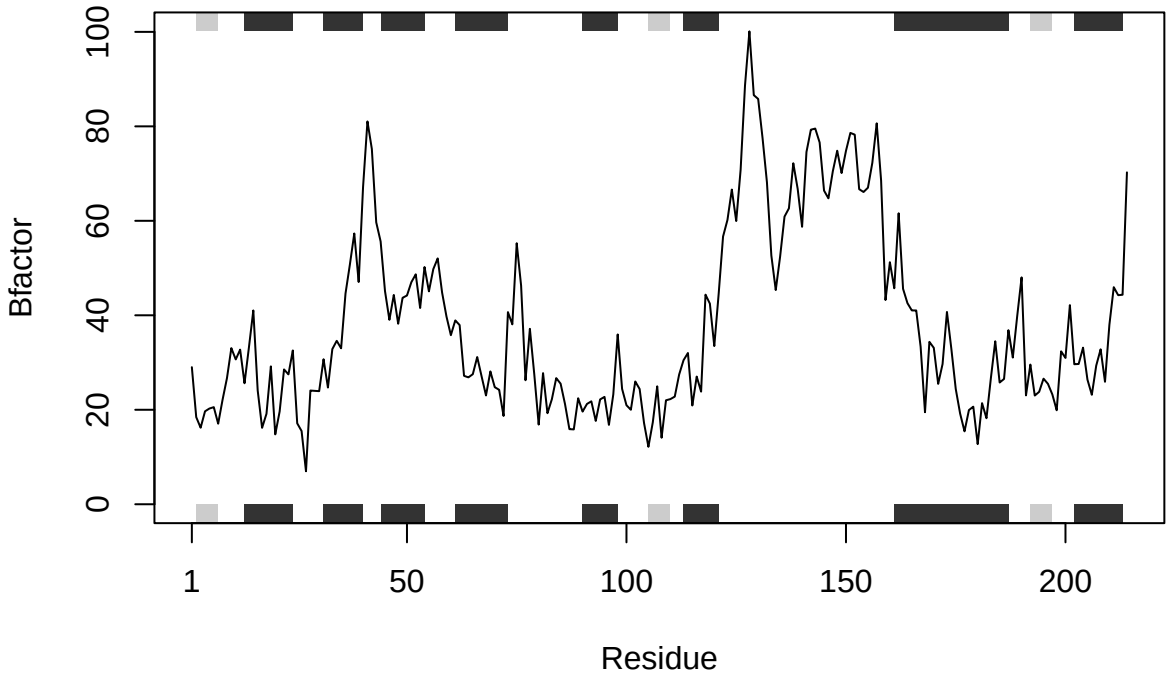
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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



`read.pdb()` returns a list with all of the protein data base parameters. `trim.pdb()` subsets a desired portion of the PDB object `top=FALSE` and `bot=false` would remove the marginal rectangles at the top and bottom of the plot. They represent the secondary structure elements of alpha helices and beta strands. Points rather than lines would make a better plot so you can see the individual data points.

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

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\clari\AppData\Local\Temp\Rtmpua1Qr0/4AKE.pdb exists. Skipping download
```

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

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\clari\AppData\Local\Temp\Rtmpua1Qr0/1AKE.pdb exists. Skipping download
```

