Class 06 HW

Joshua Mac

2025-01-27

Analysis code for drug interactions

```
# Can you improve this analysis code?
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

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

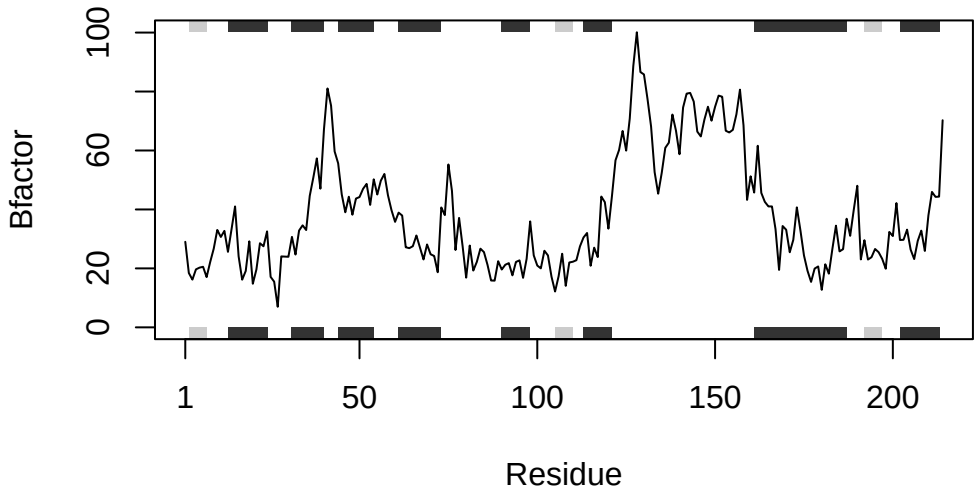
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

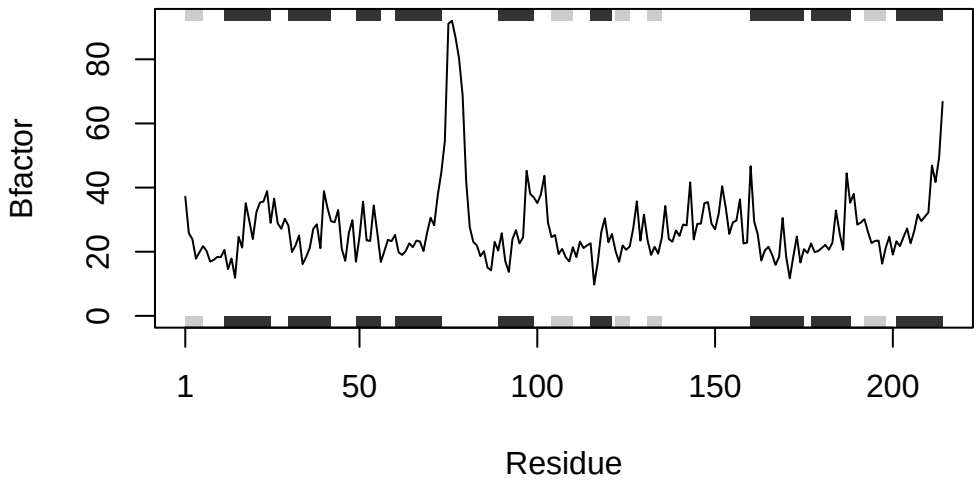
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

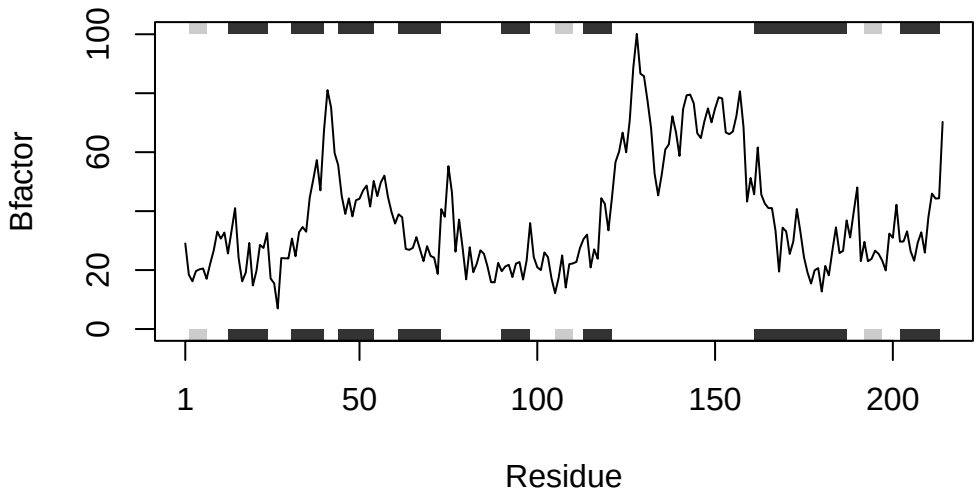
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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



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



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

`libary(bio3d)` is already on

Remember, function has name, input, and body in this structure:

function_name<-function(input){function_body}

