Class 6 Homework

Medhini Sosale

Original Code:

```
# Can you improve this analysis code?
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.3.3
```

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

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

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

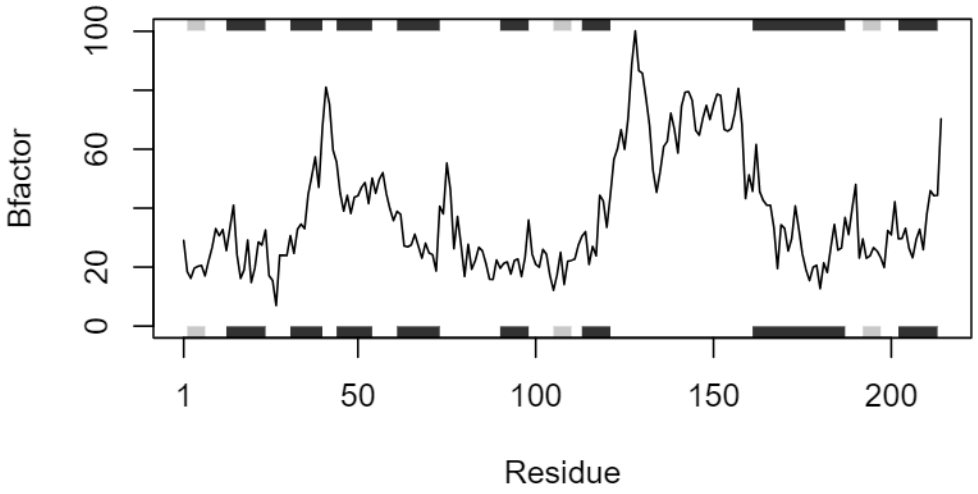
```
  Note: Accessing on-line PDB file
   PDB has ALT records, taking A only, rm.alt=TRUE
```

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

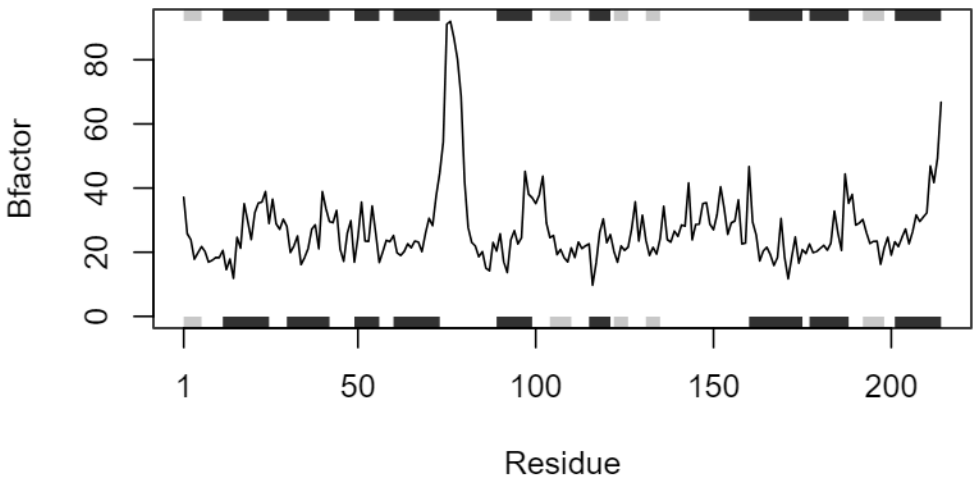
```
  Note: Accessing on-line PDB file
```

```
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
```

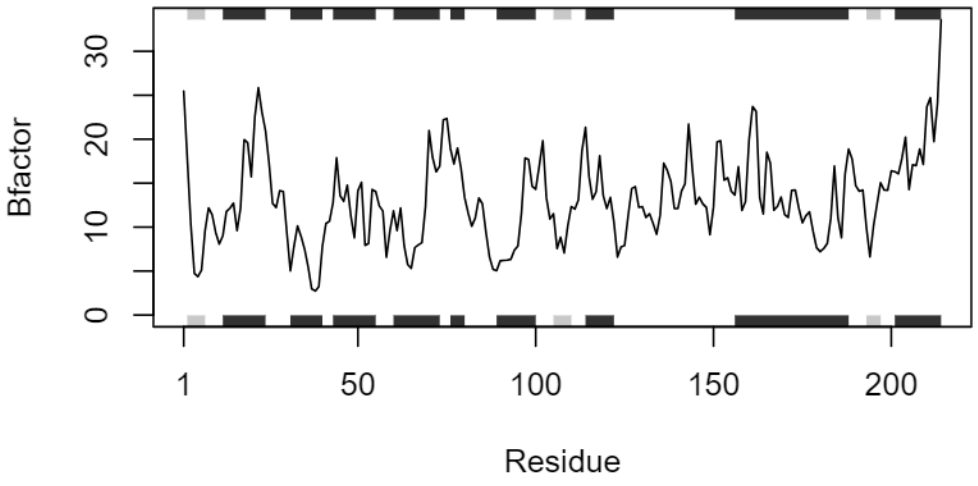
```
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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



Currently, the code reads online pdb data, trims it, pulls out the important information, and then plots it.

Improved Code:

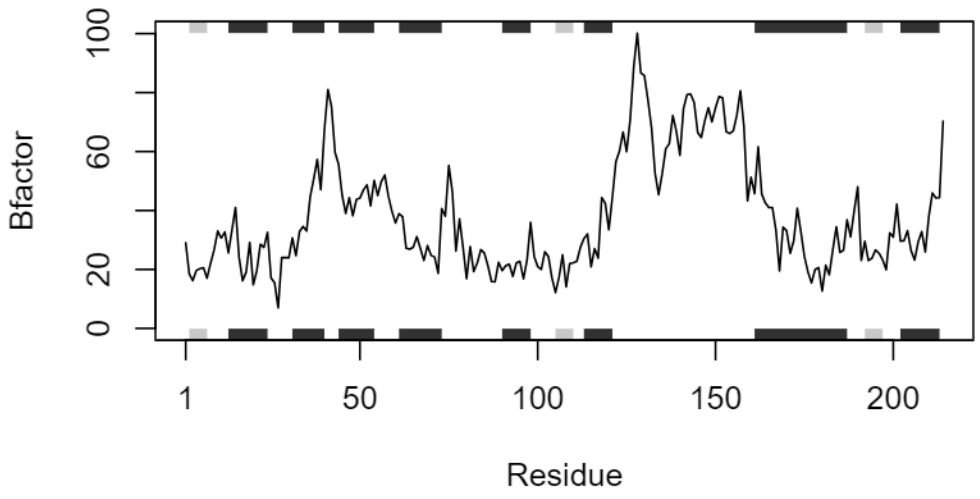
```
library(bio3d)

