

Lab 6 HW

Lindsey China (A17023629)

2024-04-29

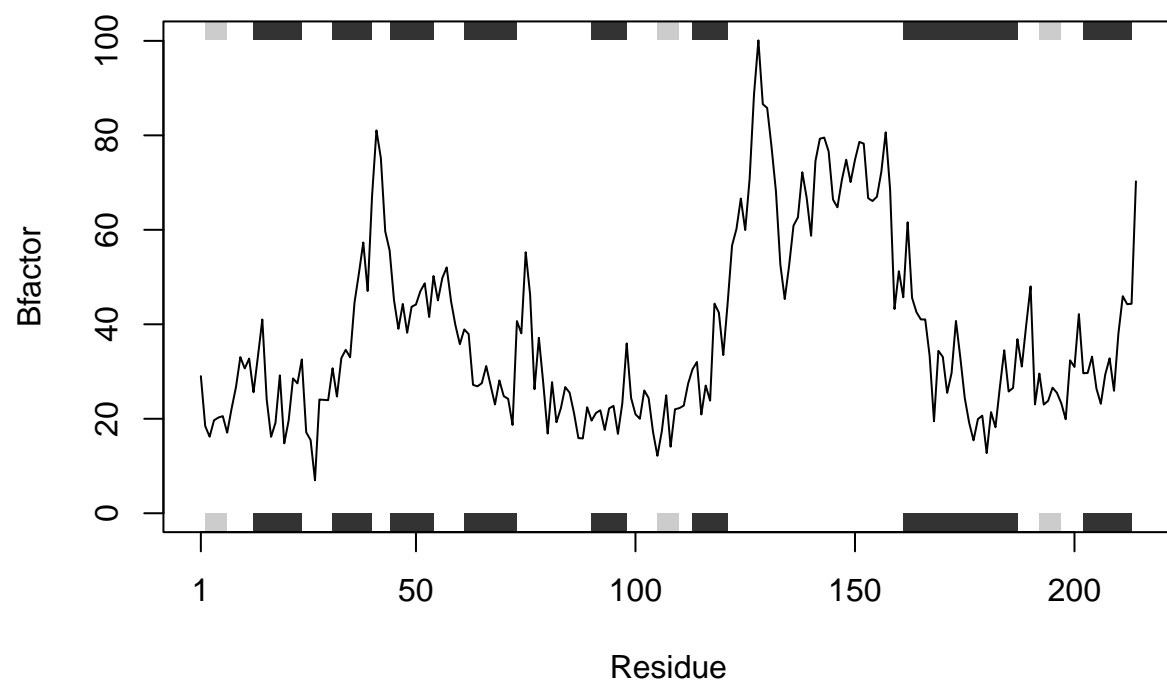
```
library(bio3d)
df <- data.frame(a=1:10, b=seq(200,400,length=10),c=11:20,d=NA)

# Can you improve this analysis code?
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

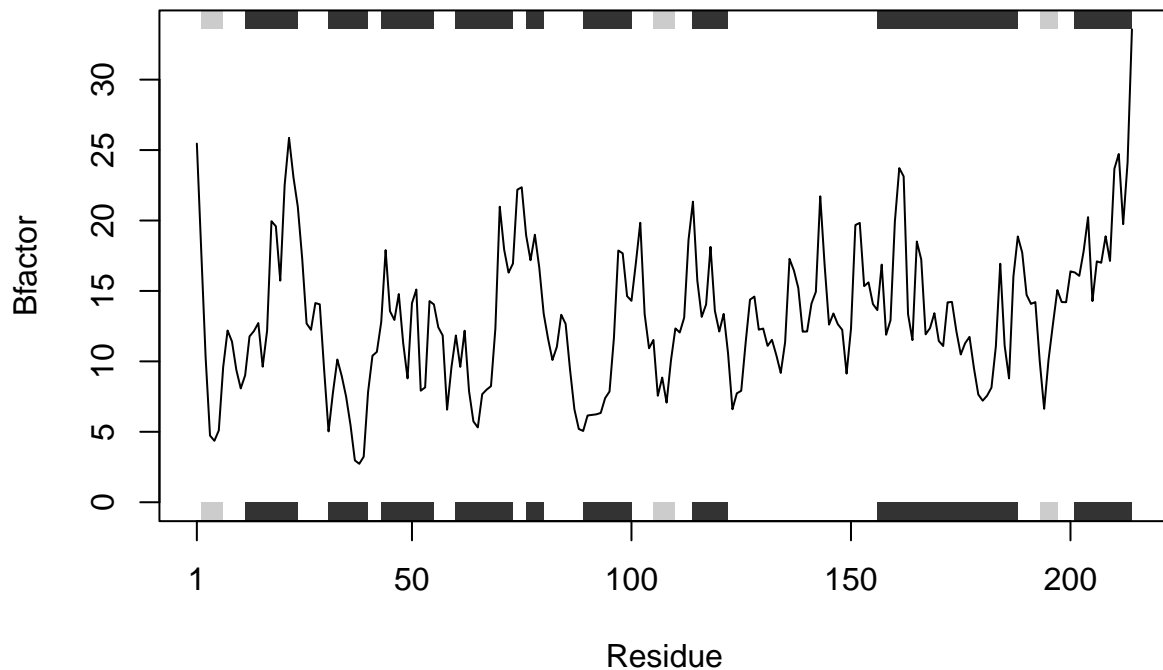
## Note: Accessing on-line PDB file
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
#s3.chainA <- trim.pdb(s1, chain="A", elety="CA") Typo with "s1"
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- 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")
```



