bggn213_HW_Class6

Jibin (PID: A53300326)

2021/10/21

Can you improve this analysis code?

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

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

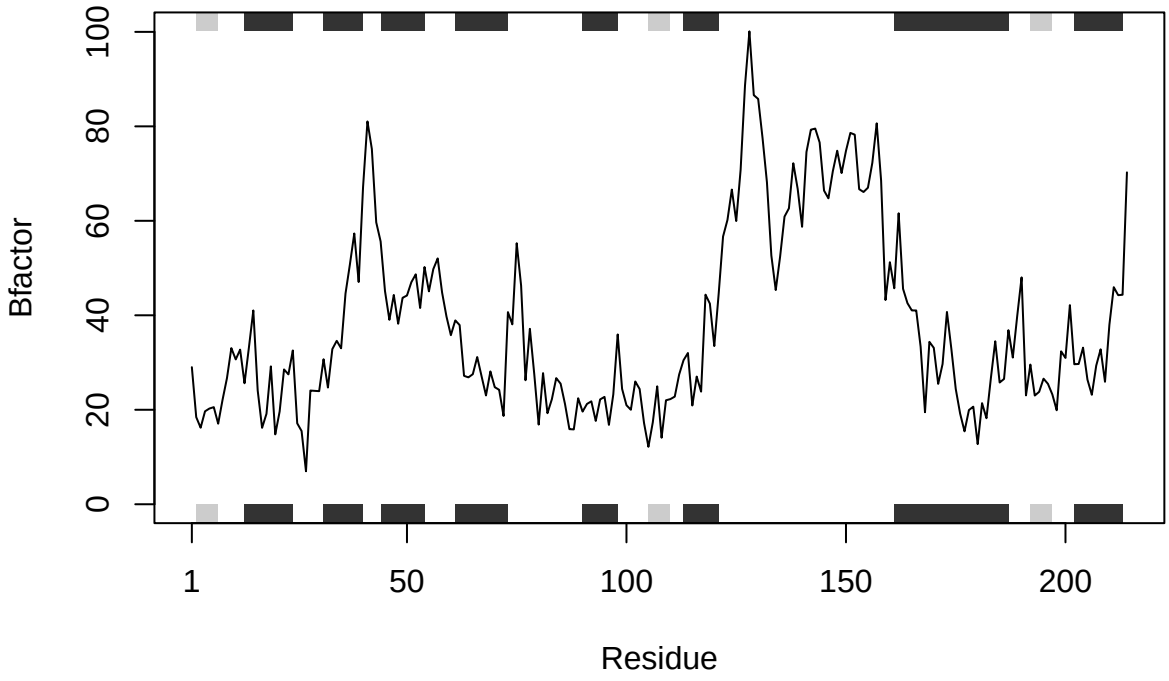
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

