Homework Question 6

Carolina Merino (PID 14484883)

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

Description

We define a reusable R function that reads a protein PDB structure, extracts the alpha carbons from chain A, and plots the B-factor values. Using this function avoids repeating the same steps for each protein and allows easy comparison of protein flexibility with or without drug binding.

Originally, each protein required multiple lines of code to read, trim, extract B-factors, and plot. Writing a function consolidates these steps into a single reusable command, reducing code duplication, improving readability, and making the analysis more efficient.

Original analysis code:

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

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

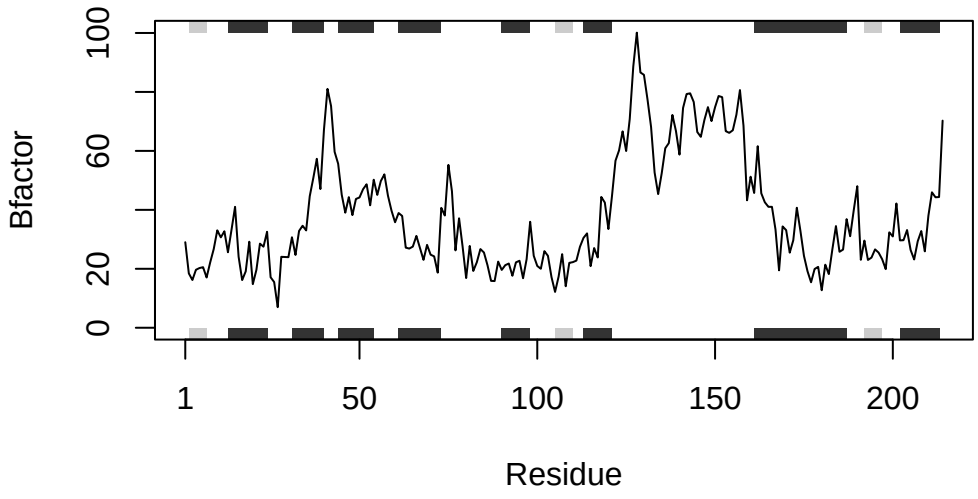
```
s2 <- read.pdb("1AKE") # kinase no drug
```

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

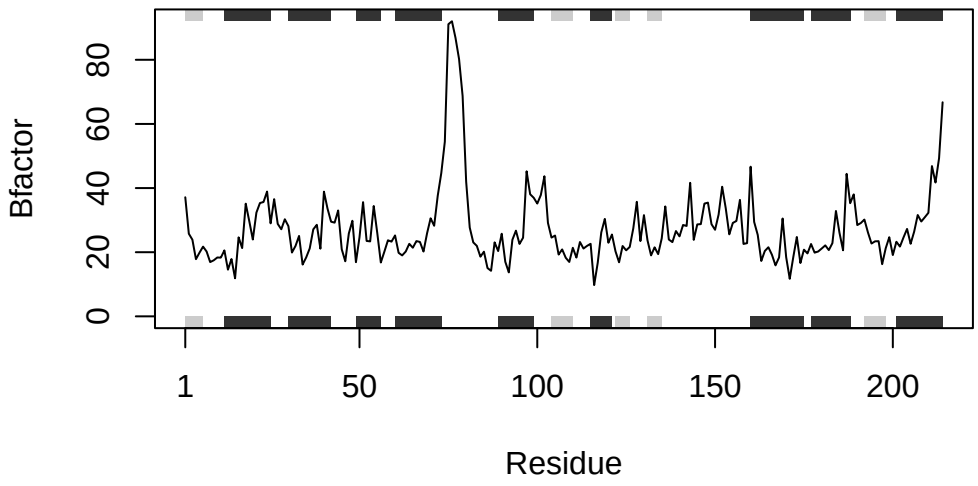
```
s3 <- read.pdb("1E4Y") # kinase with drug
```

