HW06

AUTHOR
Changcheng (PID: A69027828)

> Q1. What type of object is returned from the read.pdb() function?

It is a list

> Q2. What does the trim.pdb() function do?

It trims a PDB Object To A Subset of Atoms and produces a new smaller PDB object, containing a subset of atoms.

> Q3. What input parameter would turn off the marginal black and grey rectangles in the plots and what do they represent in this case?

#helix.col = FALSE, sheet.col = FALSE In this case, the black(gray20) rectangles represents alpha helices in the chainA of corresponding kinase, the gray(gray80) rectangles represents beta strands in the chainA of corresponding kinase .

> Q4. What would be a better plot to compare across the different proteins?

dendrogram plot.

> Q5. Which proteins are more similar to each other in their B-factor trends. How could you quantify this?

#hc <- hclust(dist(rbind(s1.b, s2.b, s3.b))) #plot(hc) “1AKE” and “1E4Y” are more similar to each other in their B-factor trends, which is shown in the Dendrogram. They share Height:250

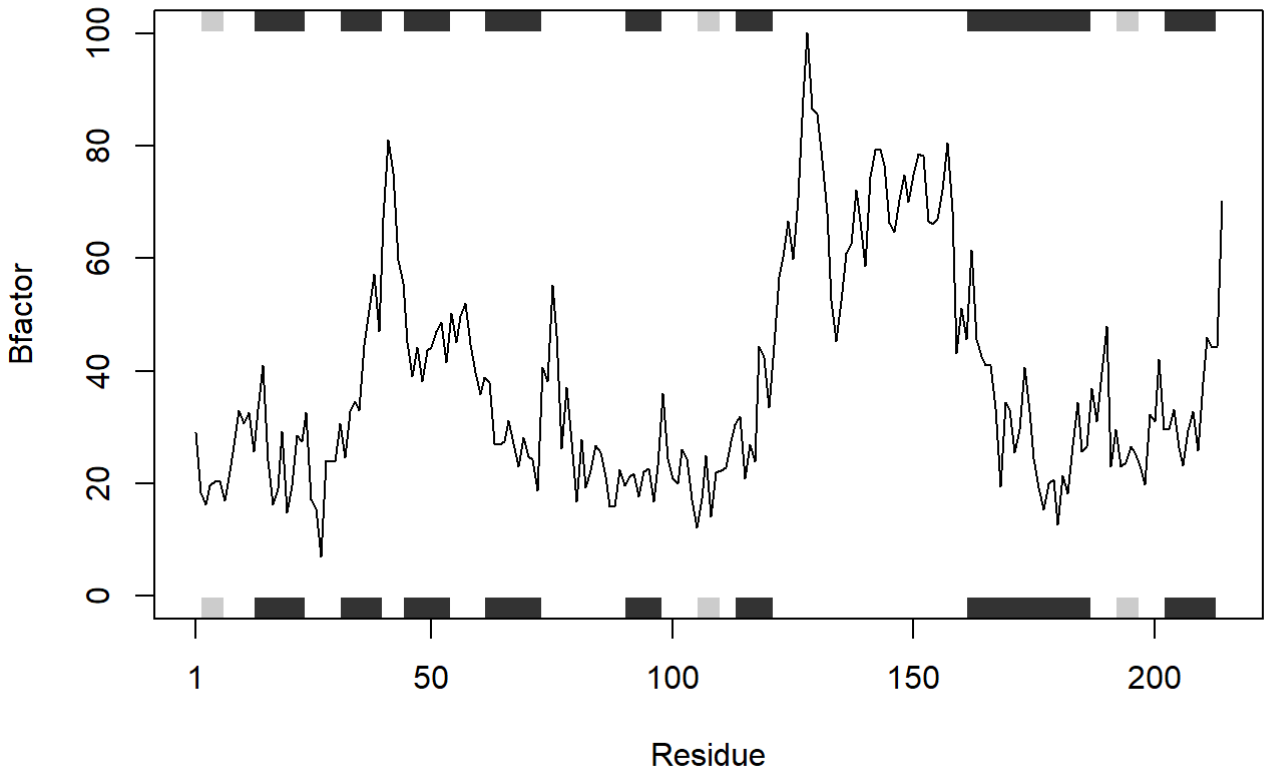
> Q6. How would you generalize the original code above to work with any set of input protein structures?

```
library(bio3d)

#PDB code to be analyzed are integrated to the 'vector_of_PDB_Code' vector
#Elements to be analyzed in input 'vector_of_PDB_Code' should be a (PDB_Code, Chain) vector
stru_analysis <- function(vector_of_PDB_Code){
  #Define a subfunction to analyze a single code
  analysis_of_one_Code <- function(Element_to_be_analyzed){
    s <- read.pdb(Element_to_be_analyzed[1])
    s.chain <- trim.pdb(s, chain= Element_to_be_analyzed[2], elety="CA")
    s.b <- s.chain$atom$b
    plotb3(s.b, sse=s.chain, typ="l", ylab="Bfactor")
  }
  #Analyze every PDB code in the vector_of_PDB_Code
```