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

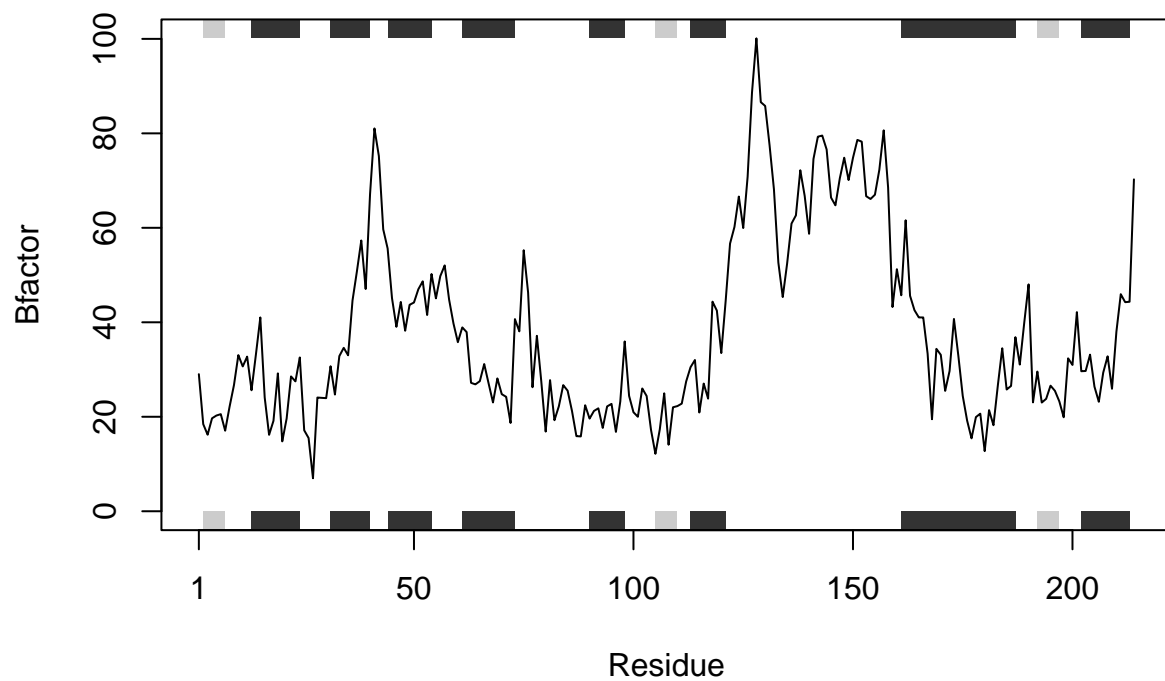
protein_analysis() will read the input X (a protein code, ex. 4AKE) and identify the structure of the protein inputted. It will produce a line graph plotting the residues and Bfactors of the protein.

```
protein_analysis <- function(x){
  # Read the protein
  sX <- read.pdb(x)
  # Trim the data
  sX.chainA <- trim.pdb(sX, chain="A", elety="CA")
  # Specify into b
  sX.b <- sX.chainA$atom$b
  plotb3(sX.b, sse=sX.chainA, typ="l", ylab="Bfactor")
}
```