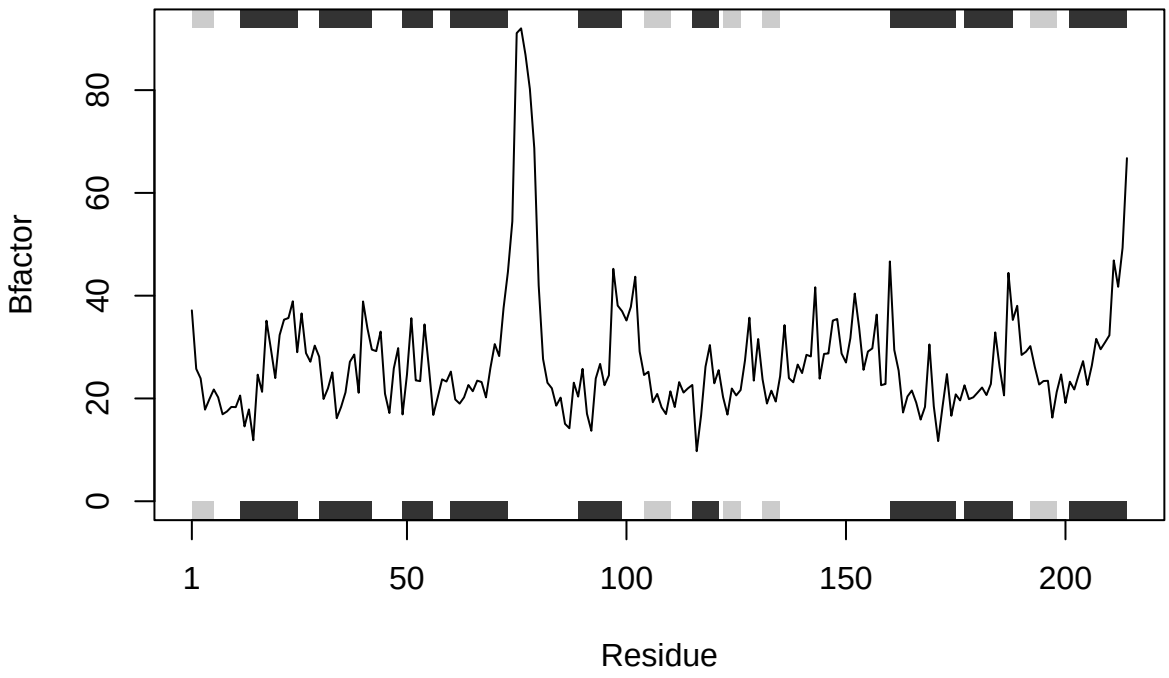
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

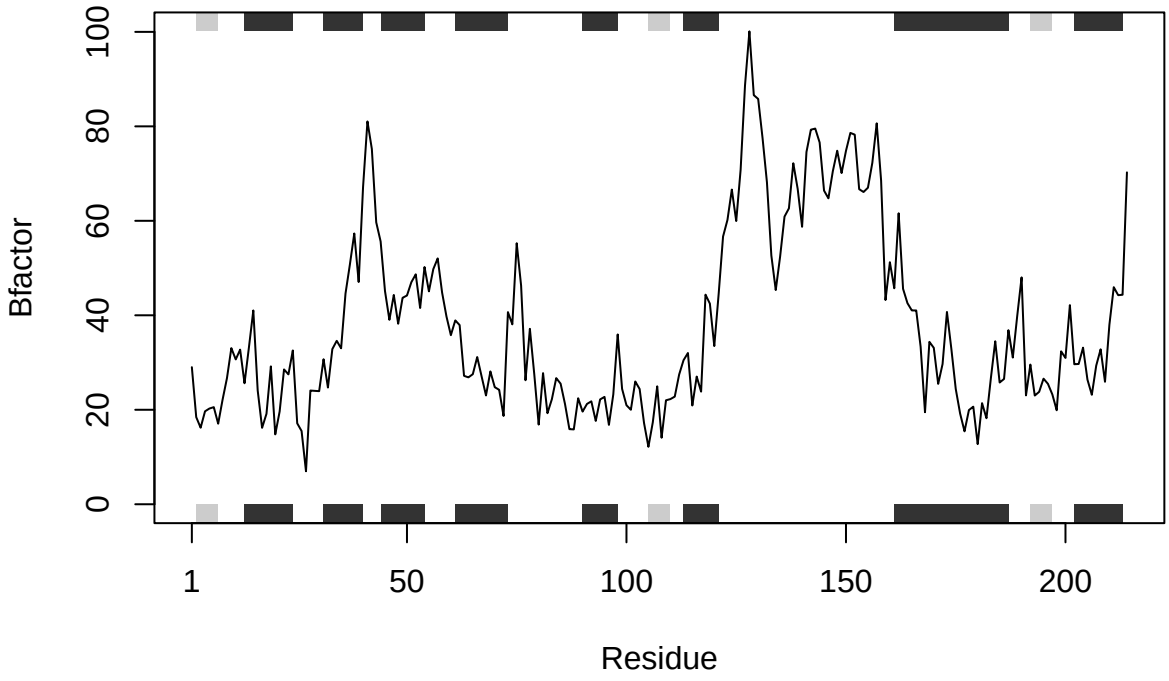
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