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

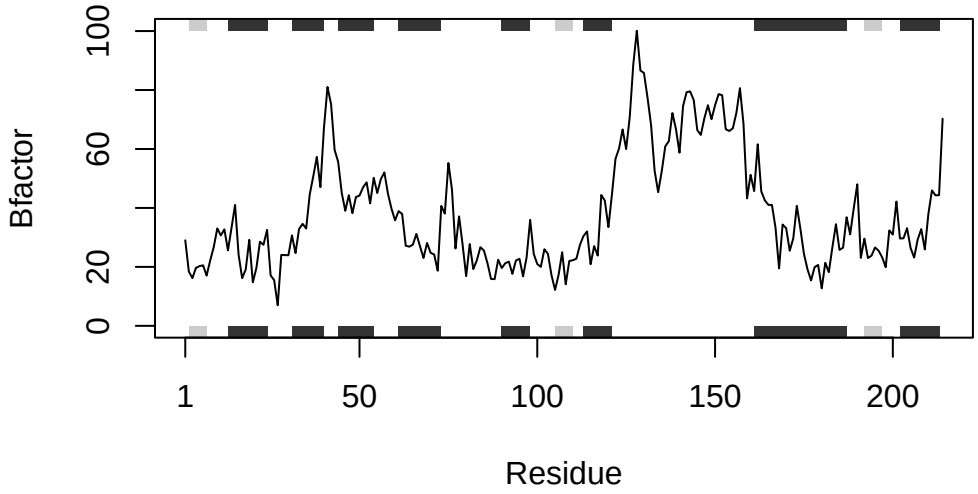
```
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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




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



Let's download the library first

This chunk loads the bio3d package, which contains functions for working with protein structures (PDB files) such as read.pdb(), trim.pdb(), and plotb3().

```
library(bio3d)
```

This function takes a protein's PDB ID (for example, "4AKE") and:

- Downloads and reads its 3D structure.
- Extracts only the alpha-carbon (CA) atoms from chain A.
- Plots the B-factor (a measure of atomic flexibility) along the amino acid sequence.

```
# Function to plot B-factors for a protein
plot_protein_bfactor <- function(id, chain="A") {
  pdb <- read.pdb(id)                  # read protein structure
  pdb_chain <- trim.pdb(pdb, chain=chain, elety="CA")  # trim to chain A alpha carbons
  plotb3(pdb_chain$atom$b, sse=pdb_chain, typ="l",
         ylab="B-factor", main=id)     # plot B-factors
}
```

Here we can call the function for several example proteins to generate B-factor plots.

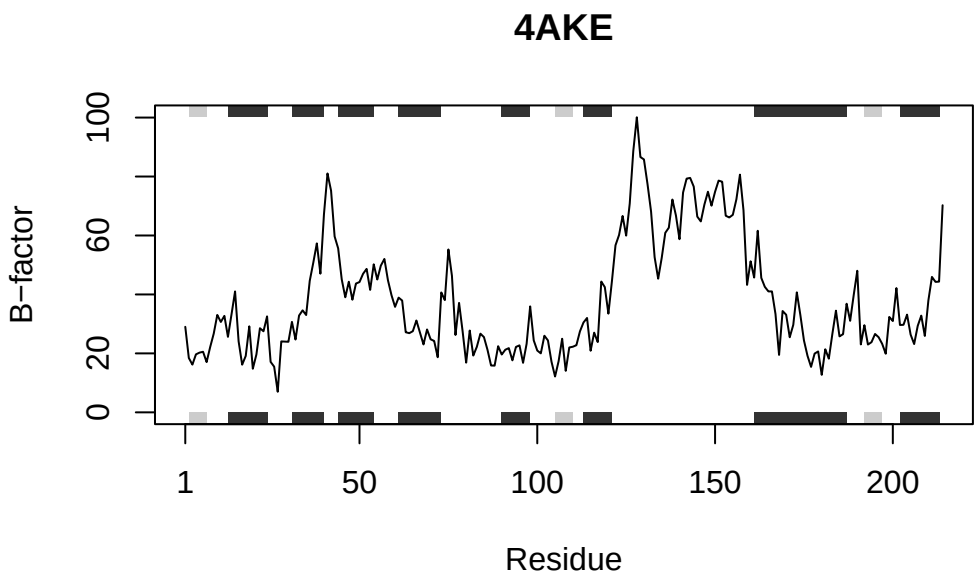
Example: - 4AKE – kinase bound to a drug (inhibitor) - 1AKE – unbound kinase (apo form) - 1E4Y – another kinase–drug complex

> Each plot shows the B-factor for residues in chain A. Higher B-factors correspond to greater flexibility or motion in that region of the protein.

