hw06

Saba Heydari Seradj (A17002175)

Can you improve this analysis code?

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

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

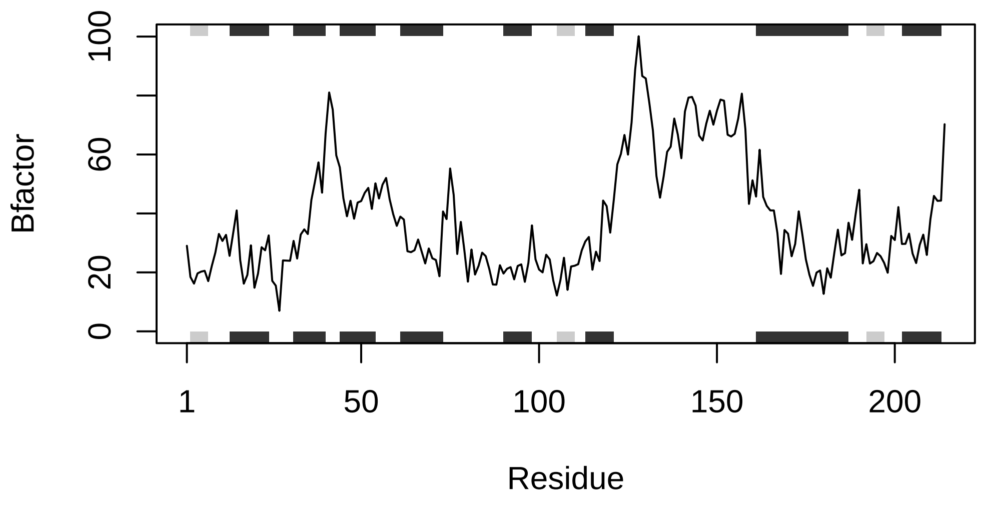
```r
s2 <- read.pdb("1AKE") # kinase no drug
```

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

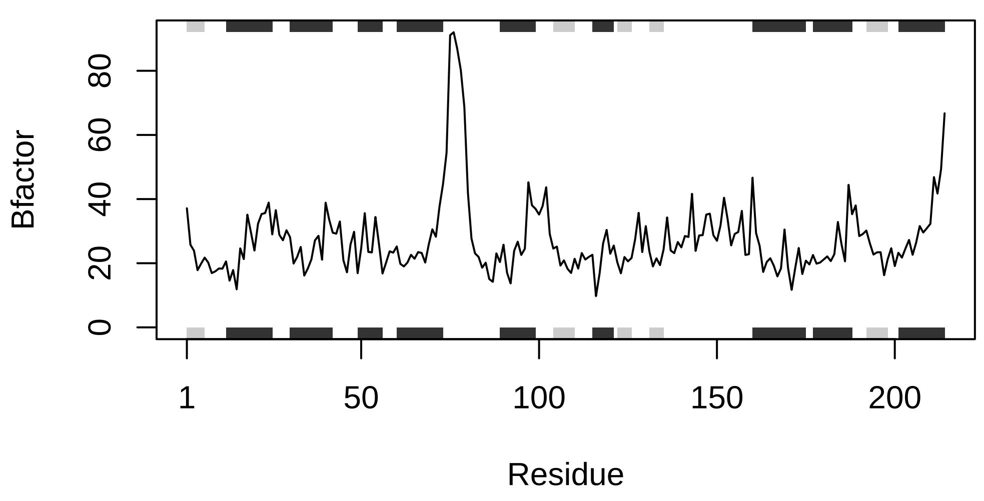
```r
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

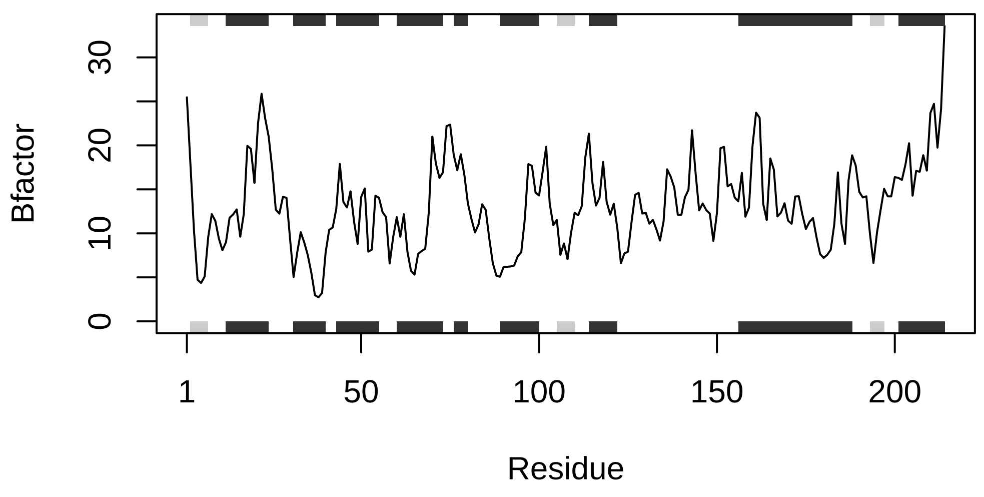
```r
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



