Class06 Homework

Emily Chen (PID:A16925878)

Can you improve this analysis code?

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

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

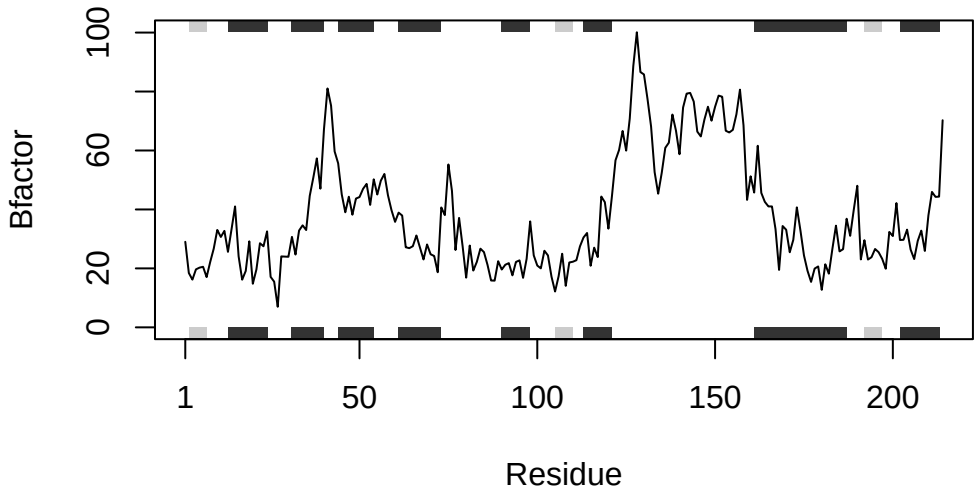
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

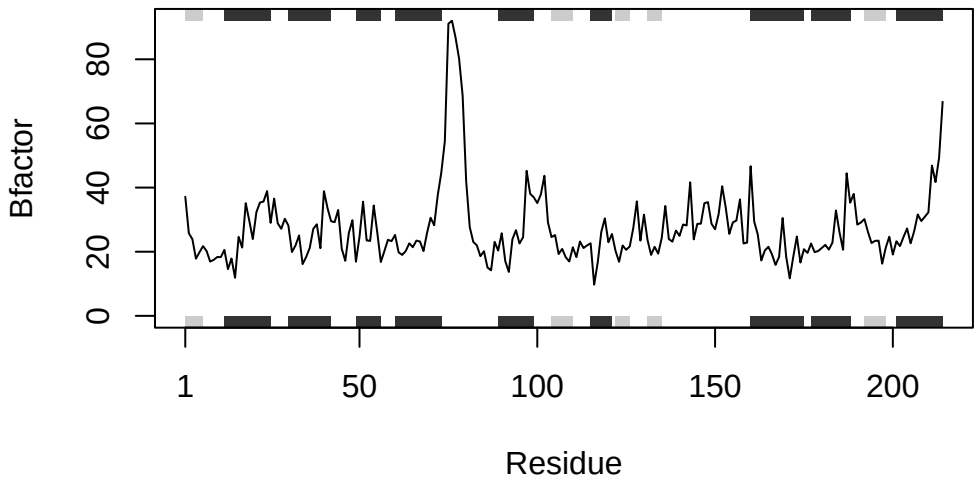
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

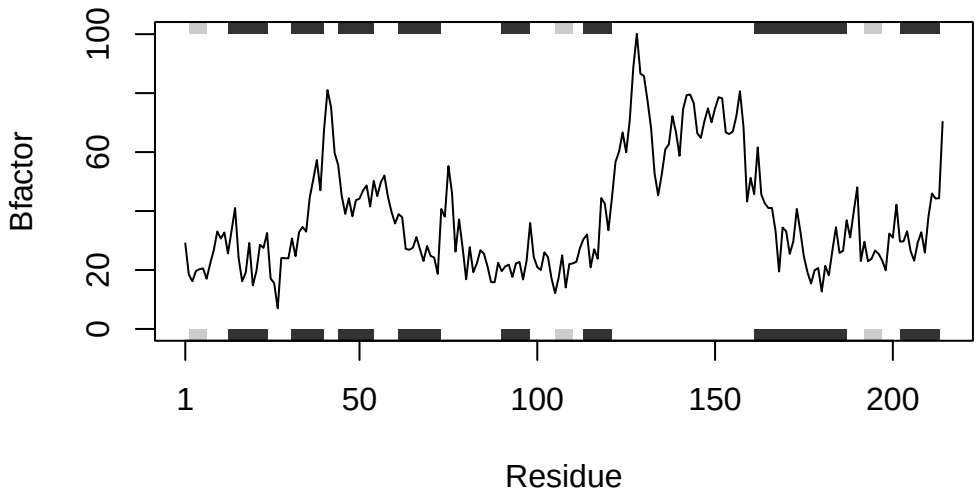
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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



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



Q2. What are the inputs to the function?