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

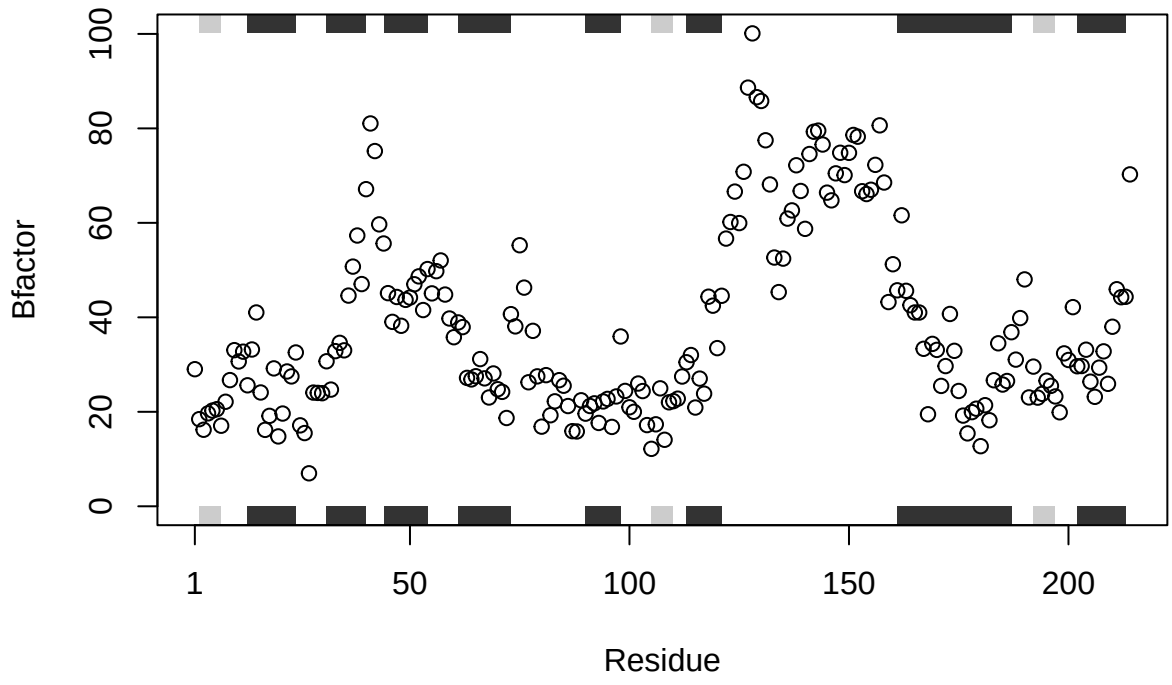
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

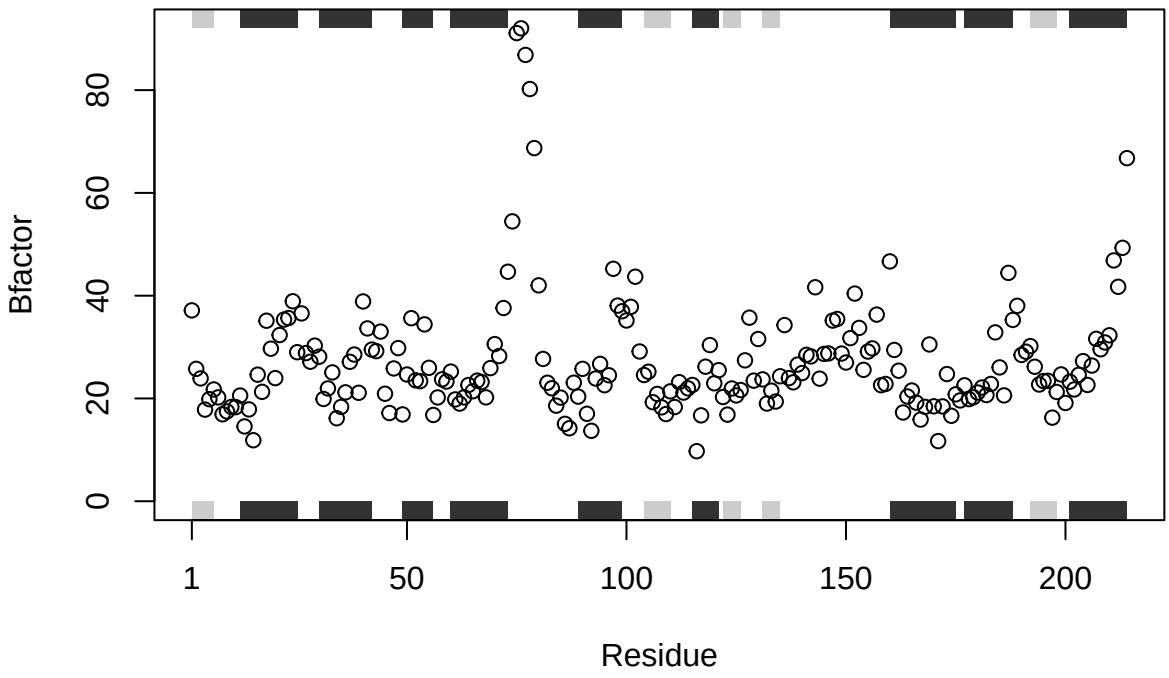
```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\clari\AppData\Local\Temp\Rtmpua1Qr0/1E4Y.pdb exists. Skipping download
```