These lines use the `read.pdb()` function from the `bio3d` package to read in three PDB structures: `4AKE`, `1AKE`, and `1E4Y`.

We could write this as a general statement: `sx <- read.pdb(pdb_id)`.

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

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/4AKE.pdb exists.
## Skipping download
```

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

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJ197E/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):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJ197E/1E4Y.pdb exists.
Skipping download
```

The `trim.pdb()` prodcues a new smaller PDB object, containing a subset of atoms from a larger PDB object (each structure). Here we are selecting only chain 'A' and only the alpha carbon atoms (CA).

General statement: `sx.chainA <- trim.pdb(sx, chain="A", elety="CA")`

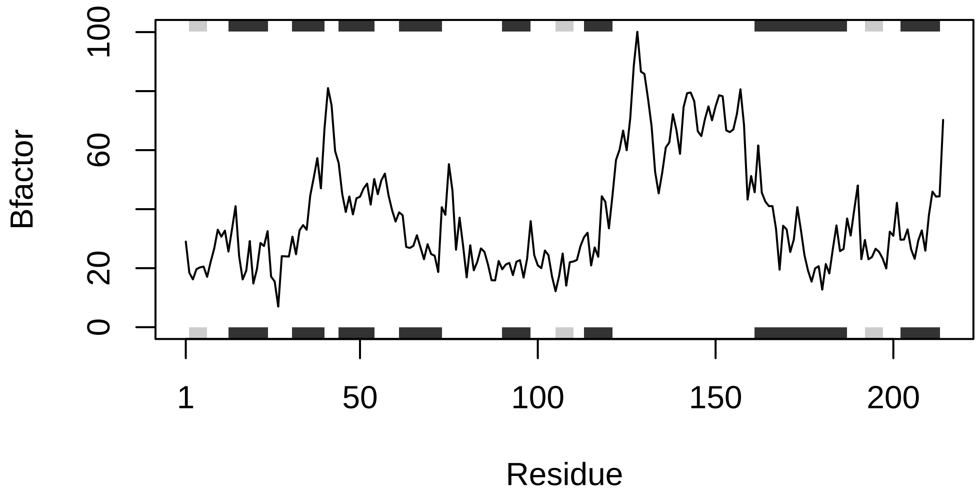
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
```

These lines extract the B-factor values for the alpha carbon atoms from the trimmed PDB objects.

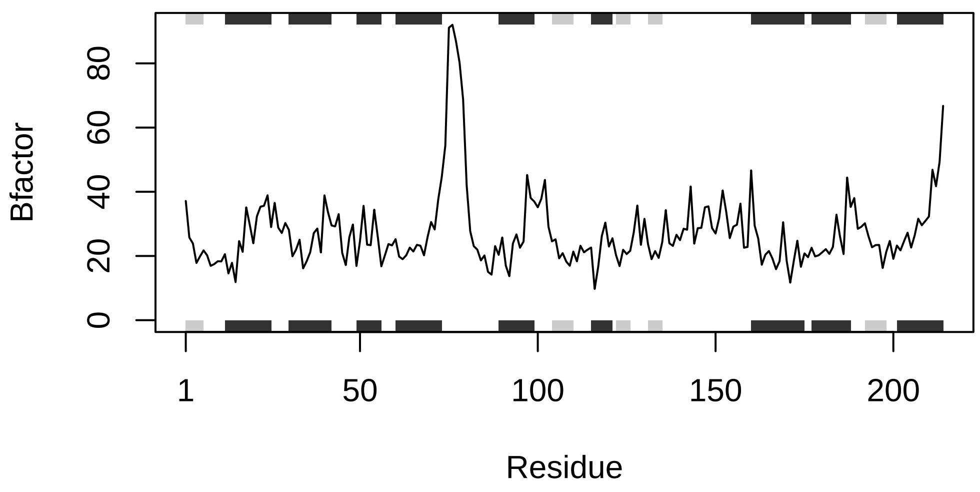
```
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
# General statement: sx.b <- sx.chainA$atom$b
```

`plotb3()` is a function in the `bio3d` package which 'draws a standard scatter plot with optional secondary structure int he marginal regions'. Here it is used to plot the B-factors along the protein sequence, annotated with secondary structure elements (SSE).

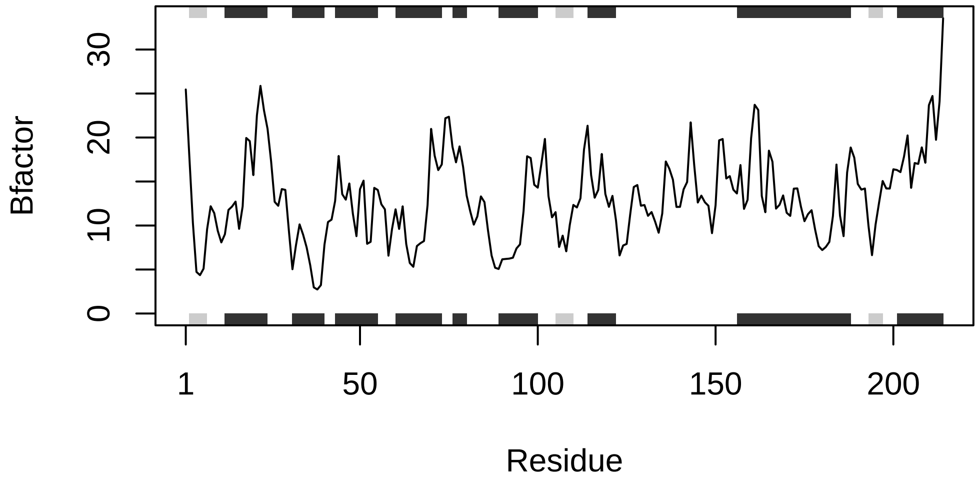
```
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



```
# General statement: plotb3(sx.b, sse=sx.chainA, typ="l", ylab="Bfactor")
```

Now, let's write this into a function:

