Class 06: Improving analysis code by writing functions

Selma Cifric (PID A69042976)

Quarto

> Q6. Can you improve this analysis 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.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")

First, I check for semantic errors in the code. I think the `trim.pdb()` function on s3.chain object is trimming the wrong data points. It should trim s3 instead. First fix creates, still redundant, code as follows:

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

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

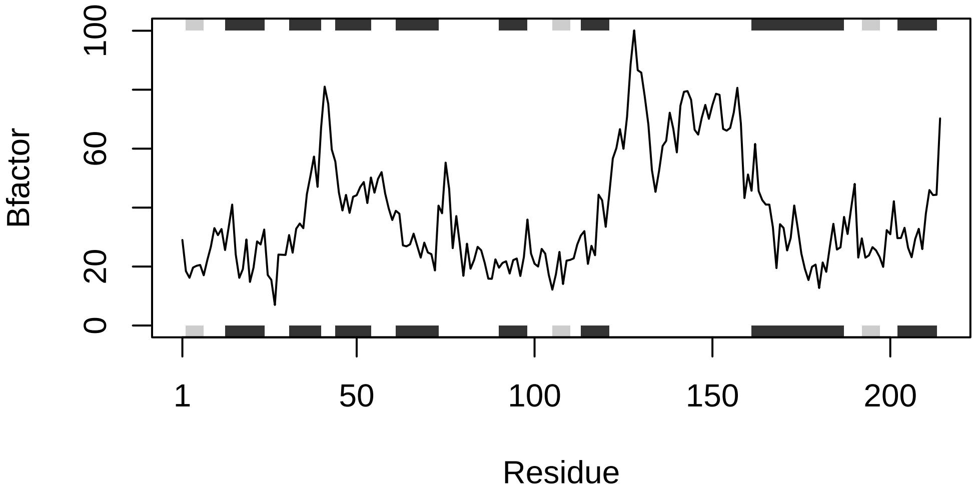
```r
s2 <- read.pdb("1AKE")  # kinase no drug
```

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

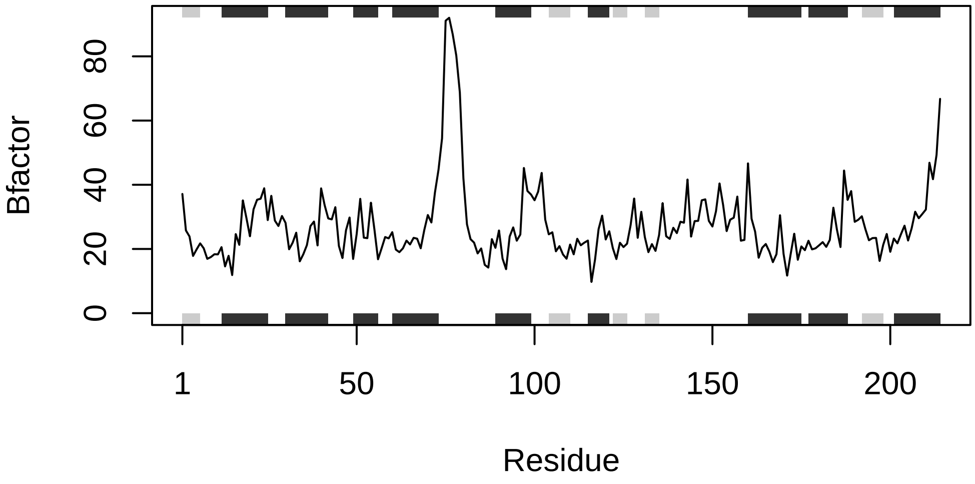
```
s3 <- read.pdb("1E4Y")  # kinase with drug
```