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

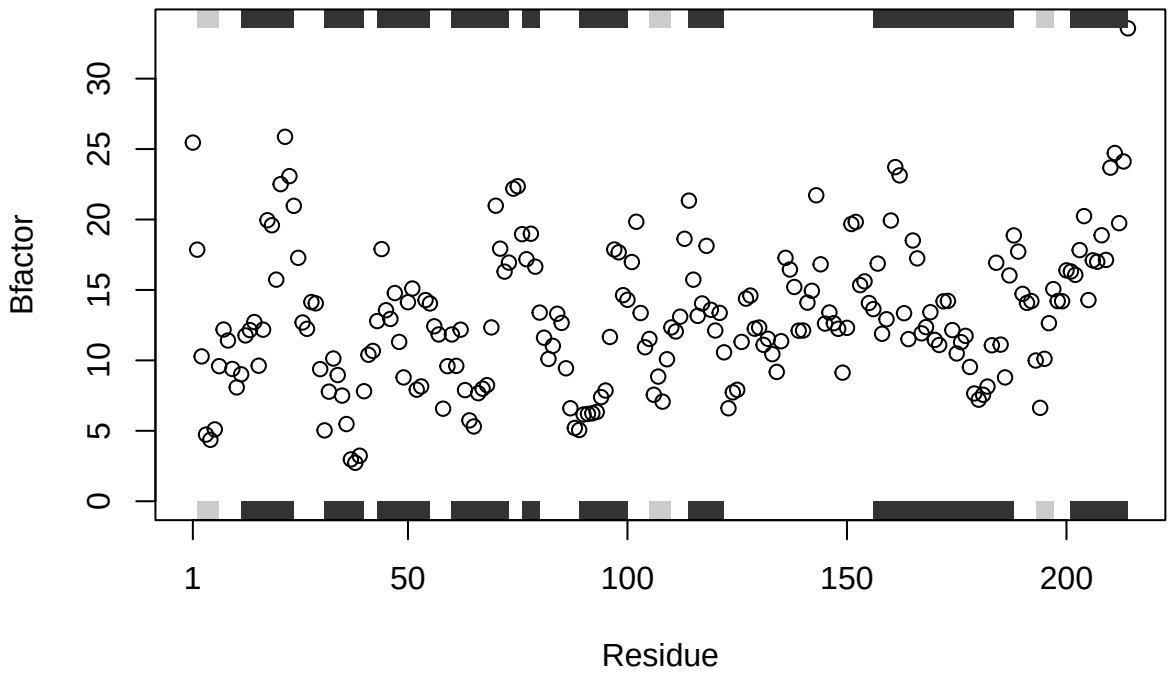
```
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="p", ylab="Bfactor")
```



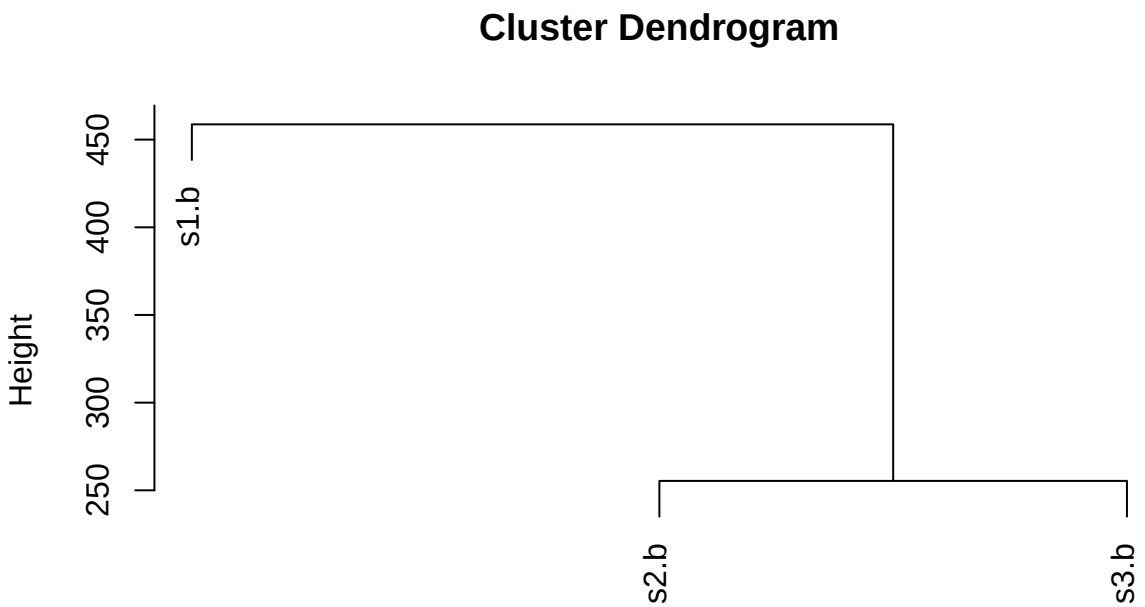

```
plotb3(s2.b, sse=s2.chainA, typ="p", ylab="Bfactor")
```