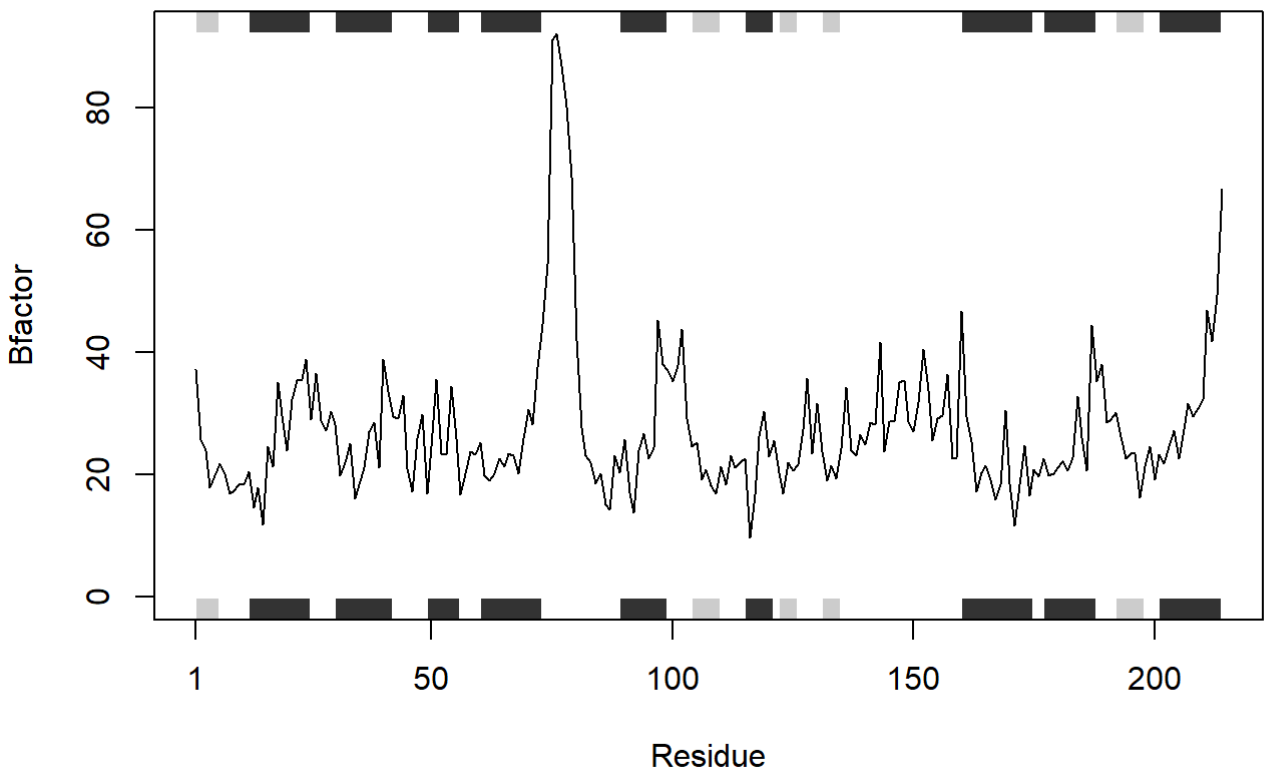
```
  apply(matrix(vector_of_PDB_Code,nrow=2), 2, analysis_of_one_Code)
  #Below is to obtain the dendrogram plot
  #Define a subfunction to obtain the B Factor
  B_Factor <- function(Element_to_be_analyzed){
    s.b <- trim.pdb(read.pdb(Element_to_be_analyzed[1]), chain= Element_to_be_analyzed[2], ele
  }
  transx <- apply(matrix(vector_of_PDB_Code,nrow=2), 2, B_Factor)
  #transx is also a matrix which is the transposition of rbind(s1.b, s2.b, s3.b, ..., sn.b)
  hc <- hclust( dist( t(transx) ) )
  #dendrogram plot
  plot(hc)
  #The branch of dendrogram plot is numbered from 1 to n corresponding to the sequence of inpu
}
```

```
#Test the performance of function
vector_of_PDB_Code <- c(c("4AKE","A"), c("1AKE","A"), c("1E4Y","A"))
stru_analysis(vector_of_PDB_Code)
```