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

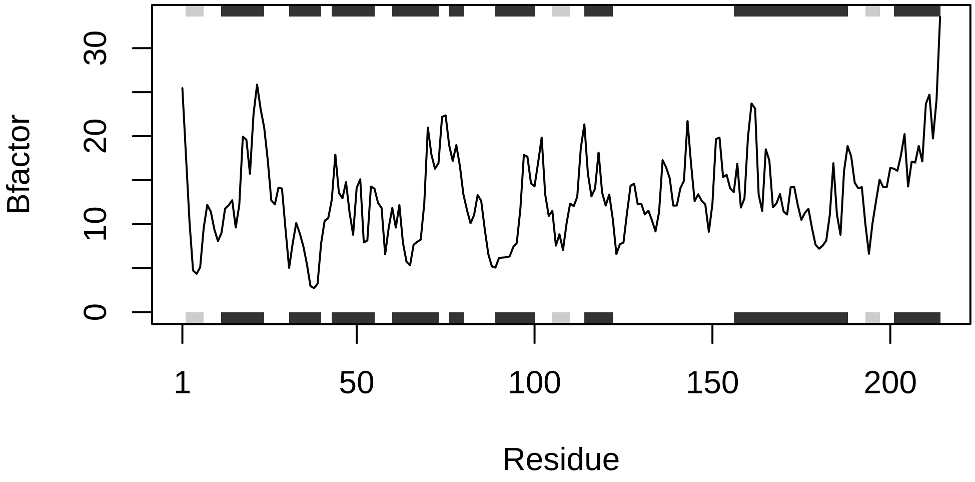
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

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")
```



Then, instead of reading each PDB file as separate, we can combine them all into a vector carrying list so we can easily call on them together. Then, we can assign names to each file in the list.

```r
pdb_files <- c("4AKE", "1AKE", "1E4Y")
names(pdb_files) <- c("Kinase_drug_1", "Kinase_no_drug", "Kinase_drug_2")
```

It seems that we want to: first, trimm lists so we only keep alpha carbons, and second, obtain b-factors for each trimmed structure.

To keep only alpha carbons on chainA:

```
pdb <- read.pdb(pdb_files)
```

```
chainA <- trim.pdb(pdb, chain="A", elety="CA")
```

Then, we take out b factors from trimmed alpha C in chain A. I assigned this to an object b, then asked for a list where B factors are matched to their corresponding position in the chain.

```
b <- chainA$atom$b
```

```
list(pdb=chainA, b=b)
```

All of these elements are to be applied to each drug condition, therefore we should make a function that I called here `bfactor` which is a function of pdb_files that I named in the beginning.

```r
bfactor <- function(pdb_file) {
  pdb <- read.pdb(pdb_file)
  chainA <- trim.pdb(pdb, chain="A", elety="CA")
  b <- chainA$atom$b
  list(pdb=chainA, b=b)
}
```

Now we have a function that can be applied to all the files in pdb_files.

```r
help(lapply)
pdb_final <- lapply(pdb_files, bfactor)
```

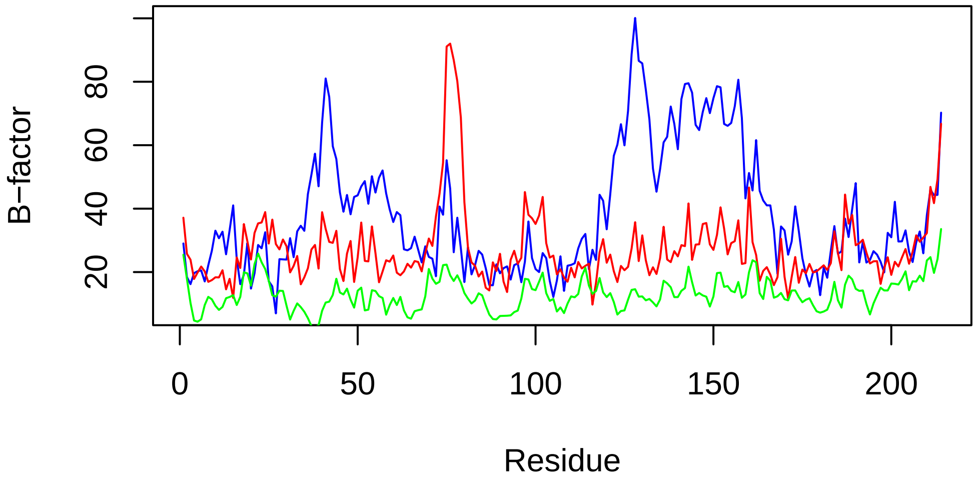
```
  Note: Accessing on-line PDB file
```

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/6q/r4lflzx92xn87c5yb_1q5yp00000gn/T//RtmpnqhR1l/4AKE.pdb exists.
Skipping download
```

