

hw6

Meha Thakur

##Q6. How would you generalize the original code above to work with any set of input protein structures?

Write your own function starting from the code above that analyzes protein drug interactions by reading in any protein PDB data and outputs a plot for the specified protein. Create a new RMarkdown document with your function code AND example output. We also suggest you include narrative text that address the rubric items below. Generate a PDF report and submit this PDF to our GradeScope site.

code to improve:

```
# Can you improve this analysis code?  
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

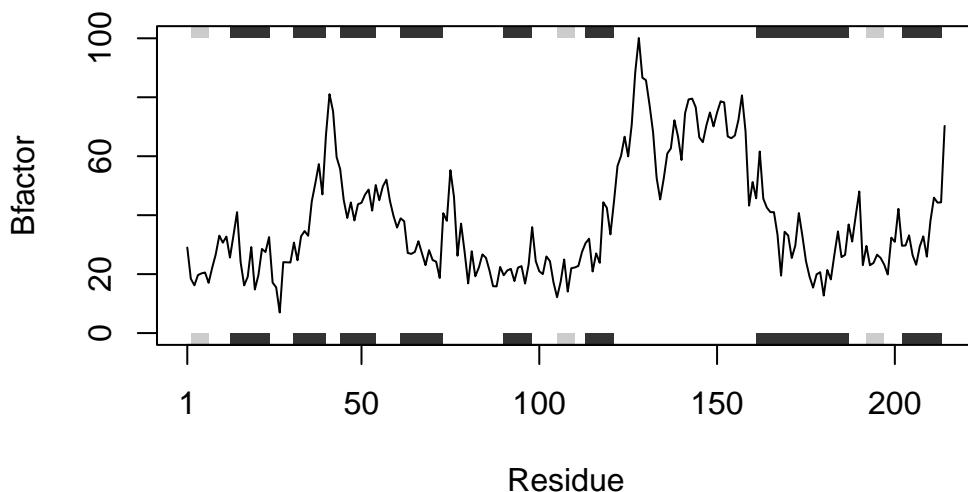
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

Note: Accessing on-line PDB file

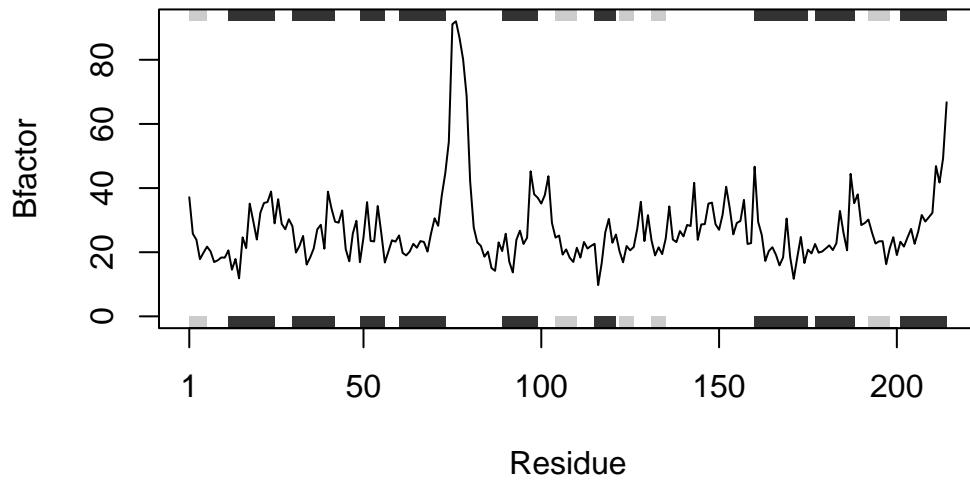
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b

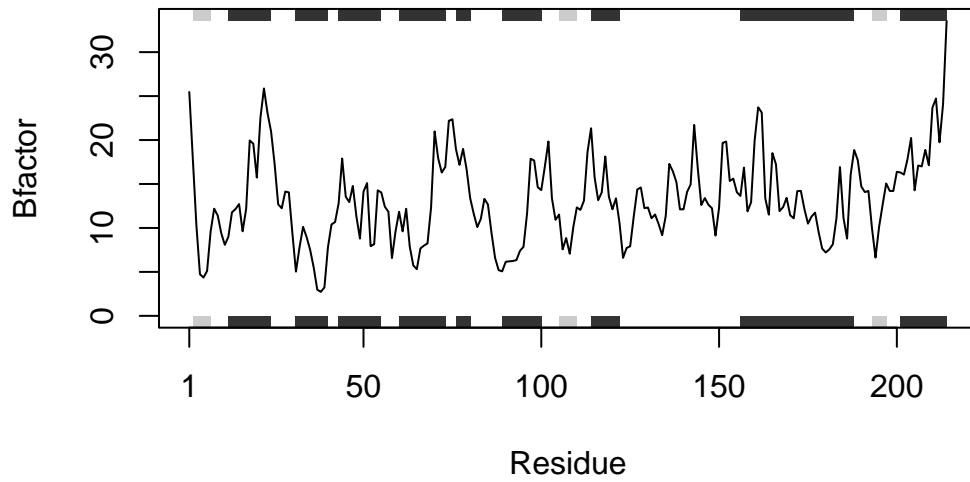
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



my code:

```
#install.packages("bio3d")
library(bio3d)

#first chunk is reading 3 pdb files
pdb_id<- c("4AKE","1AKE","1E4Y") #column vector of my pdb access codes
pdb_list<- lapply(pdb_id,read.pdb) #apply read function to my codes into a list
```

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/p8/xkd2bg1s42scryv4r2qtzdvw0000gn/T//RtmpdXf5EC/4AKE.pdb exists.
Skipping download
```

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/p8/xkd2bg1s42scryv4r2qtzdvw0000gn/T//RtmpdXf5EC/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/p8/xkd2bg1s42scryv4r2qtzdvw0000gn/T//RtmpdXf5EC/1E4Y.pdb exists.
Skipping download
```

```
names(pdb_list)<- pdb_id #name them in the list

chainA<-lapply(pdb_list,trim.pdb, chain="A", elety="CA") #applying the above function to my list

#write a function for parsing out the list

b<- function(chain){
  val<-list() #initialising an empty list to put the result in

  for (i in 1:length(chain)){ #iterating through each of the 3 list elements
    val[[i]]<-chain[[i]]$atom$b #find the b vector for each element of the list, store in con
  }
}
```

```

    return(val) #return the now full list
}

bcol<-b(chainA) #returns b vector as a list of 3, one vector for each pdb code, stored in bcol

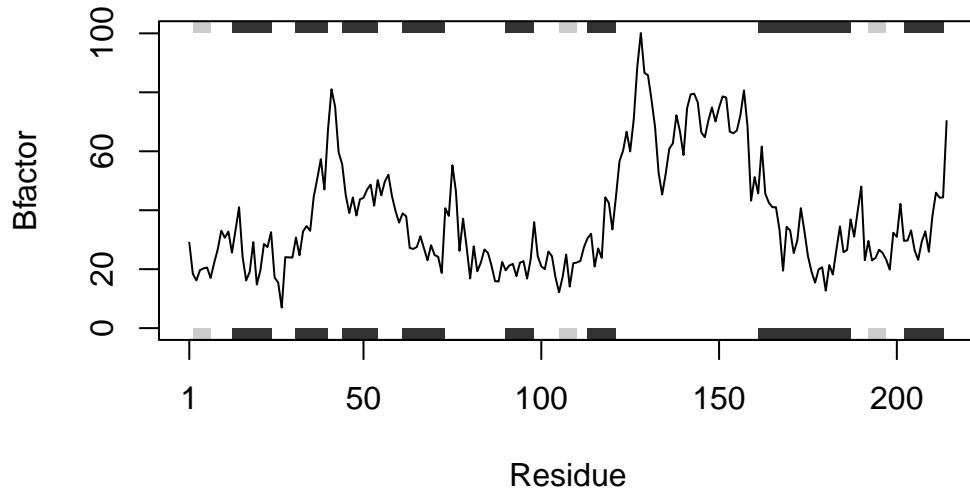
plotb<-function(bfactor,chain){
  plotv<-list() #initialising plotv as the empty plot list i will fill in this function
  for(i in 1:length(bfactor)){ #iterate over the length of b factor (one for each element)