The protein names are the inputs to this functions

Q2. What the function does and how to use it.

The way in which the function works is that when you type in the code `s1`, it will tell R to read.pdb("4AKE") as that is what is assigned to the vector `s1`. This is also what R will do if we put in the code `s1. chainA`. R will give us information about this protein, which seems to be missing a certain chain.

```
s1
```

```
Call:  read.pdb(file = "4AKE")

  Total Models#: 1
    Total Atoms#: 3459,  XYZs#: 10377  Chains#: 2  (values: A B)

    Protein Atoms#: 3312  (residues/Calpha atoms#: 428)
    Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
```

```
    Non-protein/nucleic Atoms#: 147  (residues: 147)
    Non-protein/nucleic resid values: [ HOH (147) ]

  Protein sequence:
     MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
     DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
     VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
     YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILGMRIILLGAPGA...<cut>...KILG

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

```
s1.chainA
```

```
 Call:  trim.pdb(pdb = s1, chain = "A", elety = "CA")

  Total Models#: 1
    Total Atoms#: 214,  XYZs#: 642  Chains#: 1  (values: A)

    Protein Atoms#: 214  (residues/Calpha atoms#: 214)
    Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)

    Non-protein/nucleic Atoms#: 0  (residues: 0)
    Non-protein/nucleic resid values: [ none ]

  Protein sequence:
     MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
     DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
     VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
     YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

+ attr: atom, helix, sheet, seqres, xyz,
        calpha, call
```

Q3. What is the output of the function?

The output of the function is that it provided us with information about the protein such as the protein sequence along with the number of protein atoms, nucleic acids atoms, etc. In addition to that the function will also will also give us a graph that plots the residues on the x-axis and then the Bfactor on the y-axisthe plots that we get that plots the residues on the x-axis and then the Bfactor on the y-axis.

Condensing the Code.

1. look for repetitive code- read.pdb and trim.pdb(s2, chain="A", elety="CA") are both repetitive

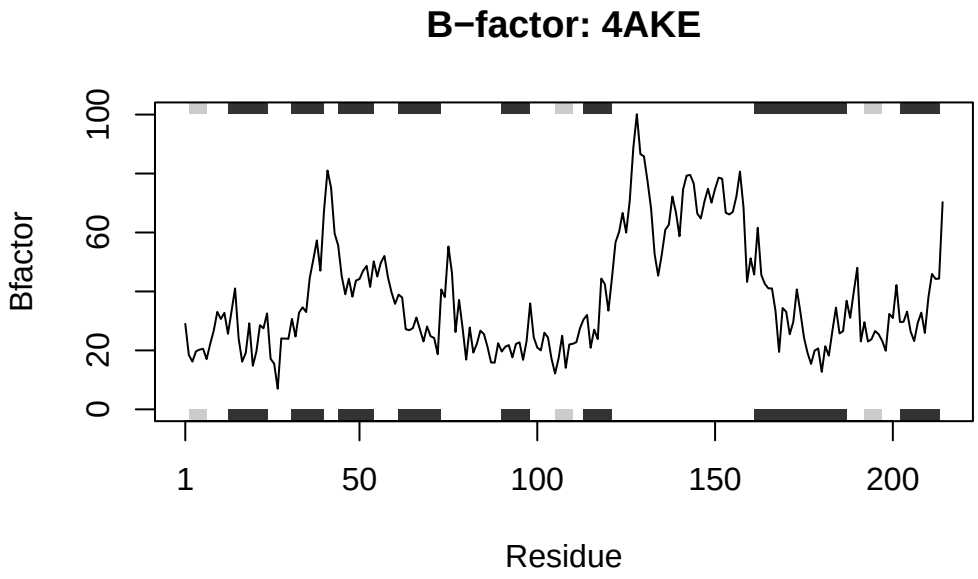
s1 <- read.pdb("4AKE") # kinase with drug s2 <- read.pdb("1AKE") # kinase no drug s3 <- read.pdb("1E4Y") # kinase with drug s1.chainA <- trim.pdb(s1, chain="A", elety="CA") s2.chainA <- trim.pdb(s2, chain="A", elety="CA") s3.chainA <- trim.pdb(s1, chain="A", elety="CA") s1.b <- s1.chainA$atom$b s2.b <- 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")

