Class 6 Function Homework

Merna Behnam (PID: A17675170)

Improving Analysis Code

Starting Code

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

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

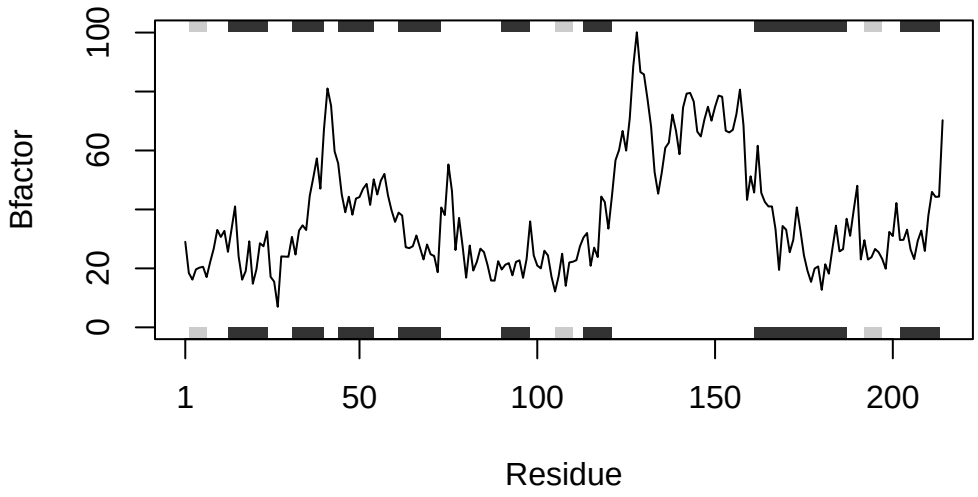
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

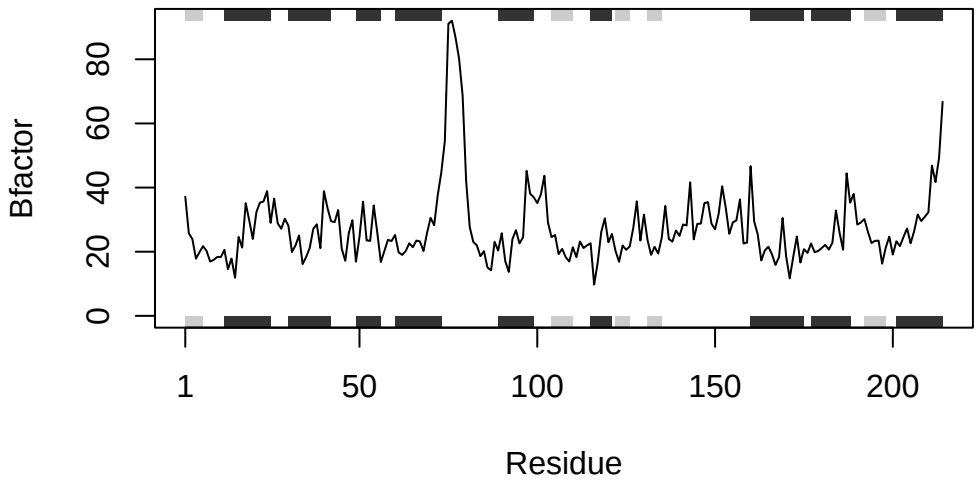
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

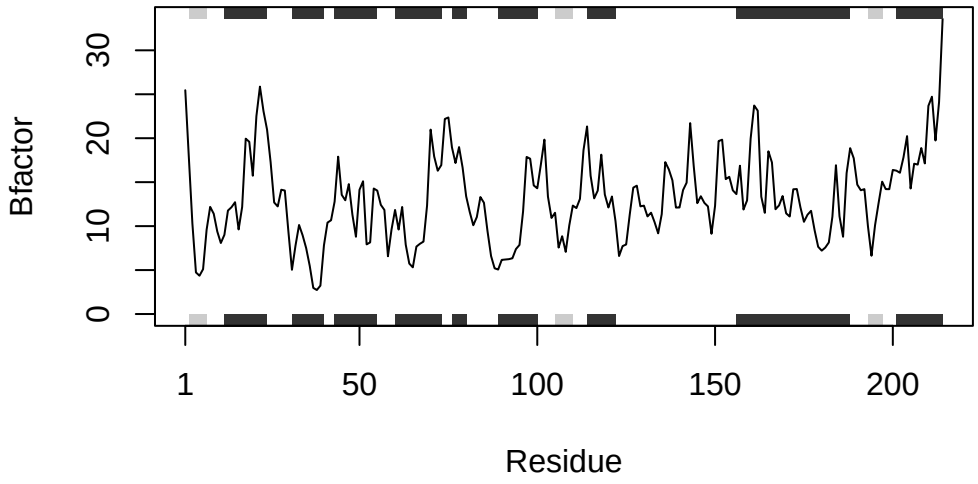
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s3, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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