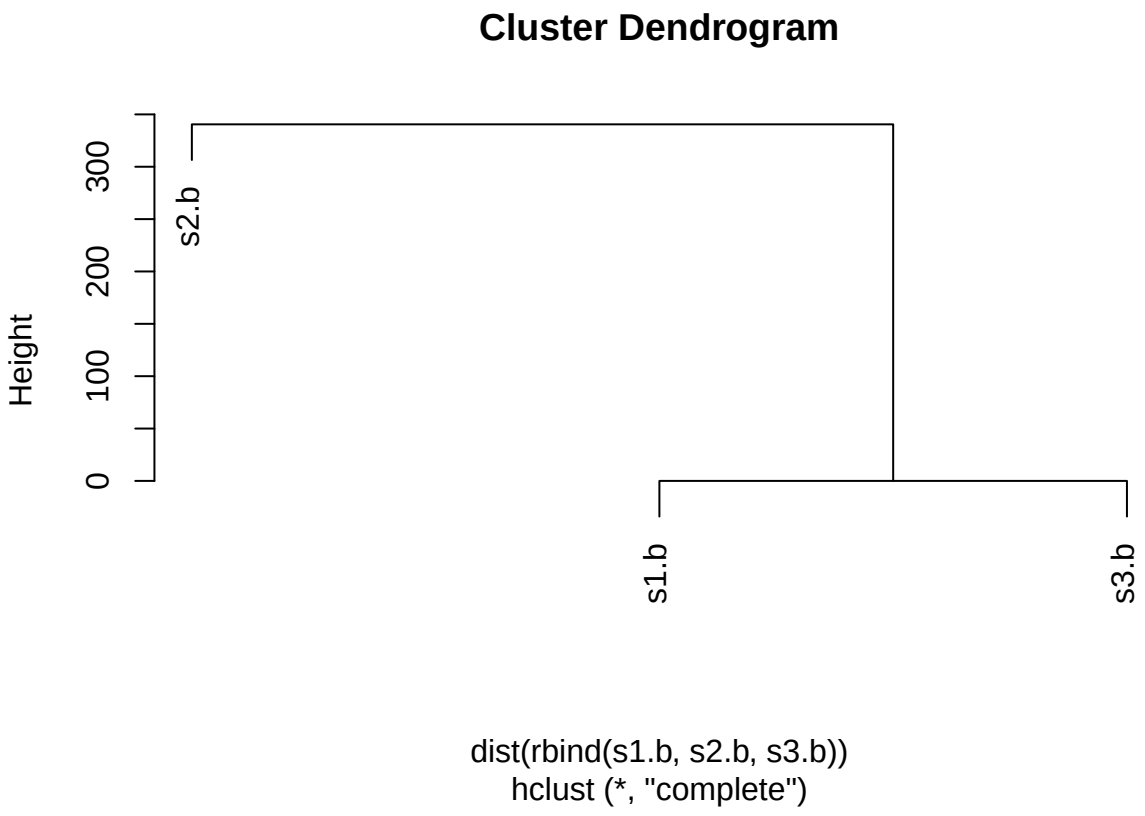

```
plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor")
```




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




```
hc <- hclust( dist( rbind(s1.b, s2.b, s3.b) ) )
plot(hc)
```



A better plot to compare across the different proteins

plotb3 function is useful for plotting per-residue numeric vectors for a given protein structure along with a schematic representation of major secondary structure elements.

trim.pdb () can be used to produce a new smaller PDB object, containing a subset of atoms, from a given larger PDB object.

atom is a character matrix containing all atomic coordinate ATOM data, with a row per ATOM and a column per record type. See below for details of the record type naming convention (useful for accessing columns).