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




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



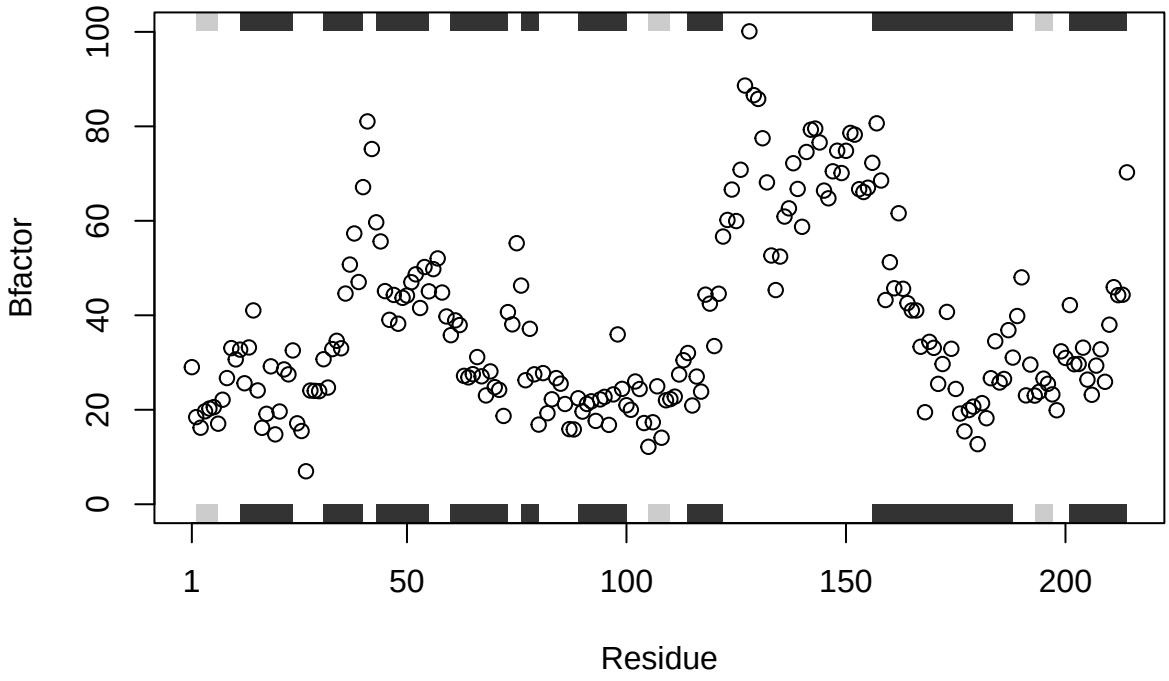
#The input of this function is the PDB accession code in quotation marks. The output is the plot.To run the code you simply call the function named `bfac()` with the input of the 4 character accession number of the protein you want to investigate ("XXXX").

```
bfac <- function(PDB) {
  s1 <- read.pdb(PDB)
  s1.chainA <- trim.pdb(s1, chain="A", elety= "CA")
  s1.b <- s1.chainA$atom$b
  plotb3(s1.b, sse=s3.chainA, typ="p", ylab="Bfactor")
}
```

```
bfac("4AKE")
```

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\clari\AppData\Local\Temp\Rtmpua1Qr0/4AKE.pdb exists. Skipping download
```


Bfactor
0
20
40
60
80
100
1
50
100
150
200
Residue