Class 06 homework

Renee Zuhars (PID: A17329856)

Table of contents

The given code (my solution is under the next heading below):

I have annotated the code below to identify mistakes.

```
# Can you improve this analysis code?

library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

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

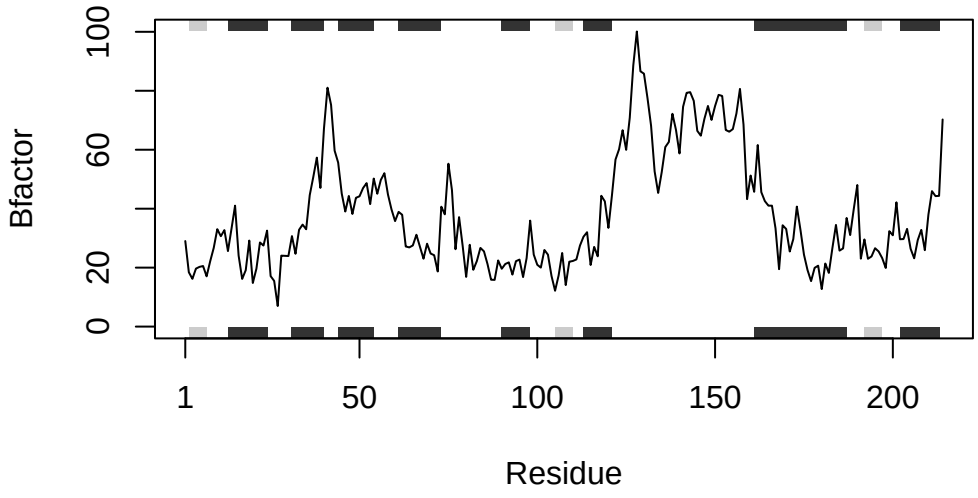
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

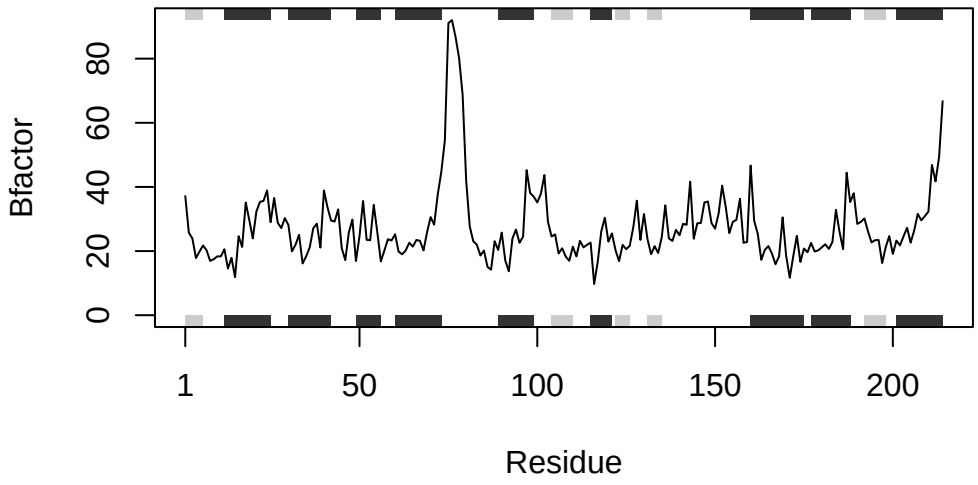
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

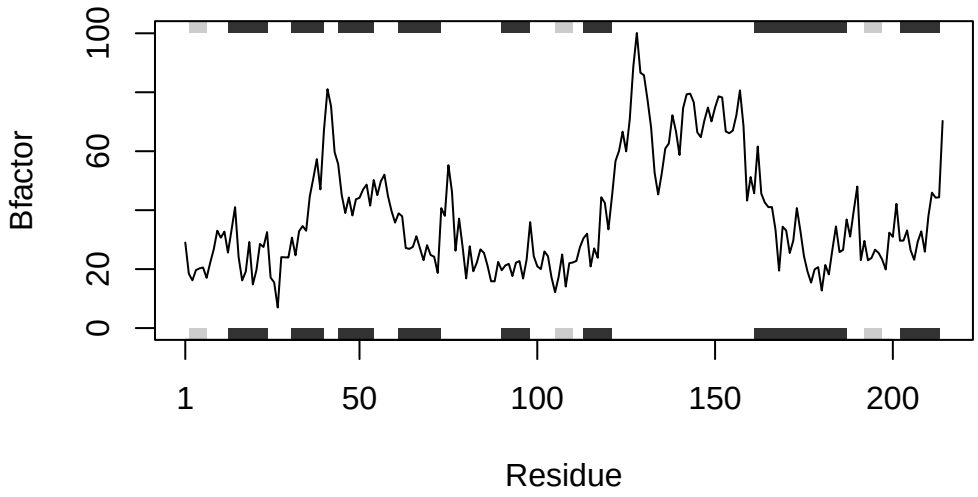
```
#According to chatgpt, the below code serves to obtain the following
#information, needed to analyze/compare flexibility of the b-factors of
#three proteins.

