HW6

Bernice Lozada (A16297973)

Original code:

```r
# Can you improve this analysis code?
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

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

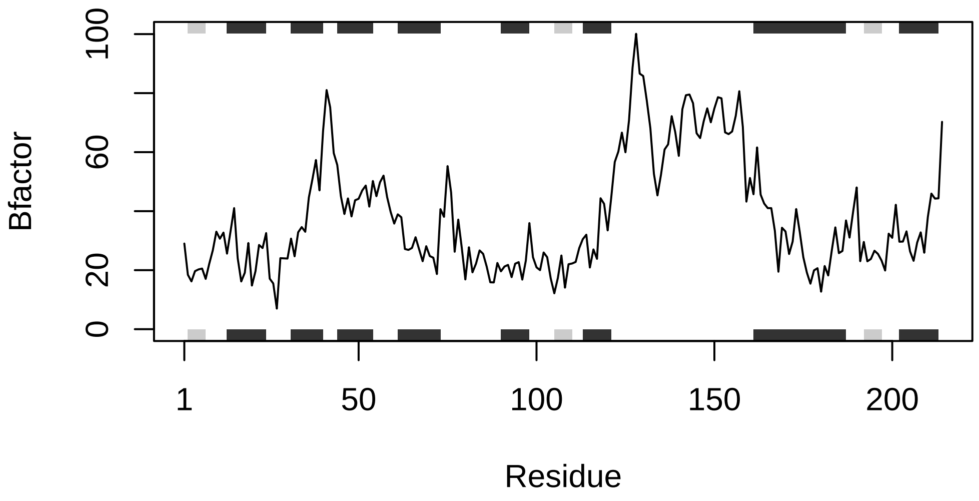
```r
s2 <- read.pdb("1AKE") # kinase no drug
```

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

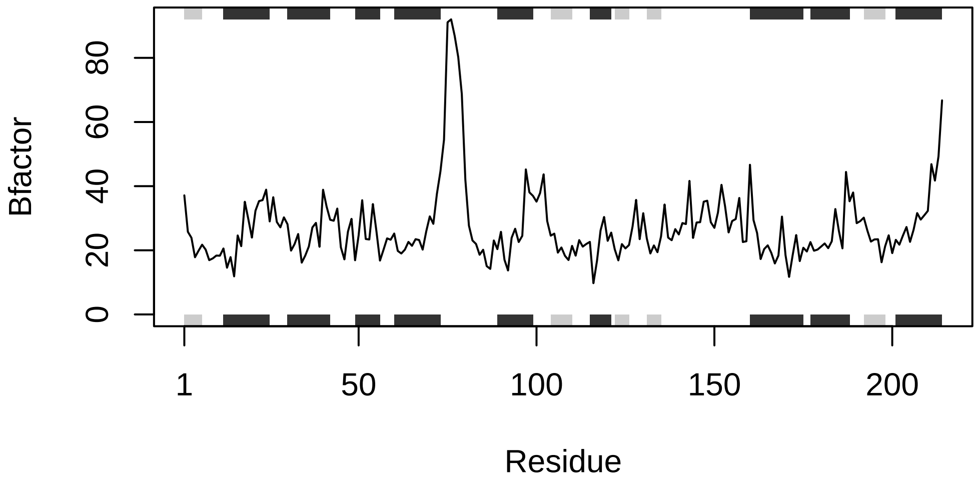
```r
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

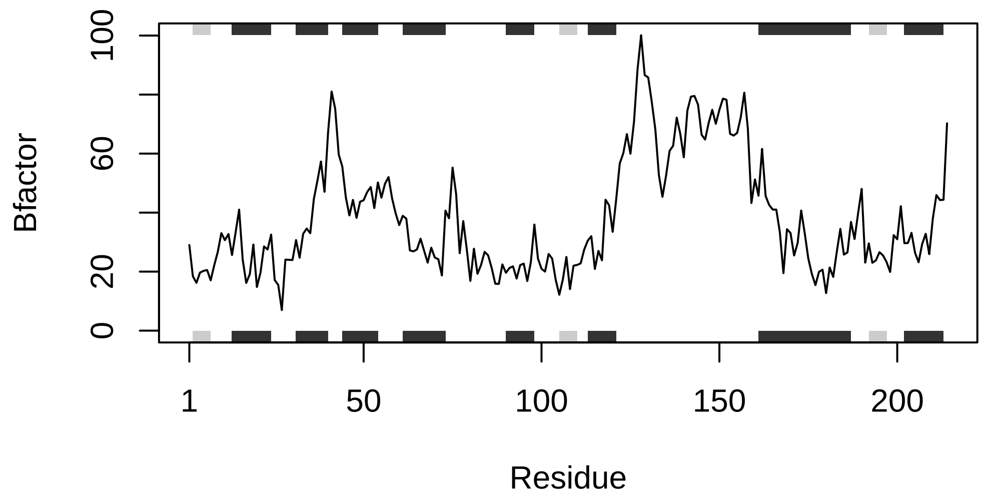
```r
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



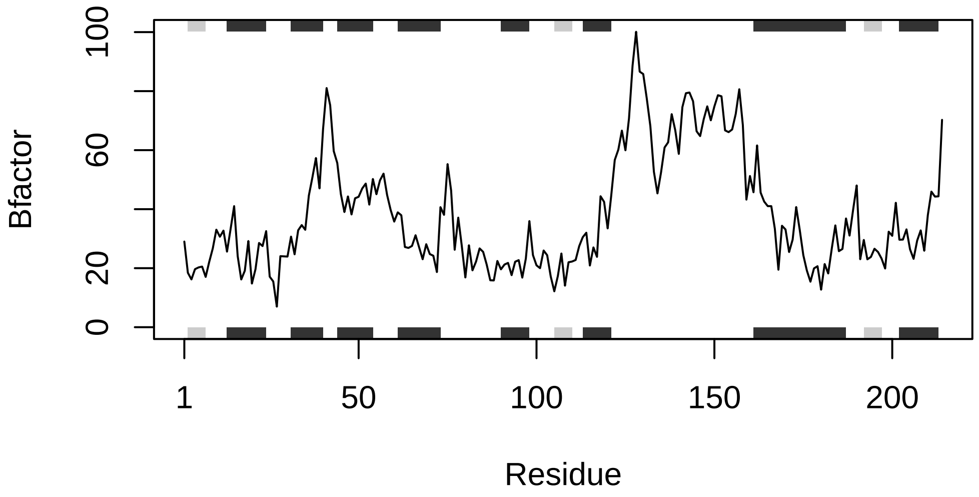
Next, we will fix the analysis code by streamlining it. This function has an input of a PDB name and will return a line graph plotting the B factor of each residue.

```r
plotbfactor <- function(kinase_name) {
  # load library into R
  library(bio3d)

  # find entry in pdb and make plot
  s1 <- read.pdb(as.character(kinase_name))
  s1.b <- s1.chainA$atom$b
  plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
}

# Example input - put the name of the gene of interest in quotes as the function input.
plotbfactor("1AKE") # s2 in original code
```

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
```

```
/var/folders/h7/rxqlfdfx7c31_xh4rq84wh3w0000gn/T//RtmpLOr0IM/1AKE.pdb exists.
Skipping download
```

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


Bfactor
Residue