Class06:Homework Function

Bianca Barriga

The provided code that we want to improve and adapt to be more general for # Can you improve this analysis code? library

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

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

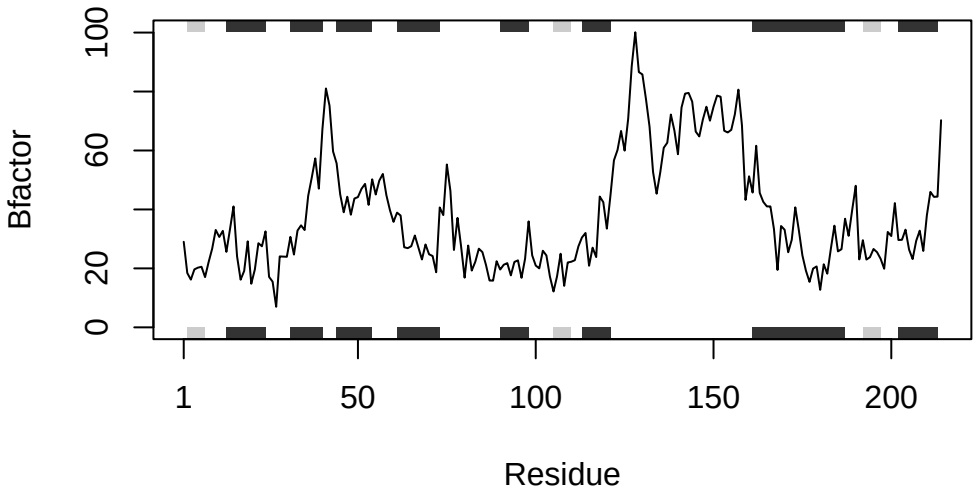
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

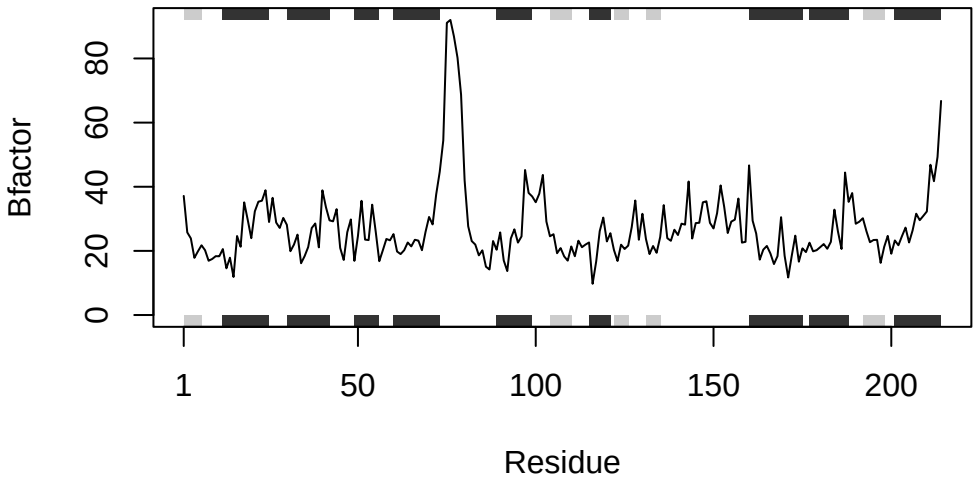
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

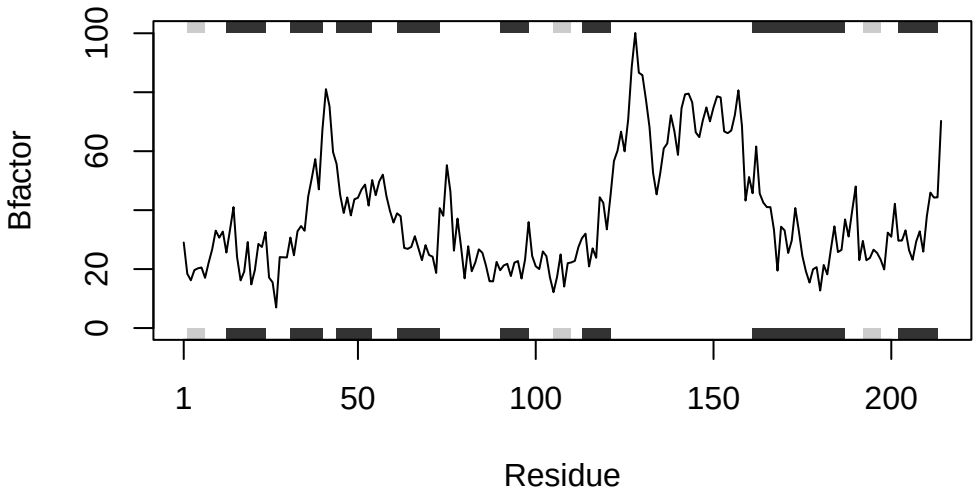
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")