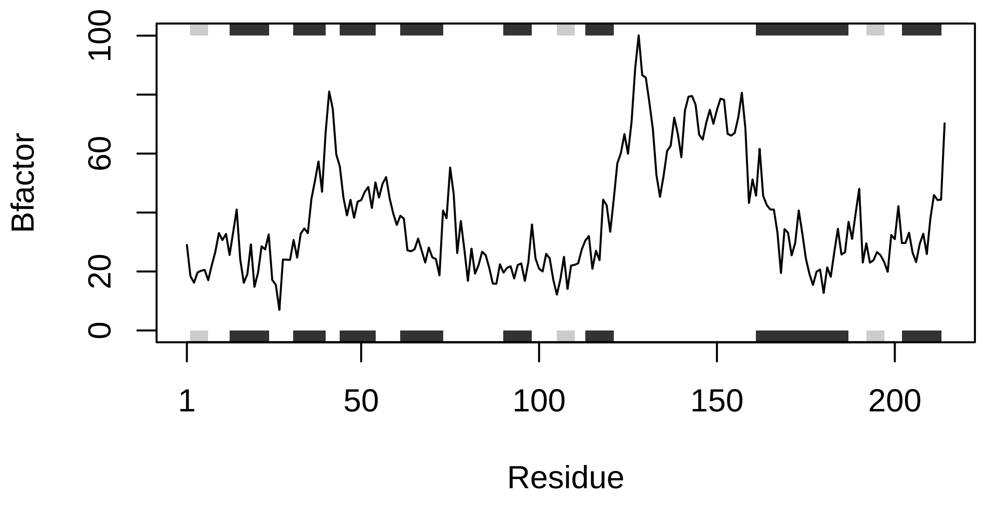
(This is the simplified code I got for my function after getting a working snippet and testing it out.)

```
# Writing a function named bfactor plotter which takes PDB ID as input.
bfactor_plotter <- function(pdb_id){
  # have to import our bio3d package
  library(bio3d)
  # Reads the specified pdb file from the bio3d library
  sx <- read.pdb(pdb_id)
  # Trims the read pdb file and extracts alpha carbon atoms from chain A
  sx.chainA <- trim.pdb(sx, chain="A", elety="CA")
  # Retrieves the B factor from the atom column
  sx.b <- sx.chainA$atom$b
  # Function outputs a plot of the pdb file.
  # The plot is a line graph of the Bfactor for each selected residue and includes the sse.
  plotb3(sx.b, sse=sx.chainA, typ='l', ylab='Bfactor')
}
```

Here I am testing the function out for a given PDB ID.

```r
bfactor_plotter('4AKE')
```

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJ197E/4AKE.pdb exists.
Skipping download
```



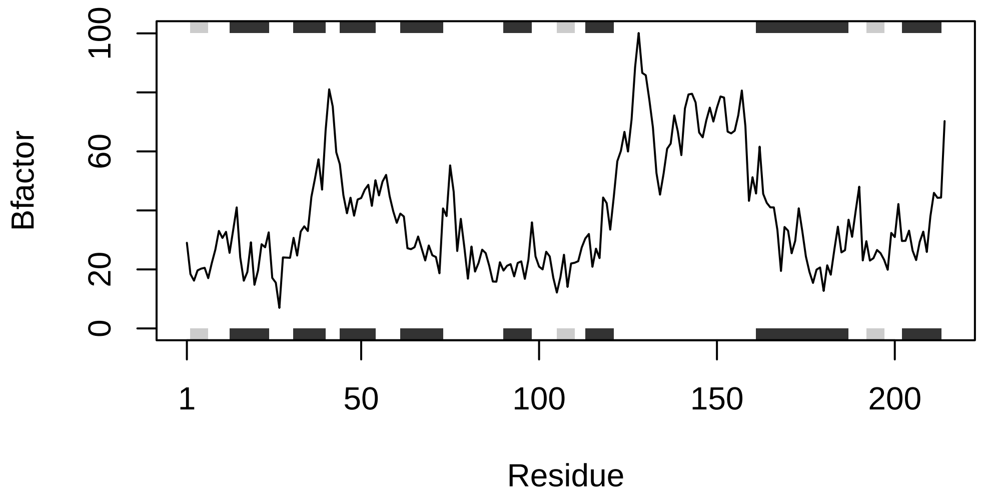
If I want it to work for a set of sequences, I could do (at least) two things:

- If I want to use `apply()`, I can cast my sequence of PDB IDs into a matrix (or array) first.
- Or, I could use `lapply()` which works for vectors.

