Class 6 Homework

Lilith Sadil

2024-04-27

Simplifying Code with Functions

In order to run the program, we first have to install the bio3d package. Going forwards, we will also have to call this package into our library (in other words, bring it to the R "brain").

The initial code provided is repetitive and leaves room for error as each iteration of the code is copy/pasted. This code can be more effectively be written using a function in order to reduces its repetitiveness.

```
library(bio3d)

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

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

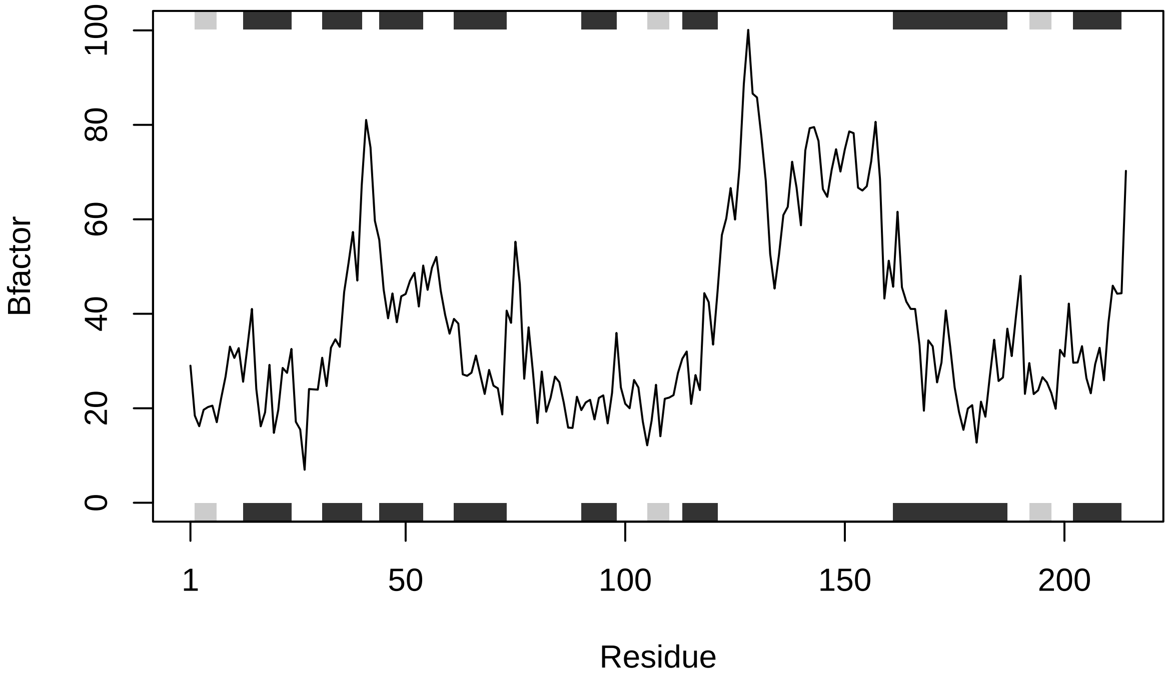
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

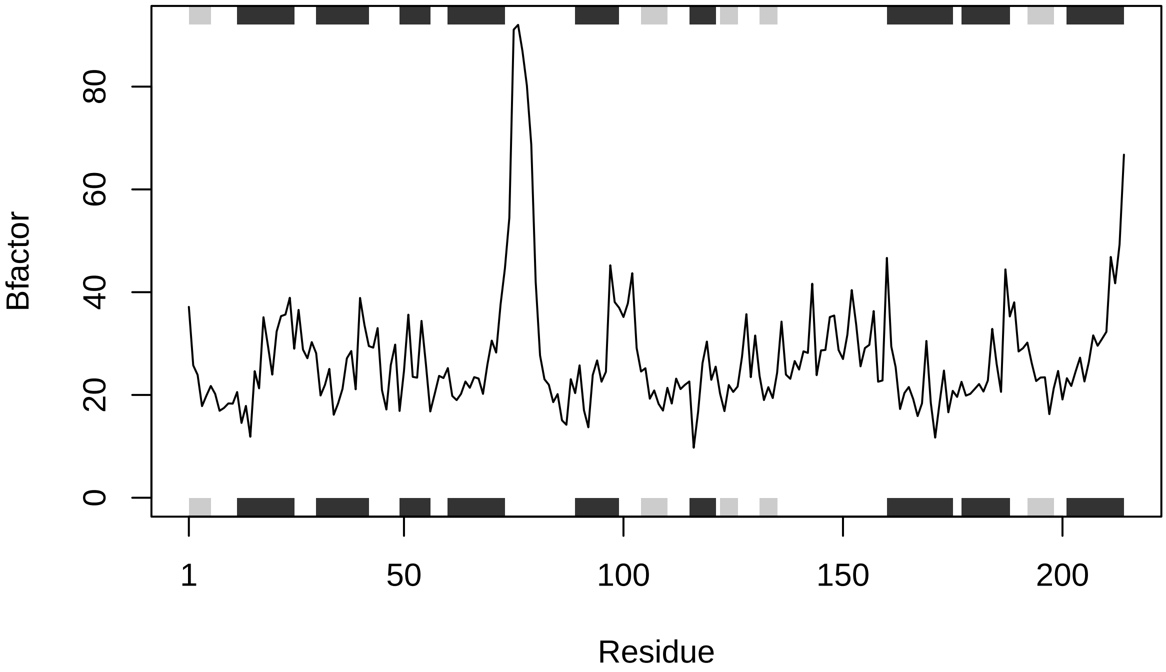
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

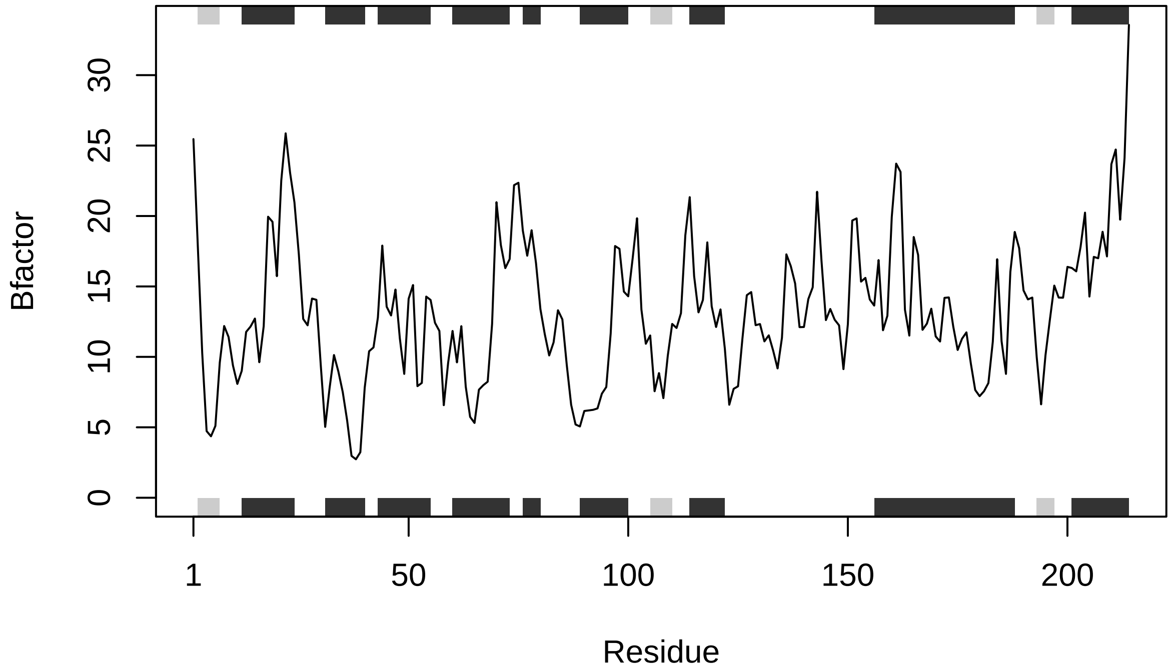
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

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")
```