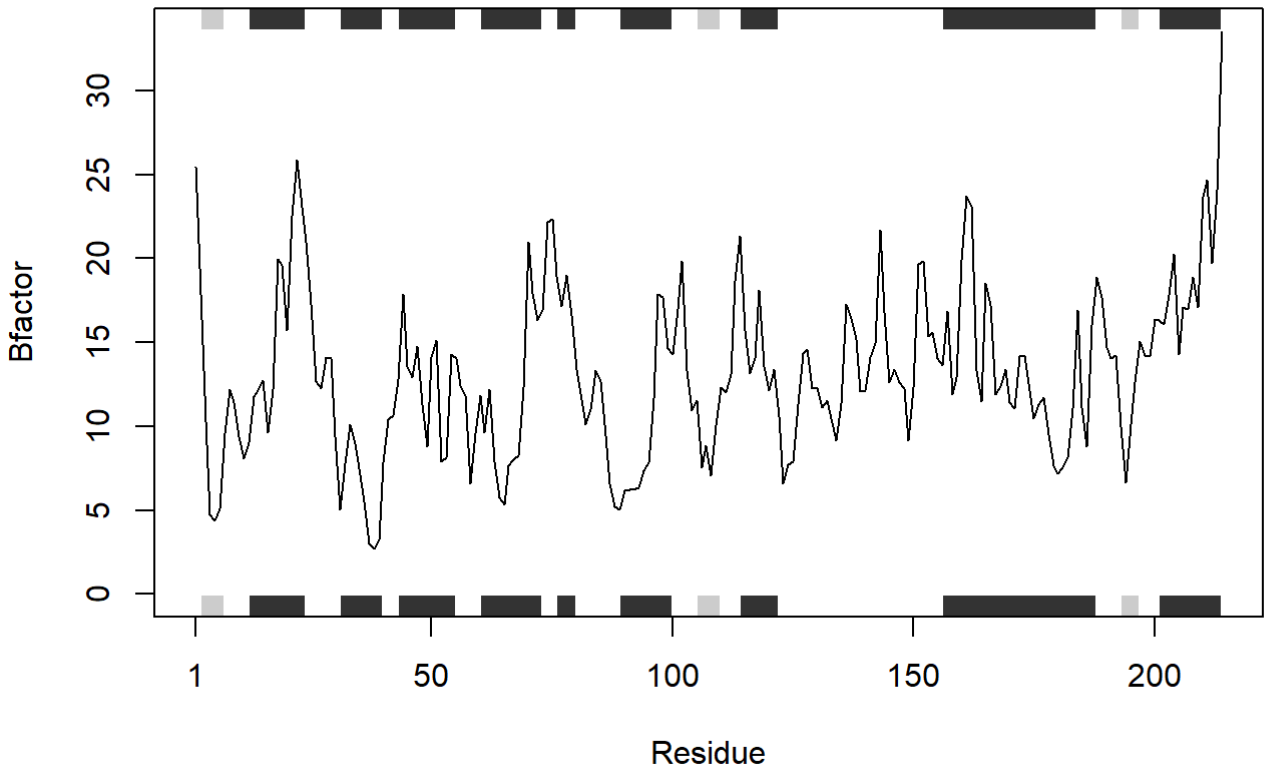
```
Note: Accessing on-line PDB file
```

