

HW6

Alfred Phyo (PID:A18129346)

Table of contents

Original Code	1
Answer	4
Example output	4
Documentation (narrative text)	5

Original Code

```
# Can you improve this analysis code?  
library(bio3d)  
s1 <- read.pdb("4AKE") # kinase with drug
```

Note: Accessing on-line PDB file

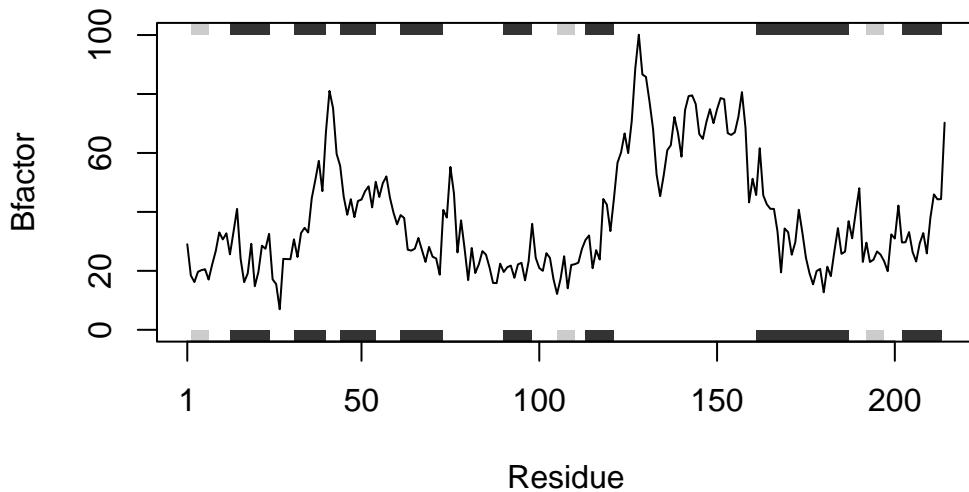
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

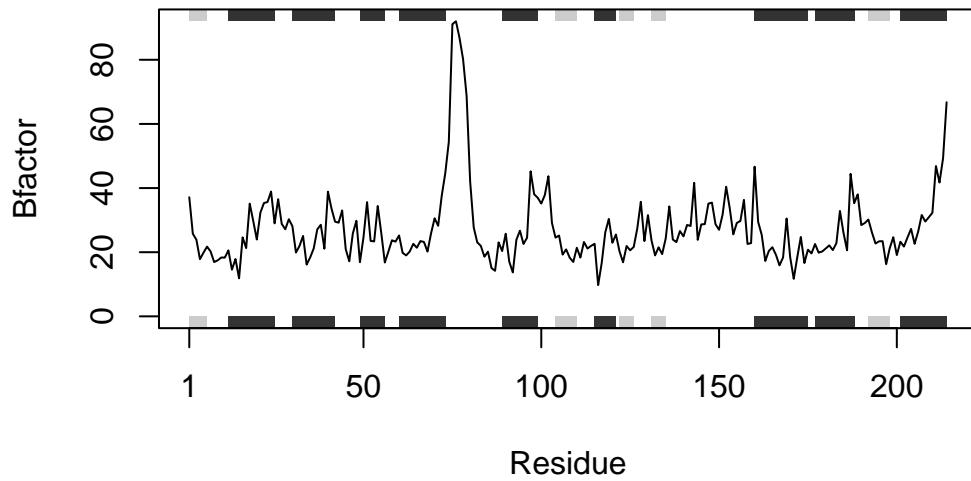
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

Note: Accessing on-line PDB file

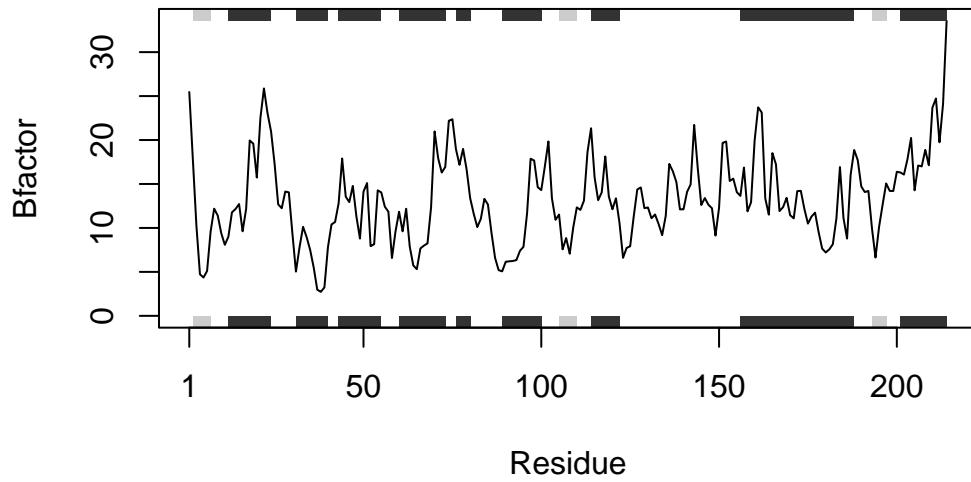
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



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



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



Answer

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

```
library(bio3d)

plot_protein <- function(x){
  ans <- read.pdb(x)
  ans.chainA <- trim.pdb(ans, chain="A", elety="CA")

  ans.b <- ans.chainA$atom$b

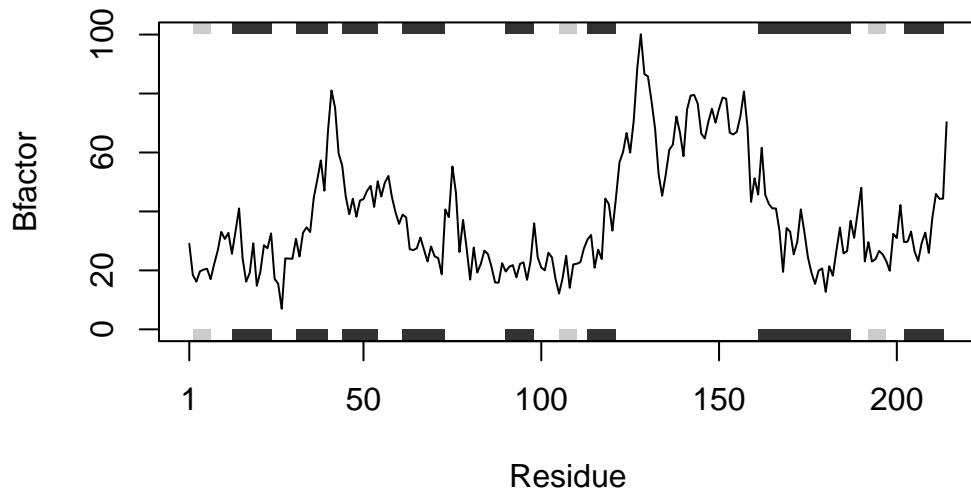
  plotb3(ans.b, sse=ans.chainA, typ="l", ylab="Bfactor")
}
```

Example output

```
plot_protein("4AKE") #The input to the function are character strings specifying the PDB ID
```

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/c9/3n6chyzs2k7_2c2z3wj3vd180000gn/T//RtmpiMYa11/4AKE.pdb exists.
Skipping download
```



Documentation (narrative text)

The function `plot_protein()` generates a B-factor plot for the user specified protein.

The function can be used by the user when they input a character string specifying the PDB ID available in the database.

The function's output is a B-factor plot that demonstrates how much a protein moves or fluctuates in a crystal structure.