Testing function:

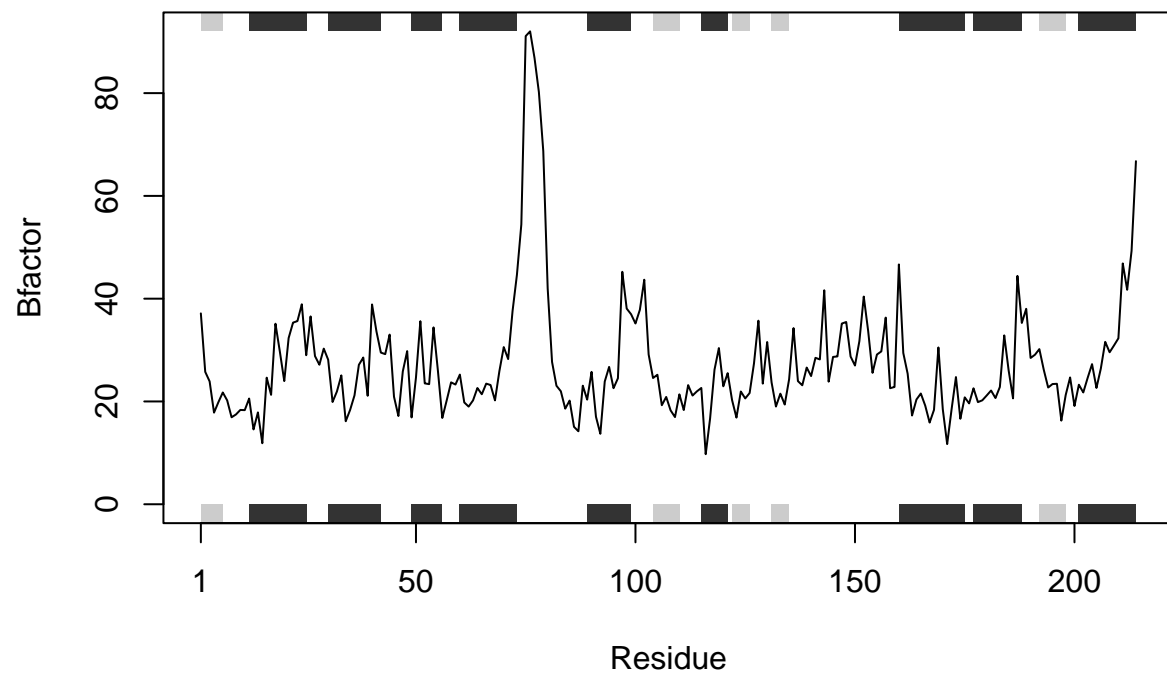
```
protein_analysis("4AKE")
```

Note: Accessing on-line PDB file



```
protein_analysis("1AKE")
```

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



```
protein_analysis("1E4Y")
```

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