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/6q/r4lflzx92xn87c5yb_1q5yp00000gn/T//RtmpnqhR1l/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/6q/r4lflzx92xn87c5yb_1q5yp00000gn/T//RtmpnqhR1l/1E4Y.pdb exists.
Skipping download
```

And we plot this data. I'm extracting enzyme 1, 2, 3 and its b factor (trimmed for alpha cardbons only).

```
plot(pdb_final[[1]]$b, type="l", col="blue", ylab="B-factor", xlab="Residue")
lines(pdb_final[[2]]$b, col="red")
lines(pdb_final[[3]]$b, col="green")
```

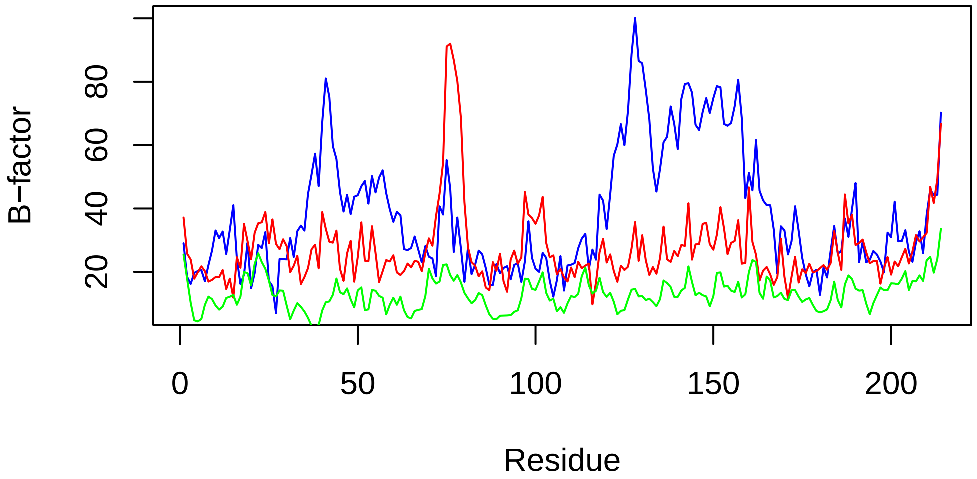


Final code:

```r
pdb_files <- c("4AKE", "1AKE", "1E4Y")
names(pdb_files) <- c("Kinase_drug_1", "Kinase_no_drug", "Kinase_drug_2")


bfactor <- function(pdb_file) {
  pdb <- read.pdb(pdb_file)
  chainA <- trim.pdb(pdb, chain="A", elety="CA")
  b <- chainA$atom$b
  list(pdb=chainA, b=b)
}

pdb_final <- lapply(pdb_files, bfactor)
```

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/6q/r4lflzx92xn87c5yb_1q5yp00000gn/T//RtmpnqhR1l/4AKE.pdb exists.
Skipping download

  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/6q/r4lflzx92xn87c5yb_1q5yp00000gn/T//RtmpnqhR1l/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/6q/r4lflzx92xn87c5yb_1q5yp00000gn/T//RtmpnqhR1l/1E4Y.pdb exists.
Skipping download
```

```r
plot(pdb_final[[1]]$b, type="l", col="blue", ylab="B-factor", xlab="Residue")
lines(pdb_final[[2]]$b, col="red")
lines(pdb_final[[3]]$b, col="green")
```


B−factor
20
40
60
80
0
50
100
150
200
Residue