```r
apply(matrix(c("4AKE", "1AKE", "1E4Y")), 1, bfactor_plotter)
```

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

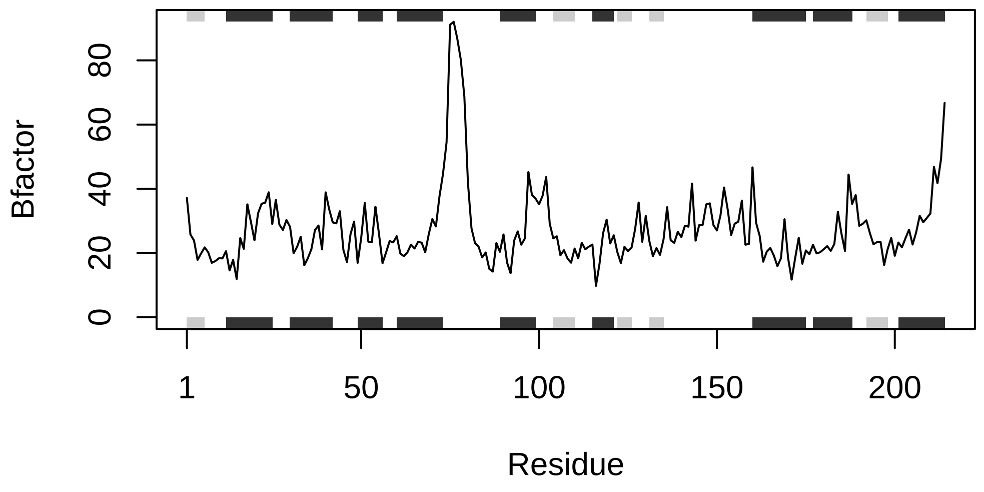
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/4AKE.pdb exists.
Skipping download
```



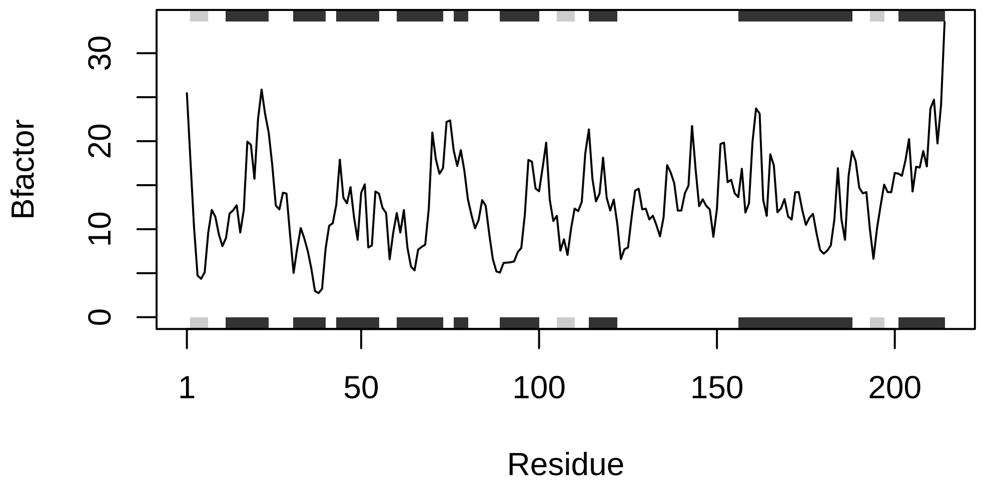
```
  Note: Accessing on-line PDB file
```