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

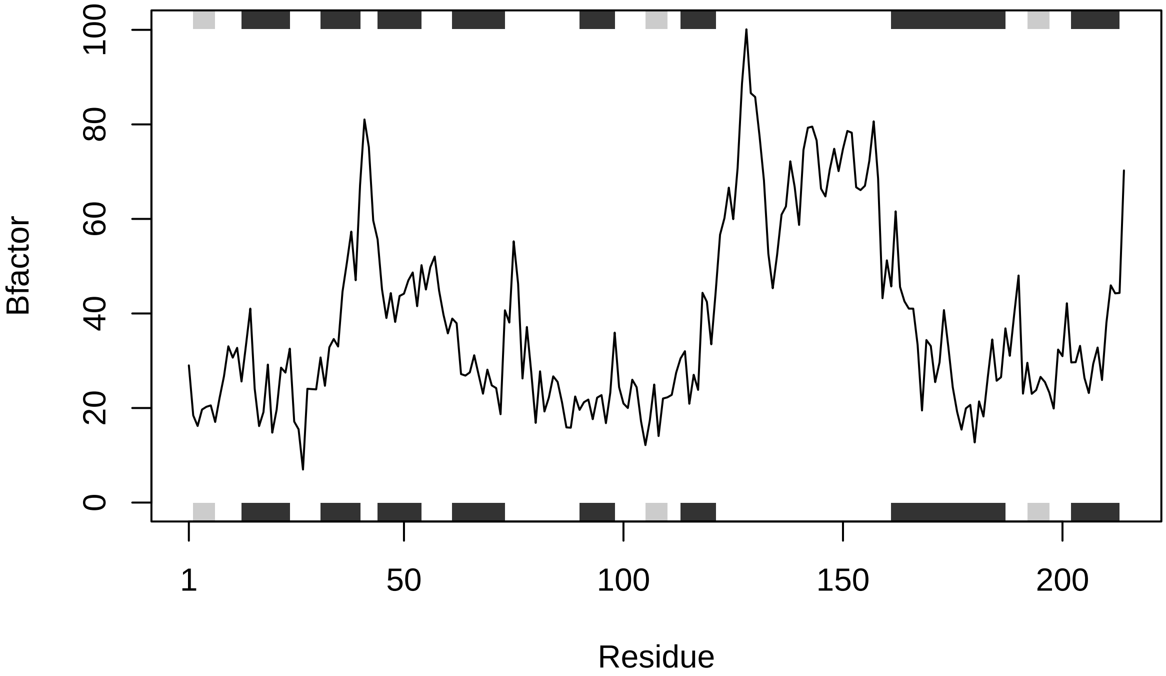


Here, the function, “creating_plots”, takes an input called “files” which may be a single file name or a vector of file names. If “files” is provided as a vector with multiple files, then each file gets run through the “for” loop in order to generate its own unique plot. Plugging in three file names - as seen in the function call at the bottom of the code - will generate a unique plot of Residue vs Bfactor for each input. If the user wishes to create plots for other files, then only the value of “files” needs to be updated - rather than the entire body of code.

```
library (bio3d) #Bringing the bio3d dataset to the R brain

creating_plots = function(files){
  for (files in files) {
    #the "for" statement runs through all the files in the provided "files" vector/variable/list
    s = read.pdb(files)