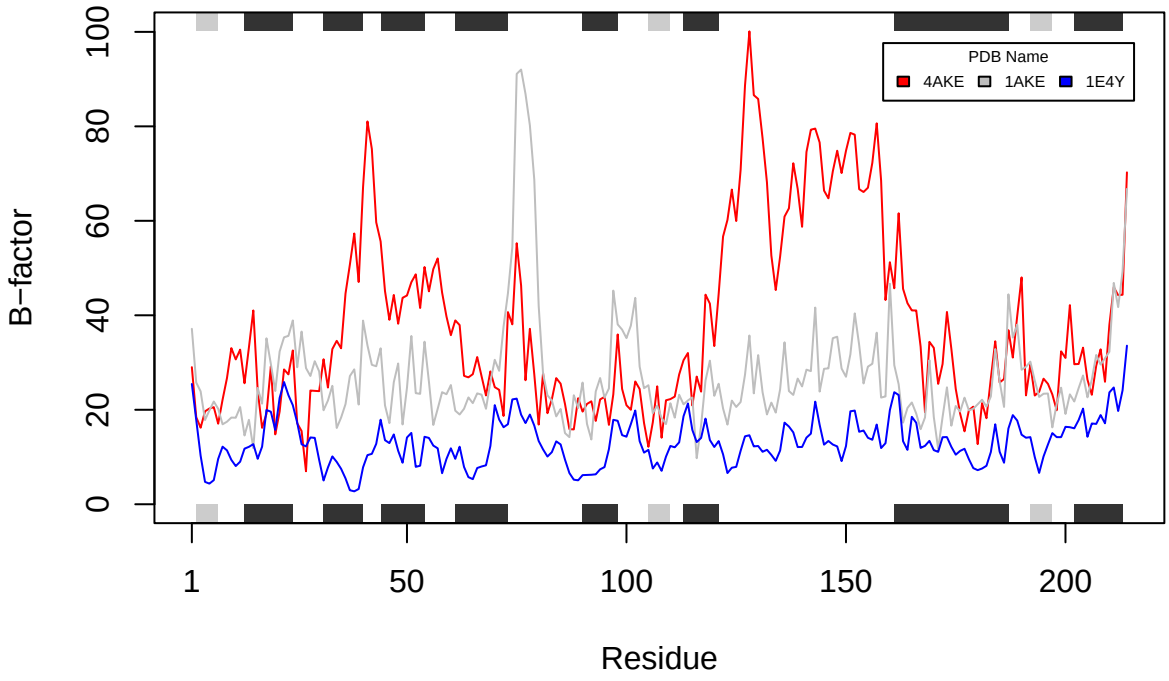
The B-factor describes the displacement of the atomic positions from an average (mean) value (mean-square displacement), which reflects the mobility or flexibility of various parts of the molecule.

```
library(bio3d)

Better_plotb3<- function(pdbname, chain, elety) {
  # assign different colors to each vector
  plot_colors <- c("red", "gray", "blue")
```

```
  # to read multiple PDB files and extract the "atom" & "b" information
  for (i in 1:length(pdbname)) {
  s1 <- read.pdb(pdbname[i])
  s1.chain_df <- trim.pdb(s1, chain = chain, elety = elety)
  s1.b<- s1.chain_df$atom$b

  # plot s1.b
  if (i == 1) {
    plotb3(s1.b, sse = s1.chain_df, typ = "l", ylab="B-factor",
           col = plot_colors[i])

  # adds additional lines to s1.b
  } else {
    lines(s1.b, col = plot_colors[i])
  }
  }

  # modify the legend
  legend("topright", title = "PDB Name", pdbname, fill = plot_colors,
         horiz=TRUE, cex = 0.5, inset = c(0.03, 0.06))
}
```

```
pdbname <- c("4AKE", "1AKE", "1E4Y")
chains <- "A"
elety <- "CA"

Better_plotb3(pdbname, chains, elety)
```

```
##   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
```