pdb_analysis <- function(accession){
  s <- read.pdb(accession) # kinase with drug
  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")
}

sapply(c("4AKE","1AKE", "1E4Y"), pdb_analysis)
```

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

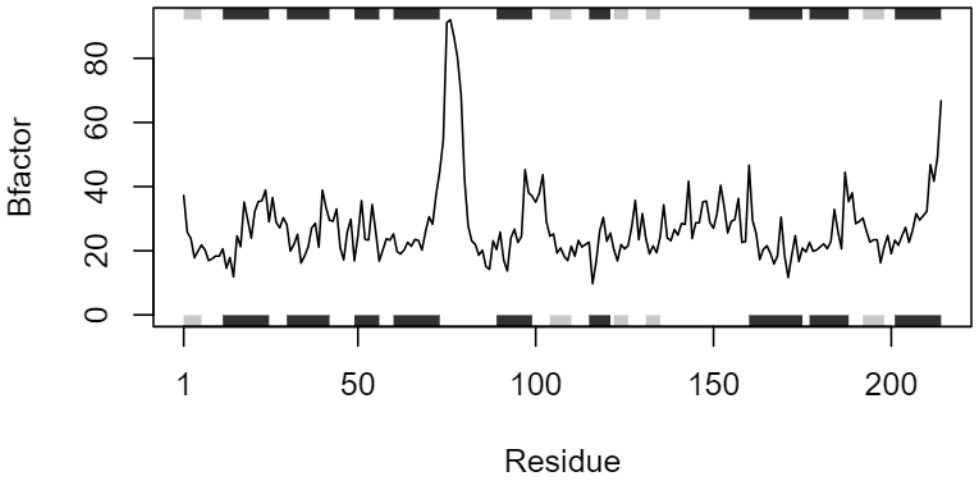
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\medhi\AppData\Local\Temp\RtmpqOYKev/4AKE.pdb exists. Skipping download
```




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

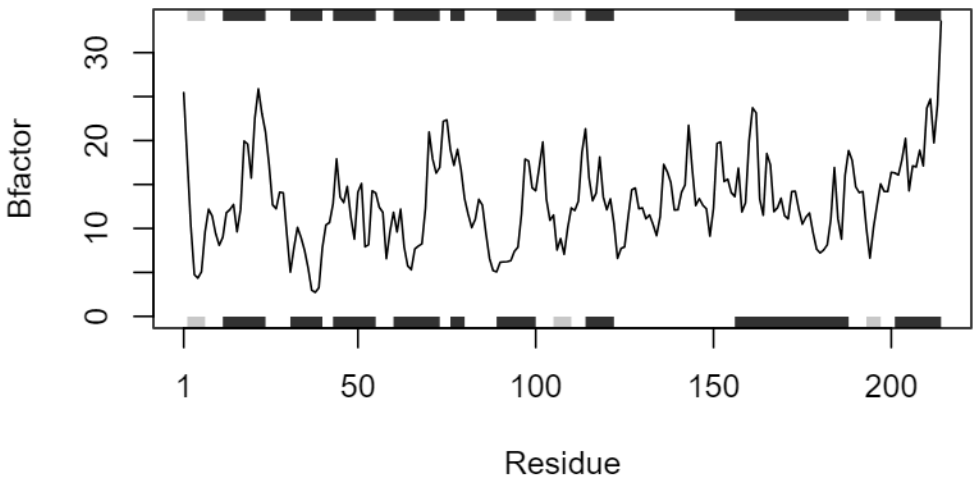
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\medhi\AppData\Local\Temp\Rtmpq0YKev/1AKE.pdb exists. Skipping download
```

```
   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\medhi\AppData\Local\Temp\Rtmpq0YKev/1E4Y.pdb exists. Skipping download
```




```
$`4AKE`
NULL

$`1AKE`
NULL

$`1E4Y`
NULL
```

The only input to the function is the PDB accession number. The read.pdb line takes that accession number and pulls the sequence from online. The s.chainA line trims the information to just the portion we want, the s.b line collects numbers, and then the final line of code in the function plots Bfactor vs. residue (this is the output of the function).