```
library(bio3d)
id<- c("4AKE", "1AKE", "1E4Y")
protein_bfactor <- function(pdb_ids) {
  for (id in pdb_ids) {
    pdb <- read.pdb(id)
    chainA <- trim.pdb(pdb, chain = "A", elety = "CA")  # Trim to chain A and CA atoms
    b <- chainA$atom$b
   plotb3(b, sse = chainA, typ = "l", ylab = "Bfactor", main = paste("B-factor:", id))
  }

}
```

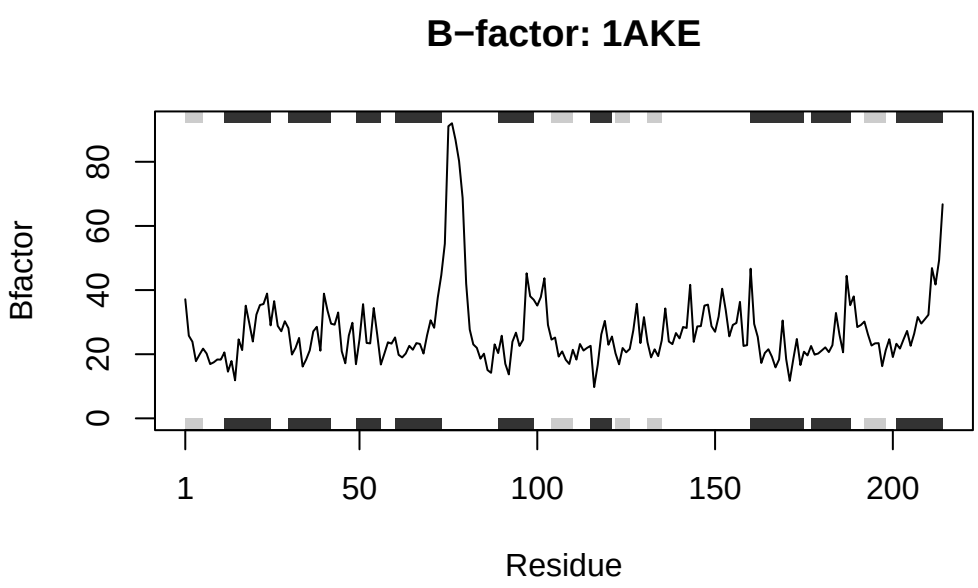
```
protein_bfactor(c("4AKE", "1AKE", "1E4Y"))
```

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

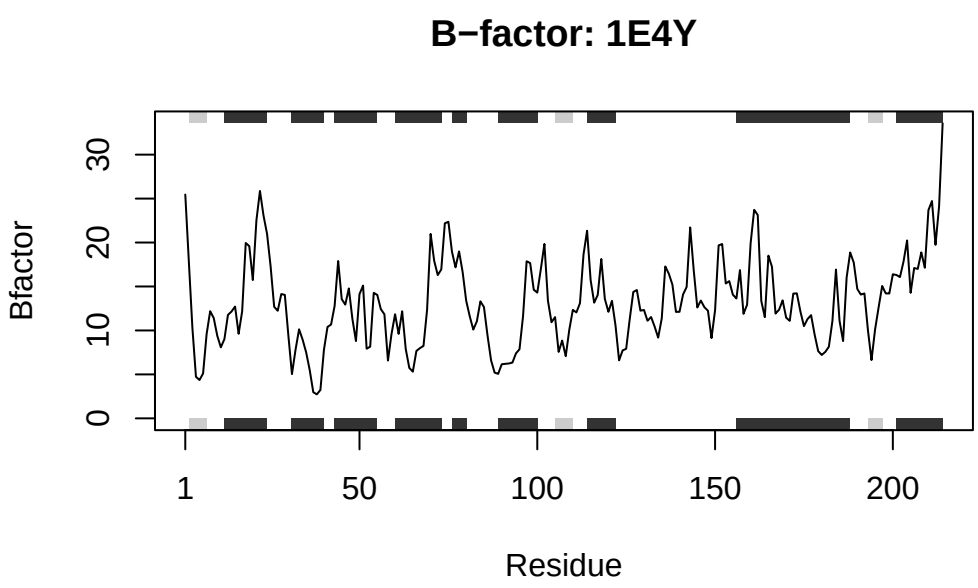
```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/41/0009hbvs0112_x1234tjnwzr0000gn/T//RtmpejExJN/4AKE.pdb exists.
Skipping download
```




```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/41/0009hbvs0112_x1234tjnwzr0000gn/T//RtmpejExJN/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):
/var/folders/41/0009hbvs0112_x1234tjnwzr0000gn/T//RtmpejExJN/1E4Y.pdb exists.
Skipping download
```


B−factor: 1E4Y
Bfactor
0
10
20
30
1
50
100
150
200
Residue