HW6

Richard Gao - PID:A16490010

2024-02-03

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

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

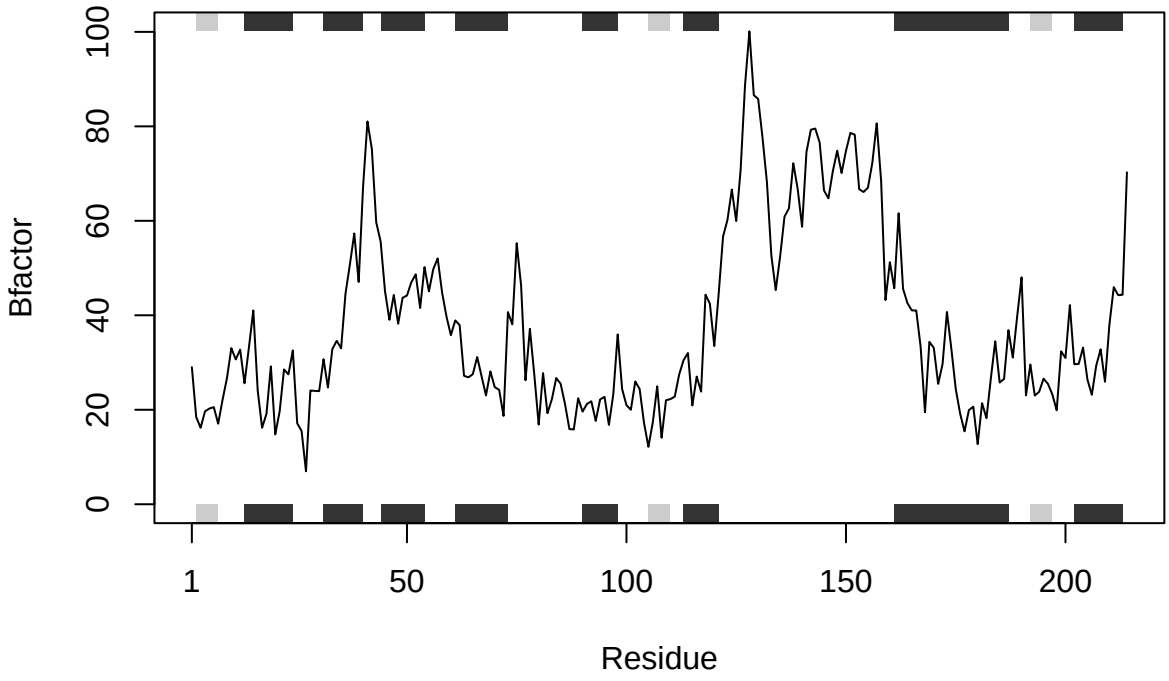
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

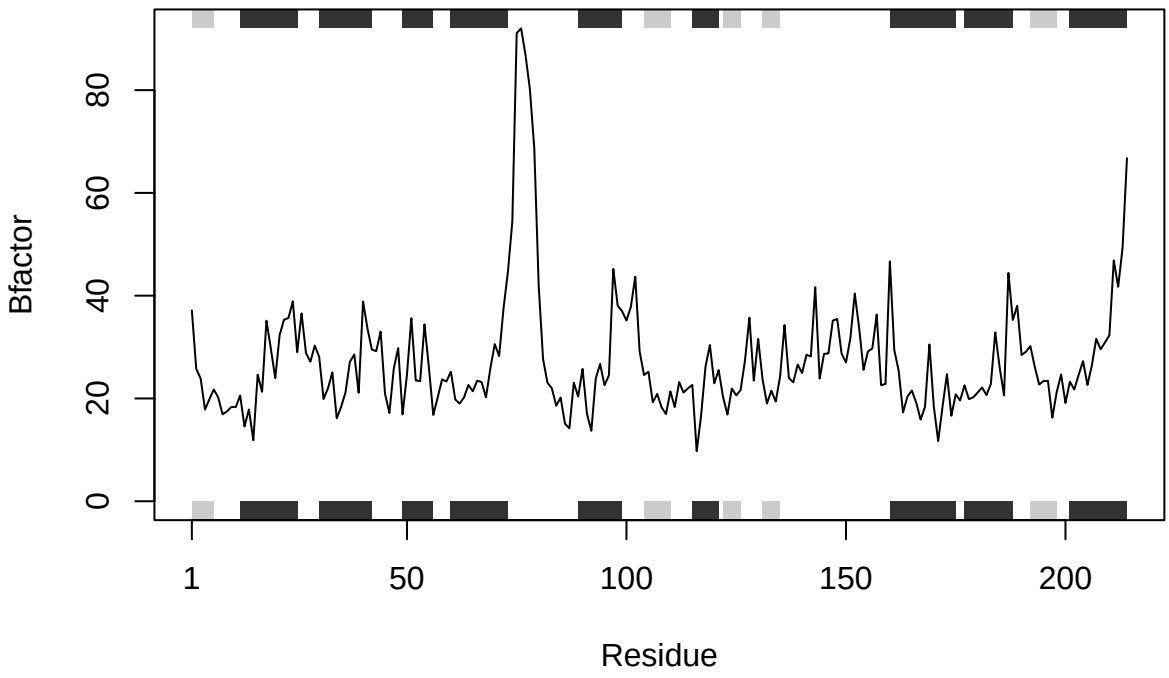
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

