Homework Class 6

Johann Tailor

HomeWork from Class 6:

Here is the code provided:

library(bio3d)

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

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

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

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")

The meat of the code:

LINE 1: s1 <- read.pdb("4AKE") # kinase with drug

LINE 2: s1.chainA <- trim.pdb(s1, chain="A", elety="CA")

LINE 3: s1.b <- s1.chainA*atom*b

LINE 4: plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")

Breaking down the above code:

First Line

Reads the PDBID provided by the user.

Second Line

The code trim.pdb is part of the bio3d package which looks for a particular chain and the atoms or residue within that chain. "Elety" means the element type and "CA" stands for alpha carbon atoms. The code uses the variable "s1" from the first line of the code which equals to protein of interest.

Third Line

This looks for chain of A of the protein of interest and assigns a variable to the specific chain of the protein of interest.

Fourth Line

Plots a graph. s1.b calls for chain A of the protein of interest as in the code above. "sse" specifies the secondary structure elements for atoms in the chain a protein of interest. "typ=1" means a line plot. "ylab" assigns label to y axis. In summary, this code will give an output of a line plot of B-factors with secondary structure for the protein structure of interest.

Here is my optimized code with in depth annotation:

```
# This code loads the plotting feature
library(bio3d)

# I unified the look up for a protein by assigning the value proteinID and generated a fuc
plot_proteinID <- function(proteinID) {

# Assigns a variable to the protein of interest for the purpose of easier coding.
s1 <- read.pdb(proteinID)
```

```
# The following codes looks up the chain of the protein of interest and assigns it a varia
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")

# This looks for chain of A of the protein of interest and assigns a variable to the speci
s1.b <- s1.chainA$atom$b

# The following code plots the variables assigned in the above codes. s1.b calls for chain
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
}

# Call the function. The will give an output of a line plot of B-factors with secondary st
plot_proteinID("1AKE")
```
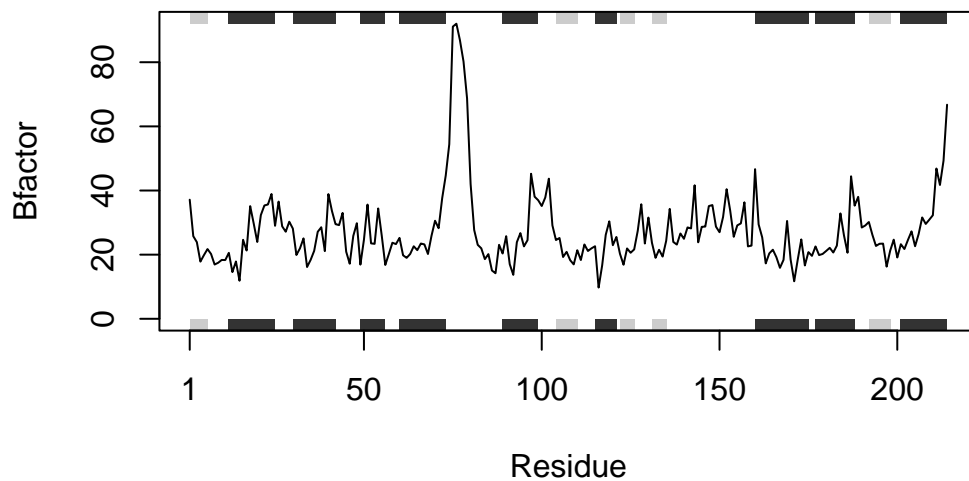
```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```
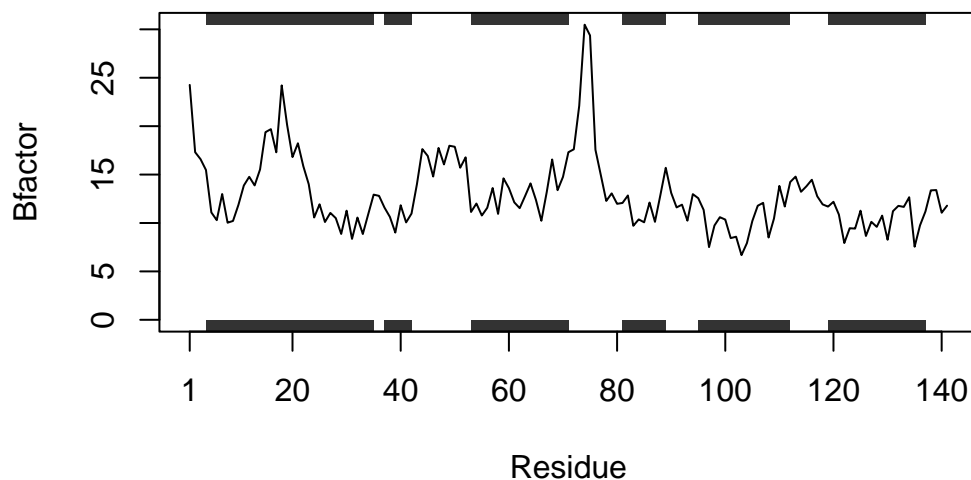

Testing for a protein:

Let's see if it works for the protein deoxy human hemoglobin (1A3N):

3

```
plot_proteinID("1A3N")
```

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


What is the input of the function?

Answer: the protein ID from the PDB database.

What does the function do?

Answer: The functions looks for the protein Id in the PDB database, looks for specific chains and atoms, and plots graph between bfactor and residues for the protein of interest.

What is the output of the function?

Answer: The out put is a graph between bfactor and residues for the protein of interest.

Note: I have also expanded on these questions in my annotations.