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



Improving Code

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

Function name: `analyze_protein_bfactors`

What you give this function:

- **pdb_ids:** a vector of PDB IDs you want to analyze (for example: `c("4AKE", "1AKE", "1E4Y")`)
- **chain:** which protein chain to look at (default is "A")
- **elety:** what atom type to use (default is "CA" for alpha carbons)

What this function does:

- Reads in each PDB structure
- Keeps only the selected chain and atom type

- Pulls out the B-factor values
- Makes a B-factor plot for each protein

What you get back:

- A named list of B-factor values (one per protein)
- Plus the plots showing B-factor trends

How to use it:

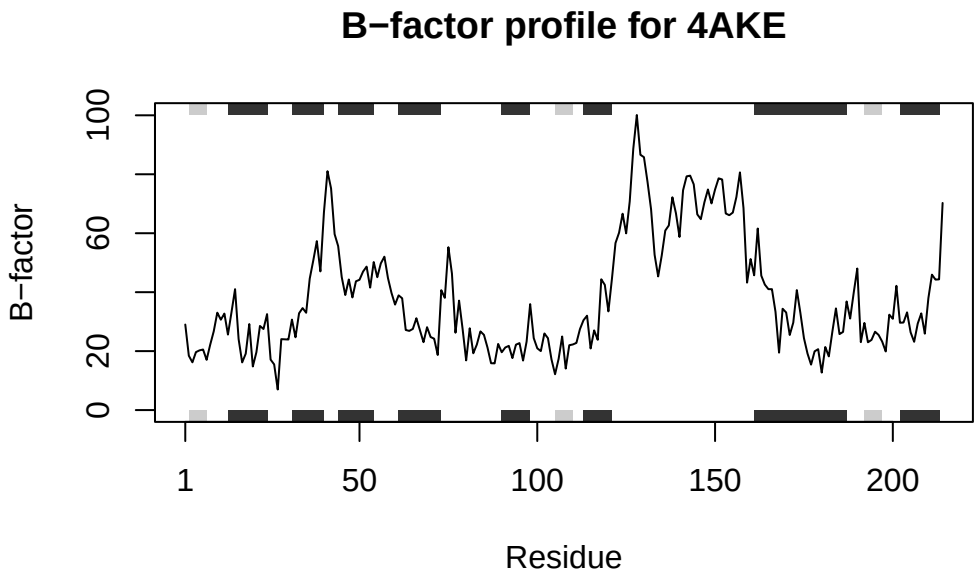
```
analyze_protein_bfactors(c("4AKE", "1AKE", "1E4Y"))
```