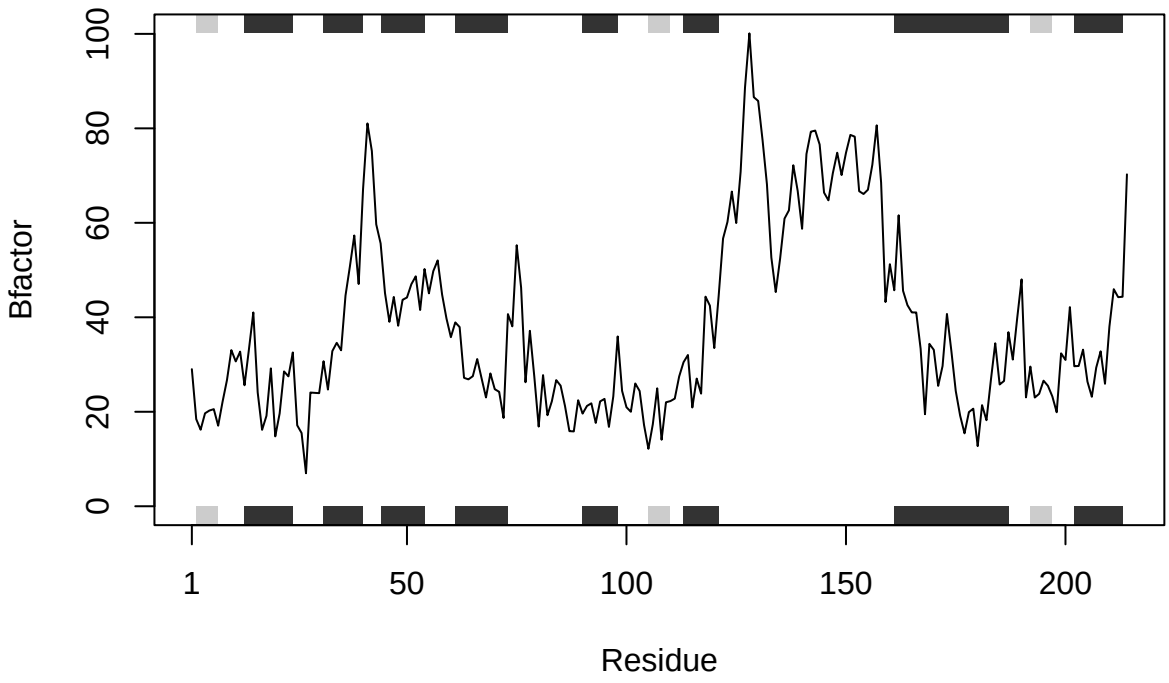
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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




```
View(s1.b)
```

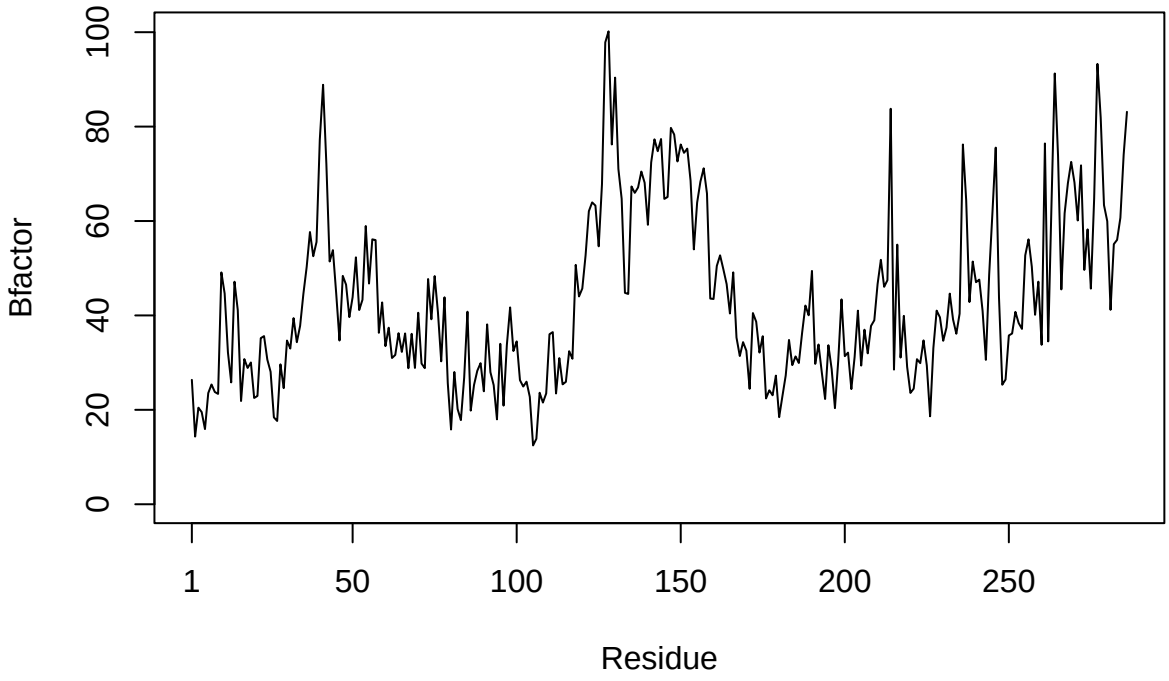
This function will plot the bfactor values of a particular chain of amino acids of a desired protein against the protein chain residue locations. You will have to input the protein file, the chain of interest, and the specific atom you are using as an electrostatic marker for the bfactor values (usually "CA"). The function will output a line graph where the x-axis is the residue location based on the chosen electrostatic marker atoms, and the y-axis is the bfactor which reflects the flexibility of the residue at that location/atom.

```
# input a protein data bank coordinate file, a string variable of the desired
# residue, and a string variable of the elety property
plotProtein <- function(proteinFile, chainName, eletyProperty) {
```