    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")}}

creating_plots(c("4AKE", "1AKE", "1E4Y")) #vector of file names whose plots in bio3d will be made
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/xp/0vsmcmld6d108kqx8jwtnhqw0000gn/T//RtmpRMDbj7/4AKE.pdb exists.
## Skipping download

##   Note: Accessing on-line PDB file

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

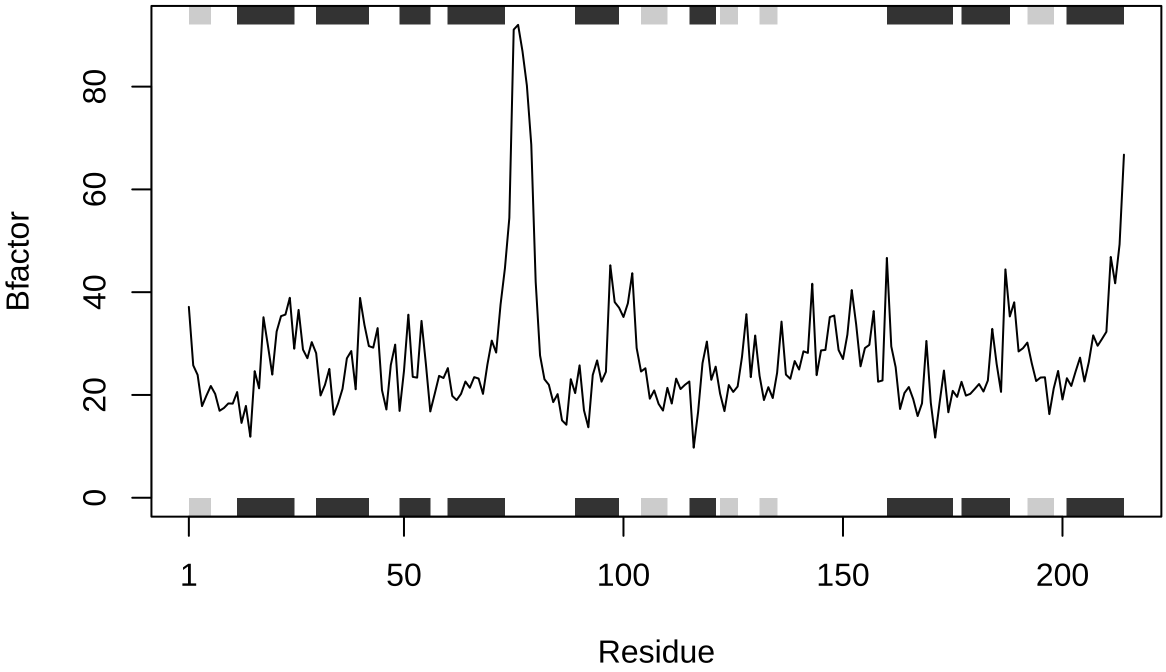
```
## /var/folders/xp/0vsmcmld6d108kqx8jwtnhqw0000gn/T//RtmpRMDbj7/1AKE.pdb exists.
## Skipping download
```

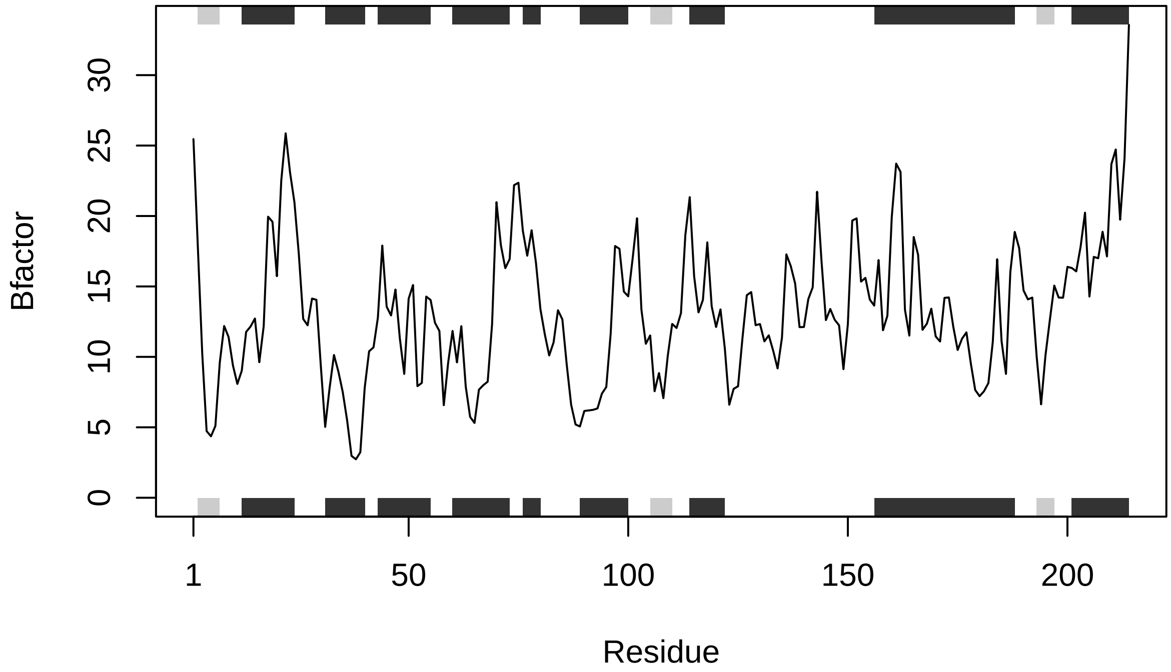


```
##    PDB has ALT records, taking A only, rm.alt=TRUE
```

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/xp/0vsmcmld6d108kqx8jwtnhqw0000gn/T//RtmpRMDbj7/1E4Y.pdb exists.
## Skipping download
```


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


Bfactor
0
5
10
15
20
25
30
1
50
100
150
200
Residue