#The code below extracts only chain A and keeps only alpha carbons.

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") #this should say s3 (not s1).

#The code below gets the B-factors for each structure.
#B-factors are a good indication of atomic mobility/flexbility.

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

#The code below plots the B-factors.
#They are annotated with secondary structure using `plotb3()`.

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")
```



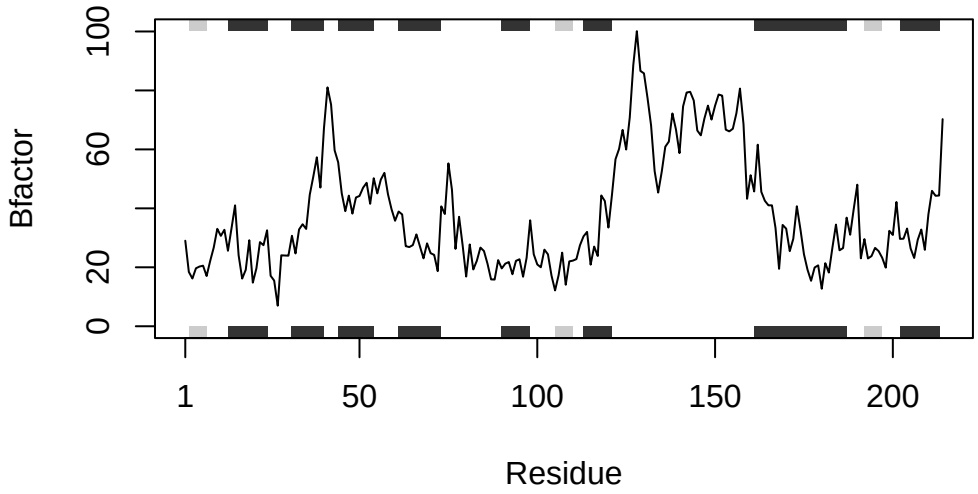
```
#this code appears to run all three sequences of interest separately.
#I want to combine all of this into a single function, reducing repetition.
```

My solution: using a function()

```
library(bio3d)
# The goal of this code is to input any sequence of interest,
#and to return a plot of the sequence's b-factors.

#I asked chatgpt what the argument of `read.pdb()` should be.
#It said it should be `pdb.id`.
#So, the only user input will be the 4-character code of each sequence.

obtain_info <- function(pdb.id) {

  #First, the pdb sequence ID will be input and read

  pdb <- (read.pdb(pdb.id))

  #Then, the data from only chain A is extracted using `trim.pdb()`-
  #we want to extract the same part of each structure of interest to ensure an
  #accurate comparison.

  pdb.chainA <- trim.pdb(pdb, chain="A", elety="CA")

  #Next, the code obtains information regarding the b-factors of each structure
  #in chain A only

  pdb.b <- pdb.chainA$atom$b

  #Finally, the code returns the output of the function- it converts the
  #b-factor data into the desired plot(s).

  return(plotb3(pdb.b, sse= pdb.chainA, typ="l", ylab="Bfactor"))
  }

#below, we use the function created above to return the desired results for
#each sequence of interest that was given.

