Class 6 Homework

Sarra Larif

1/23/2020

Improve this code: 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.chainAb s2.b <- s2.chainAb s3.b <- s3.chainAb 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”)

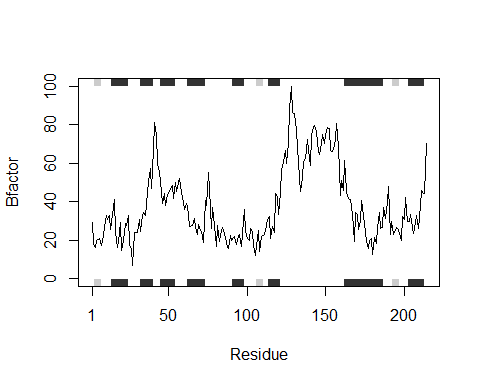
plot.prot <- function(PDB\_ID, chain = "A", atom = "CA", sse = b, typ = "l", ylab = "Bfactor"){  
 library(bio3d)  
 a <- read.pdb(PDB\_ID)   
 b <- trim.pdb(a, chain = chain, elety = atom)  
 c <- b$atom$b  
 plotb3(c, sse = sse, typ = typ, ylab = ylab)  
}

The inputs of this function include the PDB ID to identify the protein of interest, the chain which is an input for the trim.pdb which identifies the chain of interest in the protein, the atom type which is normally the optional elety input in the trim.pdb, and sse, typ, and ylab which specify the appearance of the plot as well as its y-axis label. All the inputs in my function except the protein ID have the same defaults as the original provided code but still allows the user to make changes, should they want to have different specifications for their protein or plot. Therefore, all one really needs to do to use this function is enter the PDB ID and optional parameters to receive an output of a scatter plot of the selected protein structure residues including secondary structure element representations in the margins. It does this by first using the given PDB ID to read its data with read.pdb, then it narrows the required data with trim.pdb with the optional inputs or defaults and then plots said data using plotb3.

Examples of output:

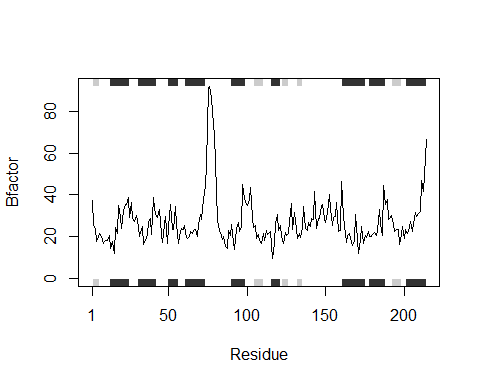
plot.prot("4AKE")

## Note: Accessing on-line PDB file



plot.prot("1AKE")

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



plot.prot("1E4Y")

## Note: Accessing on-line PDB file