```
plot_protein_bfactor("4AKE")  # kinase with drug
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\LINDAK~1\AppData\Local\Temp\RtmpSmIdBY/4AKE.pdb exists. Skipping
download
```

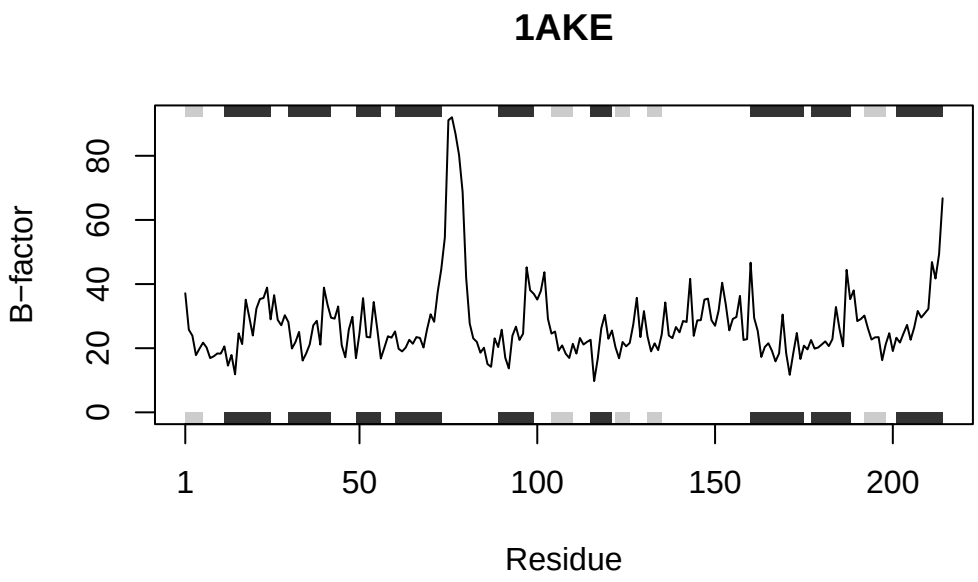



```
plot_protein_bfactor("1AKE")  # kinase without drug
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\LINDAK~1\AppData\Local\Temp\RtmpSmIdBY/1AKE.pdb exists. Skipping
download
```

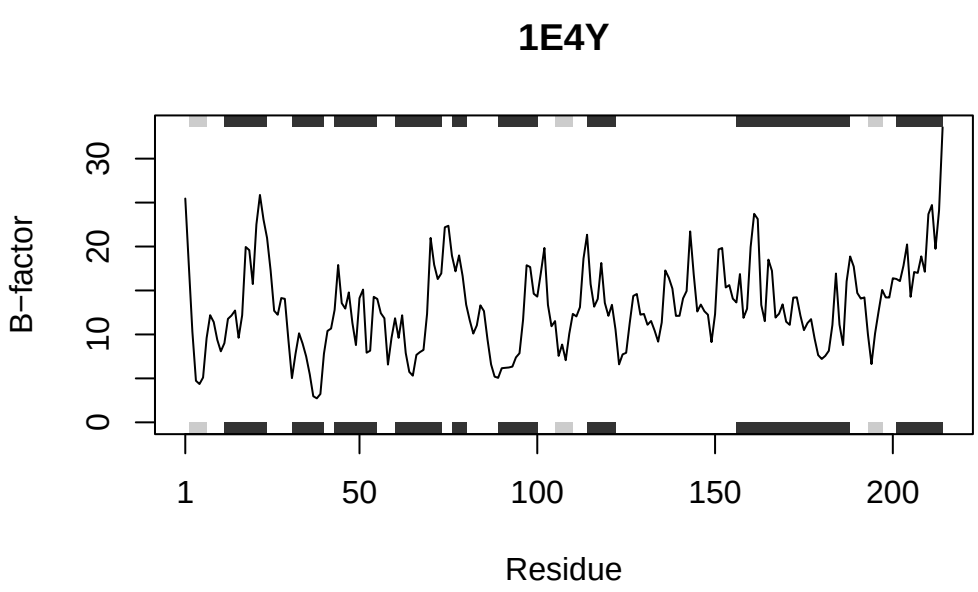
```
   PDB has ALT records, taking A only, rm.alt=TRUE
```




```
plot_protein_bfactor("1E4Y")  # kinase with drug
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\LINDAK~1\AppData\Local\Temp\RtmpSmIdBY/1E4Y.pdb exists. Skipping
download
```



This explains why we need a function and what problem it solves compared to repeating code for each protein.

For the plots:

- Regions with low B-factors are rigid and stable.
- Regions with high B-factors are more flexible or disordered.
- Comparing these plots shows how ligand binding (drug presence) changes the flexibility of specific parts of the protein.

Conclusion

The function `plot_protein_bfactor()` consolidates the repetitive steps of reading, trimming, and plotting B-factors for any protein, allowing comparisons across multiple proteins without rewriting the same code.