```
#I will coin you: "protein_analysis", with "protein"
#as a holder of a pdb

protein_analysis<-function(protein, chain="A", elety="CA"){
  #Everything from now on will be part of the function
  #First, we need to read the inputted pdb
  p<-read.pdb(protein)
  #Next, trim the pdb to extract chain A data and alpha-carbon CA atoms only
  p.chainA<-trim.pdb(p, chain="A",elety="CA")
  #Next, extract the Bfactors, which is the atom column and the b column
  p.bfactors<-p.chainA$atom$b
  #Finally, plot like in the original code
```

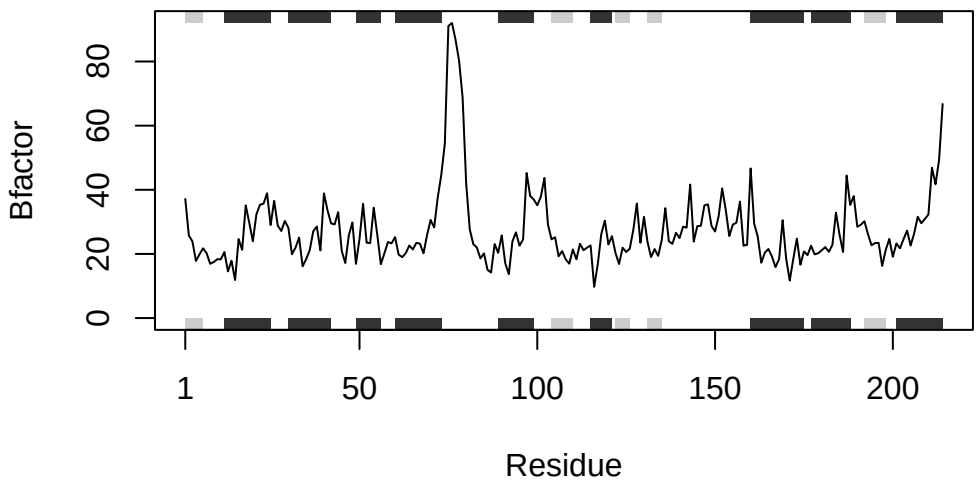
```
  plotb3(p.bfactors,sse=p.chainA,typ="l", ylab="Bfactor")
}
```

```
#now i will try and call/execute
protein_analysis("1AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/l3/0w1v1h157qj81sz_k6f27glm0000gn/T//RtmppdXqhu/1AKE.pdb exists.
Skipping download
```

```
   PDB has ALT records, taking A only, rm.alt=TRUE
```



Essentially, the function works by taking a pdb input, reading it, trimming it to create a new pdb that has only the chain information and keeps the chain element type, extracts the bfactors, and plots it.

The output is a plot that visualizes Bfactor x Residue !

For example, above, I called the function protein_analysis on the pdb “1AKE” and it plot its Bfactor x Residue relationship for me in a line type bio3d plot.