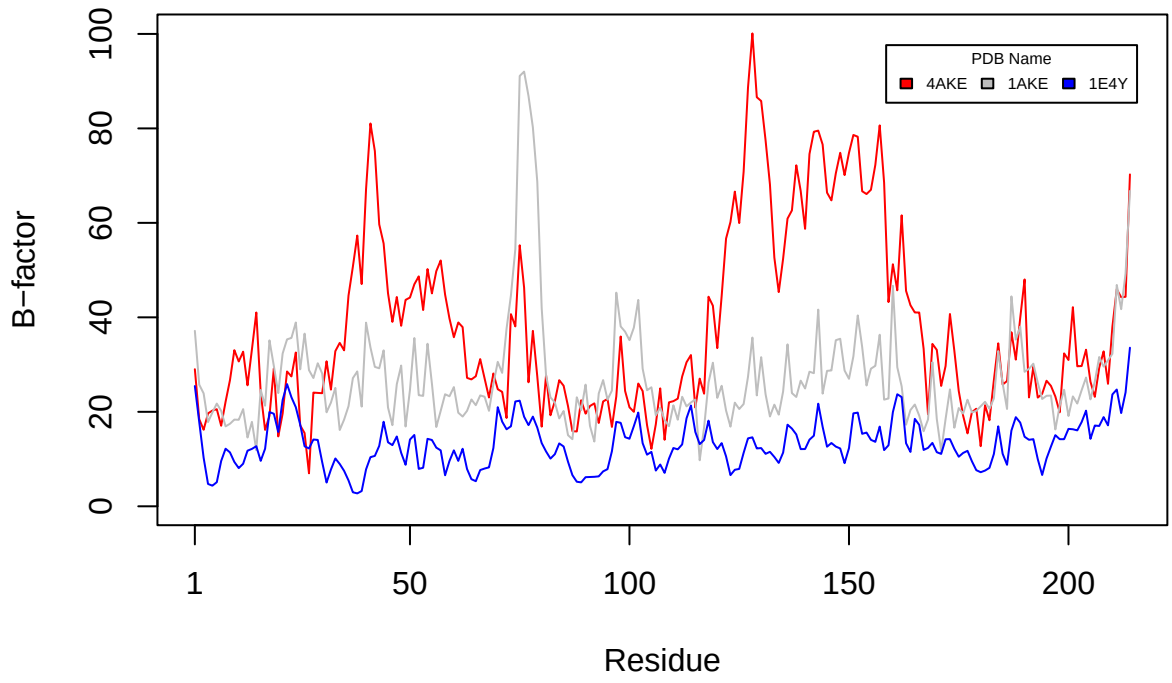
To remove the secondary structure from the plot

```
library(bio3d)

Better_plotb3_2<- function(pdbname, chain, elety) {
  # assign different colors to each vector
  plot_colors <- c("red", "gray", "blue")

  # to read multiple PDB files and extract the "atom" & "b" information
  for (i in 1:length(pdbname)) {
  s1 <- read.pdb(pdbname[i])
  s1.chain_df <- trim.pdb(s1, chain = chain, elety = elety)
  s1.b<- s1.chain_df$atom$b

  # plot s1.b
  if (i == 1) {
    plotb3(s1.b, typ = "l", ylab="B-factor", col = plot_colors[i])
```

```
  # adds additional lines to s1.b
  } else {
    lines(s1.b, col = plot_colors[i])
  }
  }

  # modify the legend
  legend("topright", title = "PDB Name", pdbname, fill = plot_colors,
         horiz=TRUE, cex = 0.5, inset = c(0.03, 0.06))
}
```

```
pdbname <- c("4AKE", "1AKE", "1E4Y")
chains <- "A"
elety <- "CA"

Better_plotb3_2(pdbname, chains, elety)
```

```
##   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
```