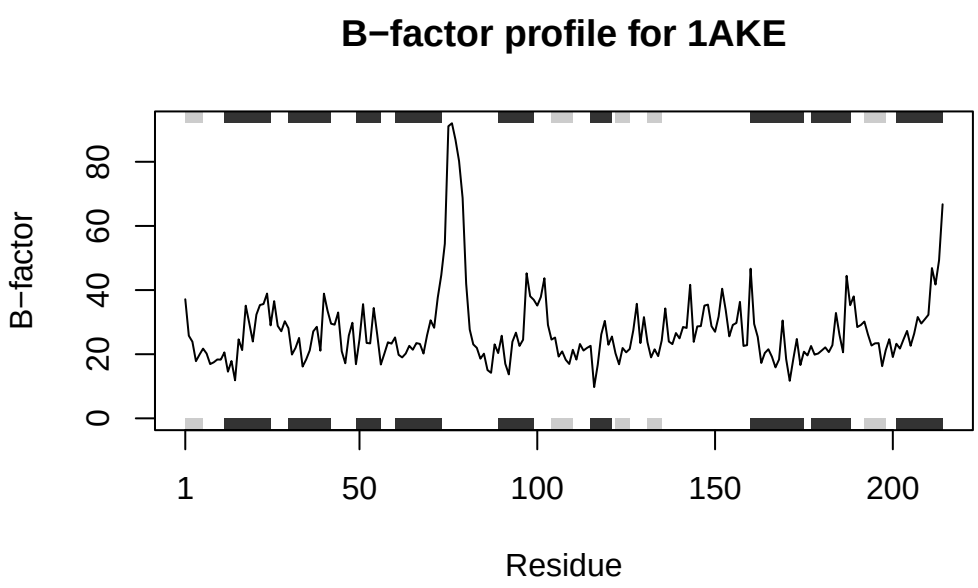
```r
analyze_protein_bfactors <- function(pdb_ids, chain = "A", elety = "CA") {

  # Step 1: read in all the PDB files
  pdbs <- lapply(pdb_ids, read.pdb)

  # Step 2: trim each structure to just chain A and CA atoms
  trimmed <- lapply(pdbs, trim.pdb, chain = chain, elety = elety)

  # Step 3: grab the B-factor values from each structure
  b_factors <- lapply(trimmed, function(x) x$atom$b)

  # Step 4: loop through and plot the B-factors for each protein
  for (i in seq_along(b_factors)) {
    plotb3(
      b_factors[[i]],
      sse  = trimmed[[i]],
      typ  = "l",
      ylab = "B-factor",
      main = paste("B-factor profile for", pdb_ids[i])
    )
  }
  # Name the list so it's easy to tell which protein is which
  names(b_factors) <- pdb_ids

  # Send the B-factor data back in case we want to use it later
  return(b_factors)
}
```

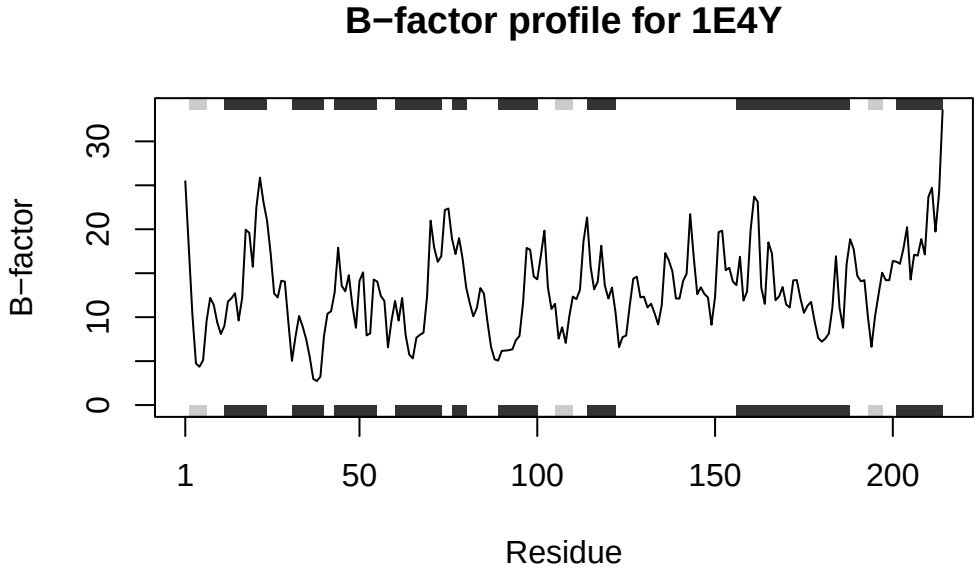
Here's an example run to show that the function actually works:

```
# Run the function on a few kinase structures
bfactors <- analyze_protein_bfactors(c("4AKE", "1AKE", "1E4Y"))
```

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/vm/8gc0hv7132d95mkx4t0xvscm0000gn/T//RtmpPD48a1/4AKE.pdb exists.
Skipping download

  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/vm/8gc0hv7132d95mkx4t0xvscm0000gn/T//RtmpPD48a1/1AKE.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/vm/8gc0hv7132d95mkx4t0xvscm0000gn/T//RtmpPD48a1/1E4Y.pdb exists.
Skipping download
```




B−factor profile for 1AKE
B−factor
0
20
40
60
80
1
50
100
150
200
Residue


B−factor profile for 1E4Y
B−factor
0
10
20
30
1
50
100
150
200
Residue

Ta-da!

This function generalizes the original analysis code so it works with any number of protein structures. Instead of repeating the same steps for each PDB file, the function loops through a vector of PDB IDs, extracts B-factors in a consistent way, and generates comparable plots automatically.