

class06hw

Bayah Essayem (A17303992)

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

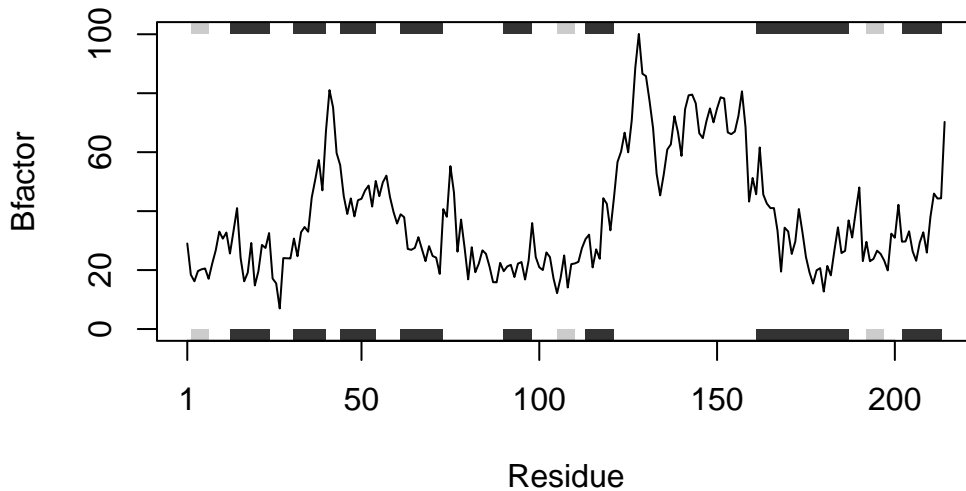
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



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. (See class lecture 9 for further details).

“4AKE” and “1E4Y” have different information. s3.chainA have the same result. Also, the 1st and 3rd plot are the same. try and sum up s1,s2,s3 as one function, make it so the chain command applies to s1,s2, and s3 with a function.

Remember to always recall the package otherwise R won't understand the variables

```
library(bio3d)
```

Creating a function that can generalize the plotting of PDB protein

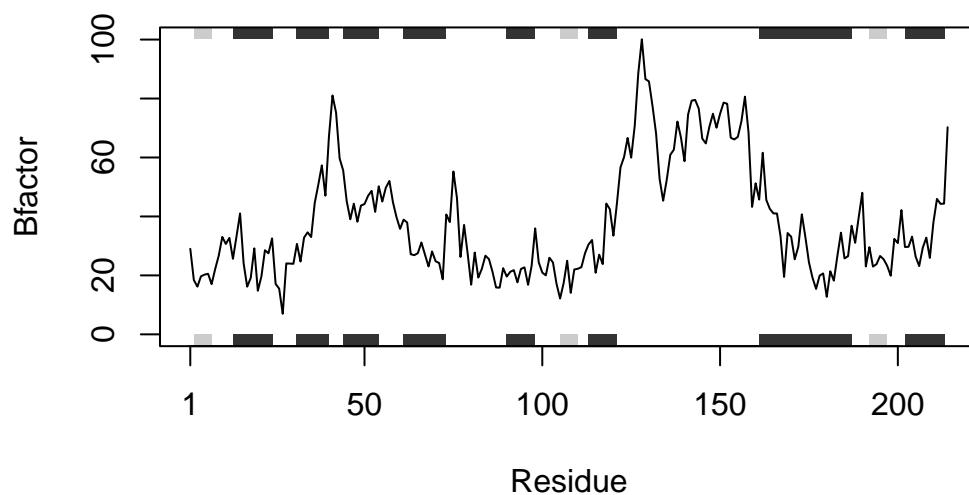
```
plot_PDB_protein <- function(PDB_code){  
  
  # This makes the read.pdb summed into just "s" instead of s1,s2,s3 and this way  
  # the code can read all the variables with one command. Basically,  
  # **read.pdb reads PDB_code**  
  
  s <- read.pdb (PDB_code)  
  
  # Having s.chainA allows you to customize the information on the plot so that  
  # it'll only contain type A chains and elty to specify the type of atom.  
  
  # **chain type A and CA (alpha carbon) atoms are filtered**  
  
  s.chainA <- trim.pdb(s, chain = "A", elty = "CA")  
  s.b <- s.chainA$atom$b  
  
  # Now, there can be modifications to the plot like ylab giving the y axis a  
  # title, and typ making it a line graph  
  
  # **plot is added to function with filters: line graph, yaxis title "Bfactors",  
  # and sse as s.chainA variable from earlier**  
  
  plotb3(s.b, sse = s.chainA, typ = "l", ylab = "Bfactor")  
}
```