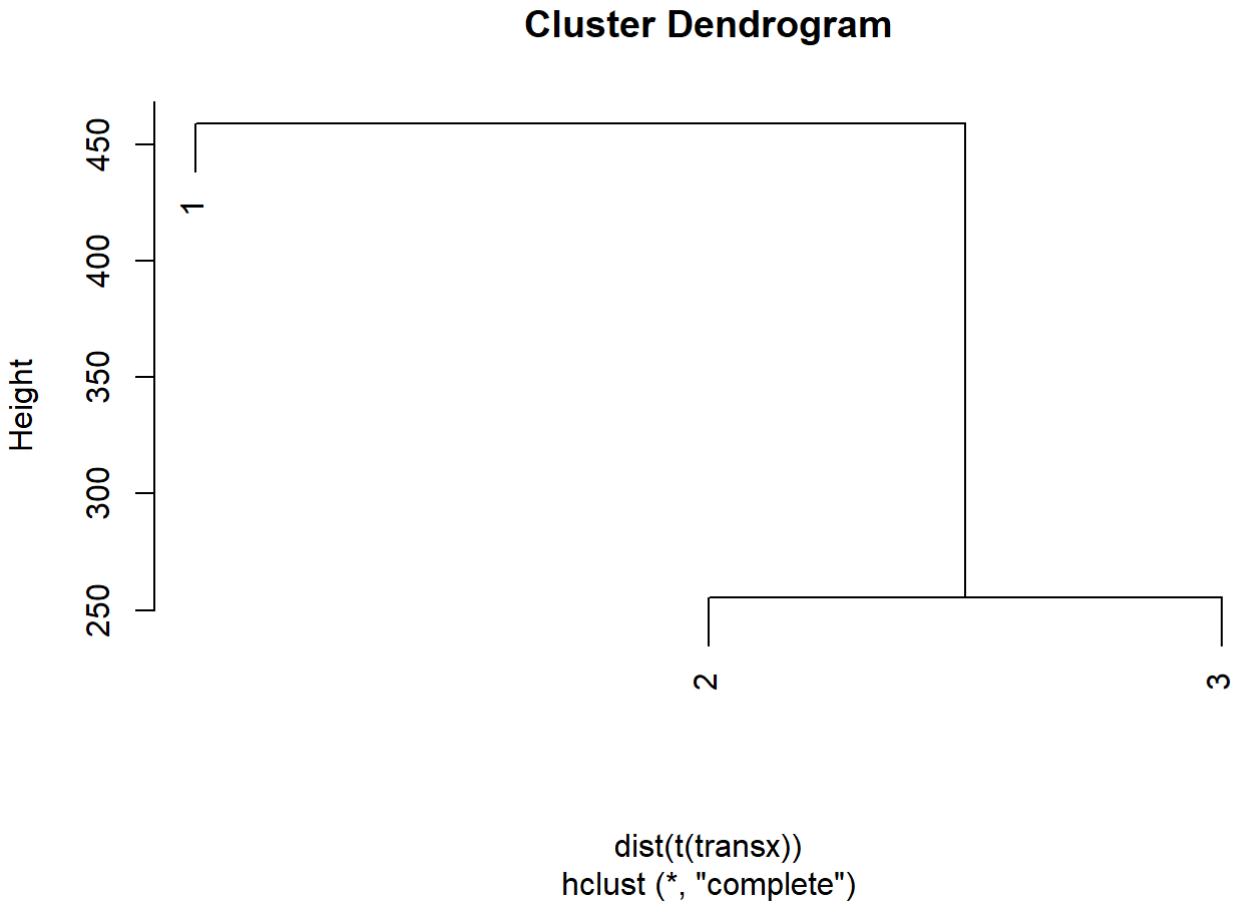



```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```




```
  Note: Accessing on-line PDB file

  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\30396\AppData\Local\Temp\Rtmpu4p6fy/4AKE.pdb exists. Skipping download

  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\30396\AppData\Local\Temp\Rtmpu4p6fy/1AKE.pdb exists. Skipping download

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

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\30396\AppData\Local\Temp\Rtmpu4p6fy/1E4Y.pdb exists. Skipping download
```


Bfactor
0
5
10
15
20
25
30
1
50
100
150
200
Residue


Cluster Dendrogram
Height
250
300
350
400
450
1
2
3
dist(t(transx))
hclust (*, "complete")