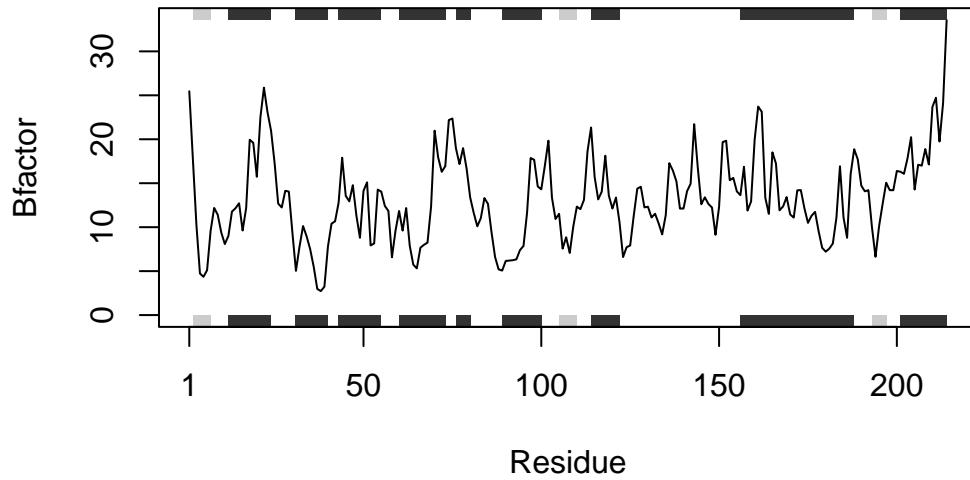
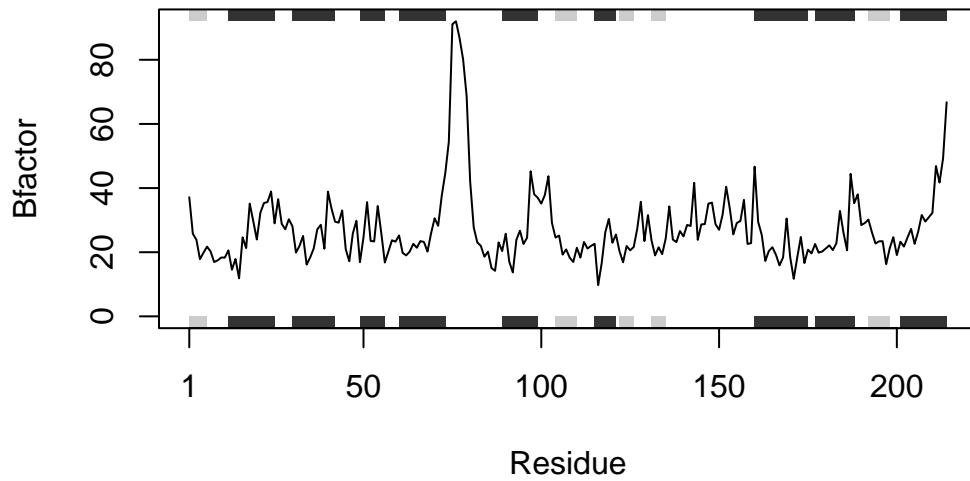
    plotv[[i]]<-plotb3(bcol[[i]],sse=chain[[i]],type="l",ylab="Bfactor") #using [[i]] to access elements in bcol

  }
  return(plotv)
}

plotb(bcol, chainA) #apply the previous function to plot b factor and alpha chain

```





```
list()
```