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")
```



Plot 1 and 3 are the same. correct the code

```
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/vt/485t7s5j0j1dw7vcn5mcxhb80000gn/T//RtmpK6T3QX/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):
/var/folders/vt/485t7s5j0j1dw7vcn5mcxhb80000gn/T//RtmpK6T3QX/1AKE.pdb exists.
Skipping download
```

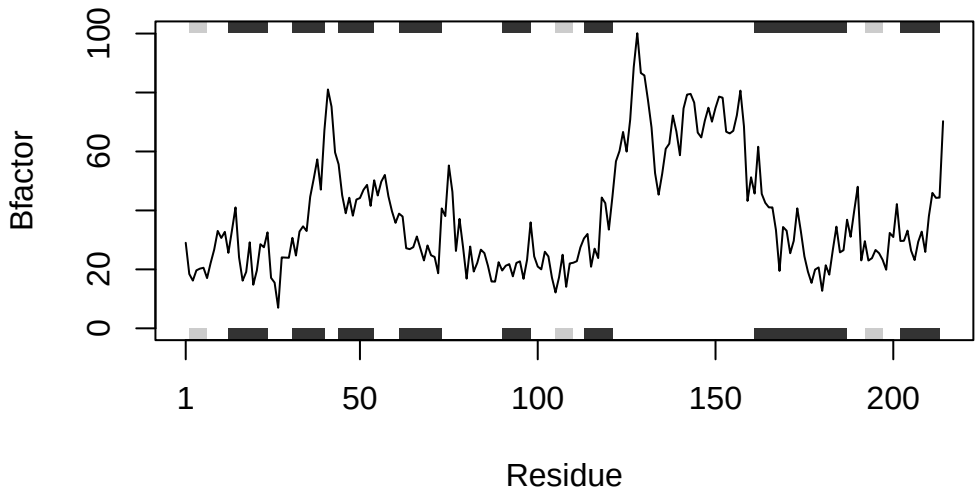
```
 PDB has ALT records, taking A only, rm.alt=TRUE
```

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

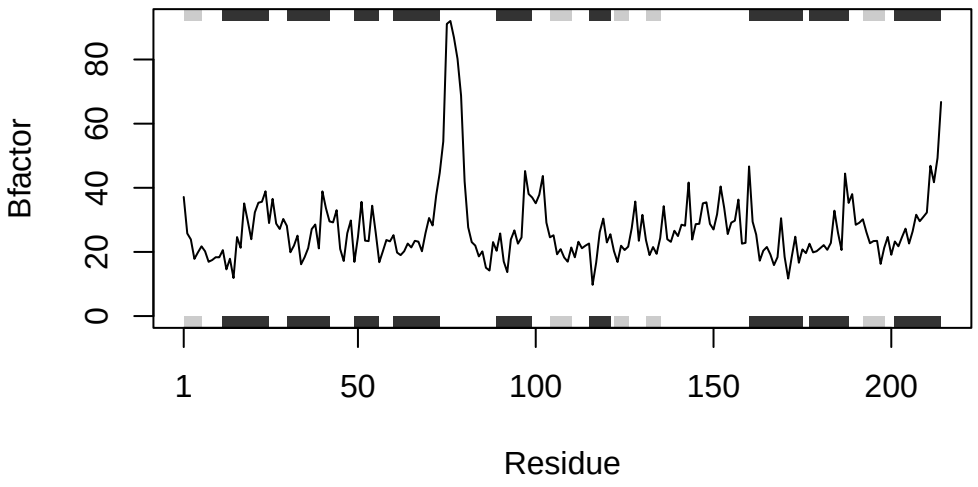
```
  Note: Accessing on-line PDB file
```

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/vt/485t7s5j0j1dw7vcn5mcxhb80000gn/T//RtmpK6T3QX/1E4Y.pdb exists.
Skipping download
```