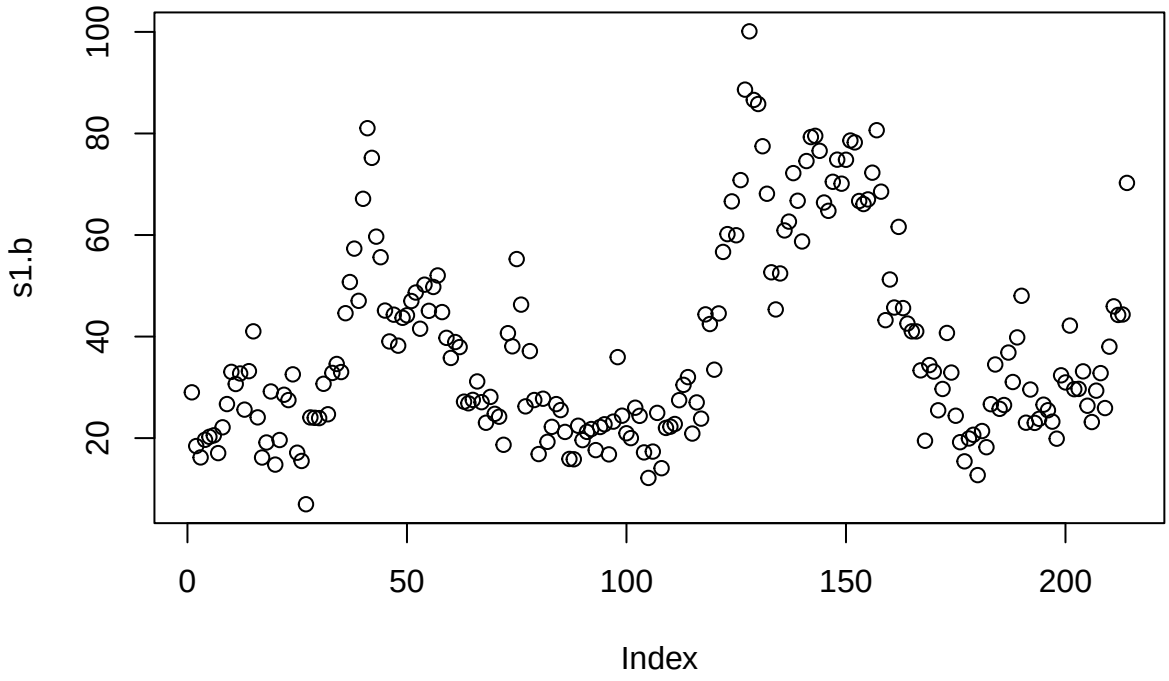
```
  # reads the protein data bank coordinate file and stores it as a list
  # "proteinData" of class "pdb" containing the structural data of the protein
  proteinData <- read.pdb(proteinFile)

  # selects a particular subset of atoms from the "proteinData" pdb object and
  # stores is as a smaller pdb object "proteinChain"
  proteinChain <- trim.pdb(proteinData, chain = chainName, elety=eletyProperty)

  # stores the Bfactor value of each amino acid in the desired chain as a
  # vector "BfactorValues
  BfactorValues = proteinChain$atom$b

  # creates a line plot of the Bfactor (which is reflective of stability) of
  # each amino acid residue in the desired protein chain
  plotb3(BfactorValues, sse=proteinChain, typ = "l", ylab = "Bfactor")
}
```

```
plotProtein("4AKE", "A", "O")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/22/xcwqrjw93r786qj_2z2gf5mw0000gn/T//RtmpvUdxFL/4AKE.pdb exists.
## Skipping download

## Warning in pdb2sse(sse): No helix and sheet defined in input 'sse' PDB object:
## try using dssp()

## Warning in plotb3(BfactorValues, sse = proteinChain, typ = "l", ylab =
## "Bfactor"): Length of input 'sse' does not equal the length of input 'x';
## Ignoring 'sse'
```




```
plot(s1.b)
```


s1.b
20
40
60
80
100
0
50
100
150
200
Index