obtain_info("4AKE") #kinase with drug
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/3d/zscn8ldn54g24rps3fhznh2w0000gn/T//RtmpViIgRd/4AKE.pdb exists.
Skipping download
```

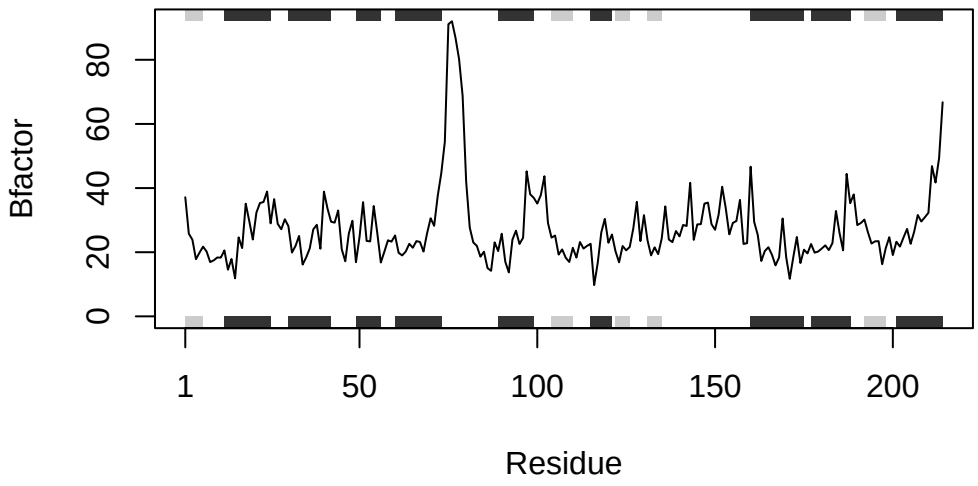



```
obtain_info("1AKE") #kinase with no drug
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/3d/zscn8ldn54g24rps3fhznh2w0000gn/T//RtmpViIgRd/1AKE.pdb exists.
Skipping download
```

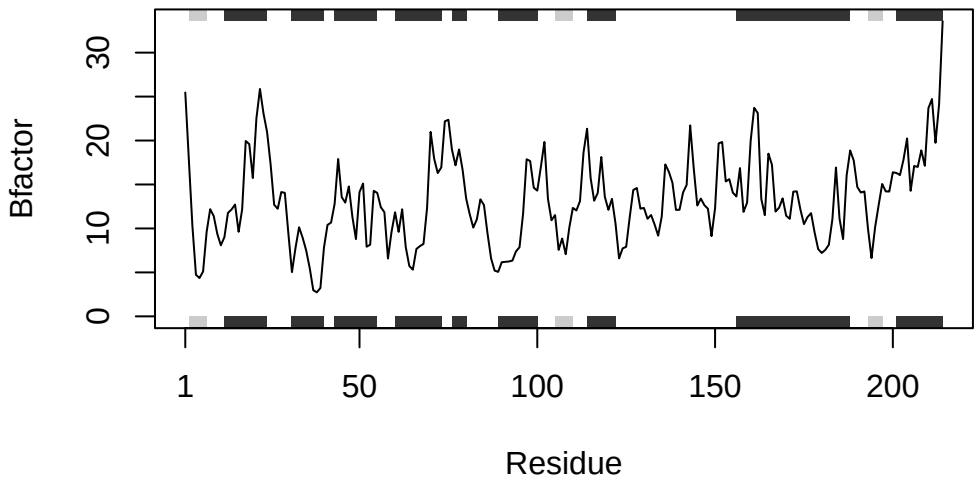
```
   PDB has ALT records, taking A only, rm.alt=TRUE
```




```
obtain_info("1E4Y") #kinase with drug
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/3d/zscn8ldn54g24rps3fhznh2w0000gn/T//RtmpViIgRd/1E4Y.pdb exists.
Skipping download
```




```
#So, the input is the 4-character sequence ID, and the output should be several
#individual plots (3, in this case) that can be used to compare the sequences.
#The output is the same as the given code, but it is more efficient because you
#do not have to type out all the steps that the function does for you!
```