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/1AKE.pdb exists.
Skipping download
```

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



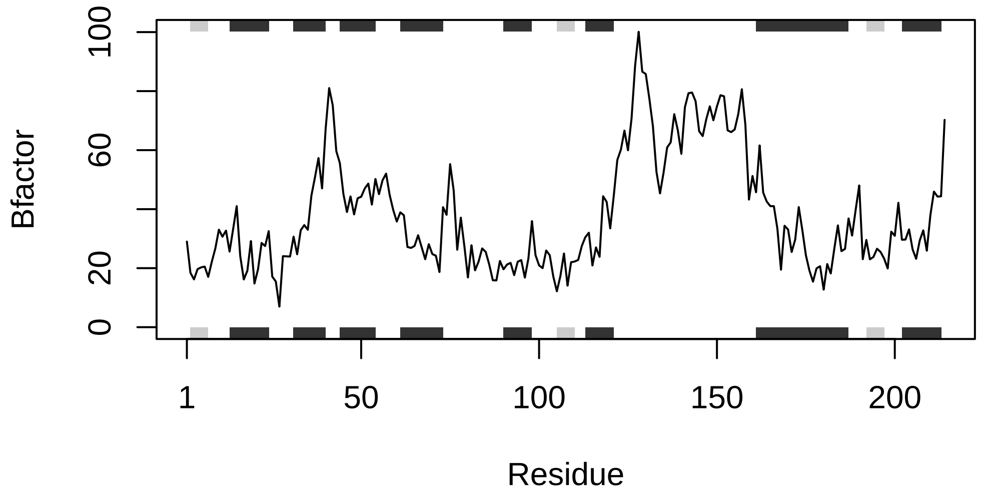
```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/1E4Y.pdb exists.
Skipping download
```



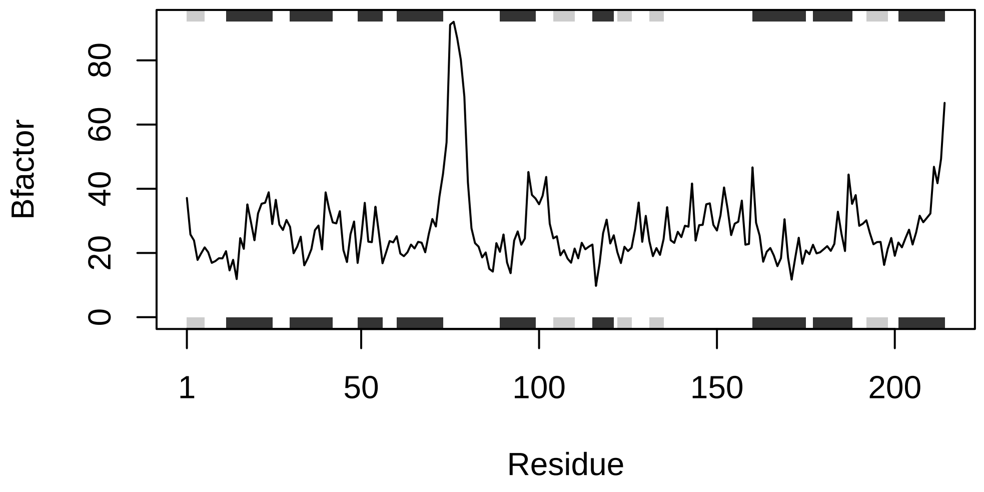
```
NULL
```

```
lapply(c("4AKE", "1AKE", "1E4Y"), bfactor_plotter)
```

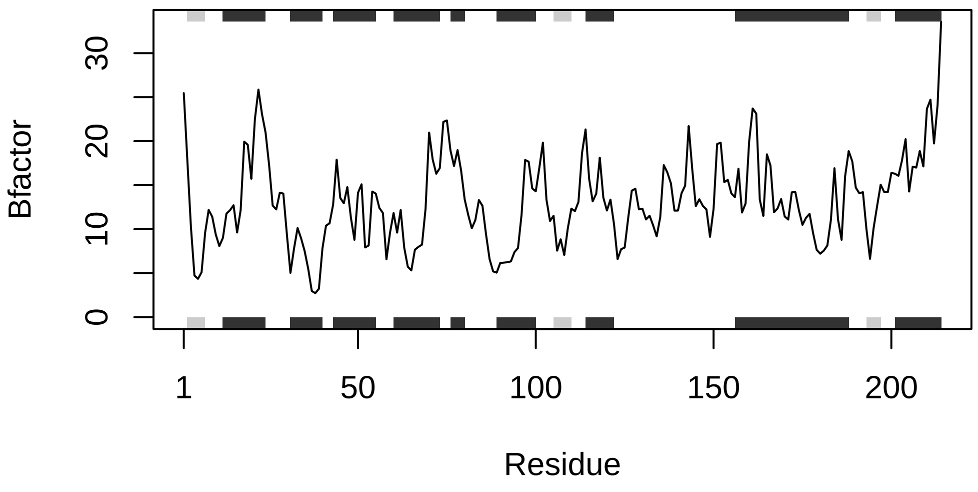
```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/4AKE.pdb exists.
Skipping download
```



```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/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):
/var/folders/pn/x_nfcj0n3t1gqpyk_pzwbwth0000gn/T//RtmprJl97E/1E4Y.pdb exists.
Skipping download
```



```
[[1]]
NULL

[[2]]
NULL

[[3]]
NULL
```