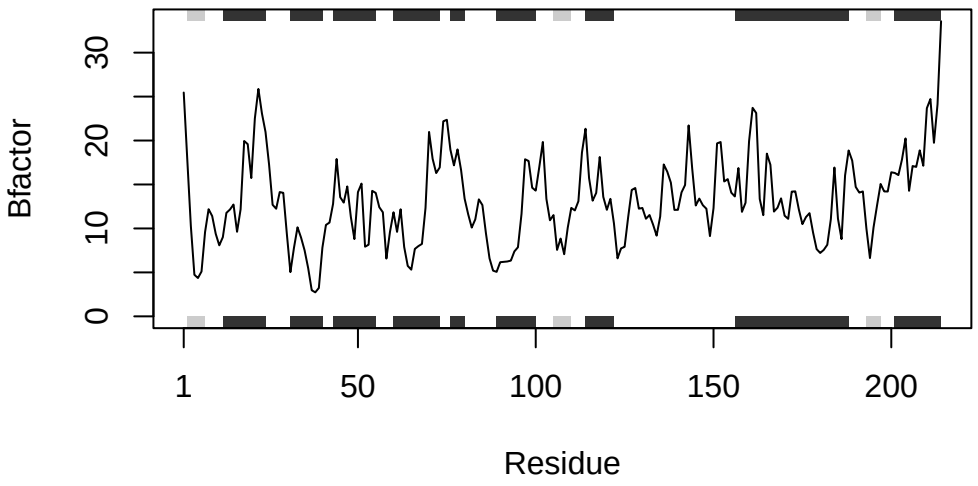
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

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="l", ylab="Bfactor")
```



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




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



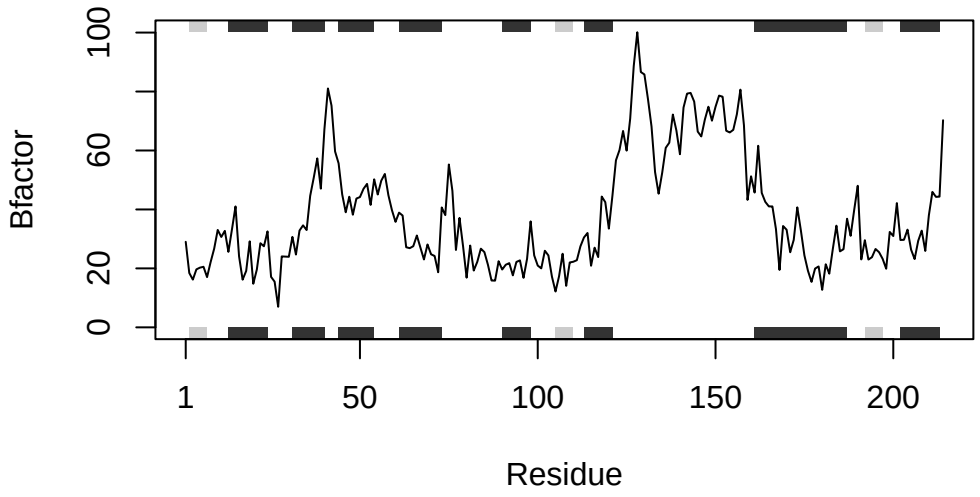
Now simplify the code - delete the lines that look the same, the only difference was the input

```
library(bio3d)

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

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/vt/485t7s5j0j1dw7vcn5mcxhb80000gn/T//RtmpK6T3QX/4AKE.pdb exists.
Skipping download
```

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

s1.b <- s1.chainA$atom$b

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



#code extract function

```
library(bio3d)

#' @param_id The PDB code of your unput structure
#'
#' @return A plot of B factor values
#'
#' @ export
#'
#' @examples pdblot("AKE")

id <- "1AKE"
pdbPlot <- function(id) {
  #read the input PDB file
  s1 <- read.pdb(id) # kinase with drug
  #Extract a subset of chain A
  chain <- trim.pdb(s1, chain="A", elety="CA")
  #
  #s1.b <- s1.chainA$atom$b
  #plot
  plotb3(chainA$atom$b, sse=chain, typ="l", ylab="Bfactor")
```

```
}
```

#use apply function over columns and rows where Ids are listed to apply the function over the multiple ids in for the dataset.

pdb <- function(id="1E4Y") { s1<- read.pdb(id) chain <- trim.pdb(s1, chain="A", elety ="CA") plotb3(chainA$atom$b, sse=chain, typ="l", ylab="Bfactor") }

Q1. What type of object is returned from the read.pdb() function?