Function can now be used and ran

```
sapply(X = c("4AKE", "1AKE", "1E4Y"), FUN = plot_PDB_protein)
```

Note: Accessing on-line PDB file

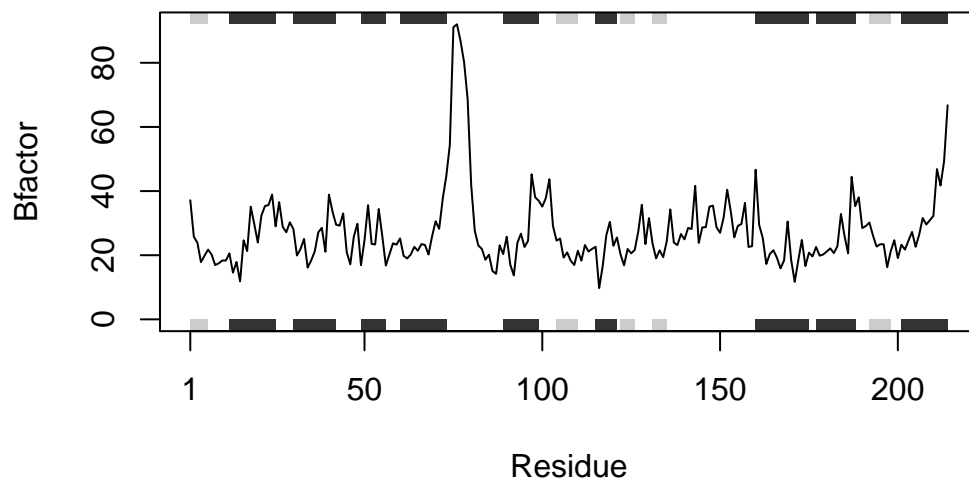
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\profe\AppData\Local\Temp\RtmpKsPgvV\4AKE.pdb exists. Skipping download



Note: Accessing on-line PDB file

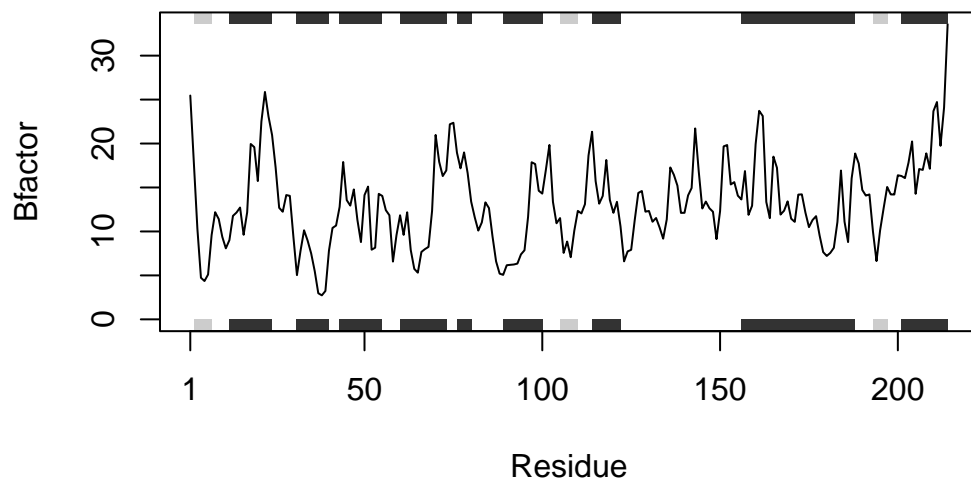
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\profe\AppData\Local\Temp\RtmpKsPgvV\1AKE.pdb exists. Skipping download

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):
C:\Users\profe\AppData\Local\Temp\RtmpKsPgvV\1E4Y.pdb exists. Skipping download



\$`4AKE`
NULL

\$`1AKE`
NULL

\$`1E4Y`
NULL