Lab 6 Homework

Thoi Tran (A17035545)

2024-10-19

Install package `bio3d` in the consol and not everytime we render the code with function `install.packages`.

```
## install.packages("bio3d")
library(bio3d)
```

Examine the `bio3d` package and see how the codes work.

Q1. What type of object is returned from the read.pdb() funciton?

```
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
```

The `read.pdb()` function reads Protein Data Bank files and returns a list of the components representing different aspects of the protein structures.

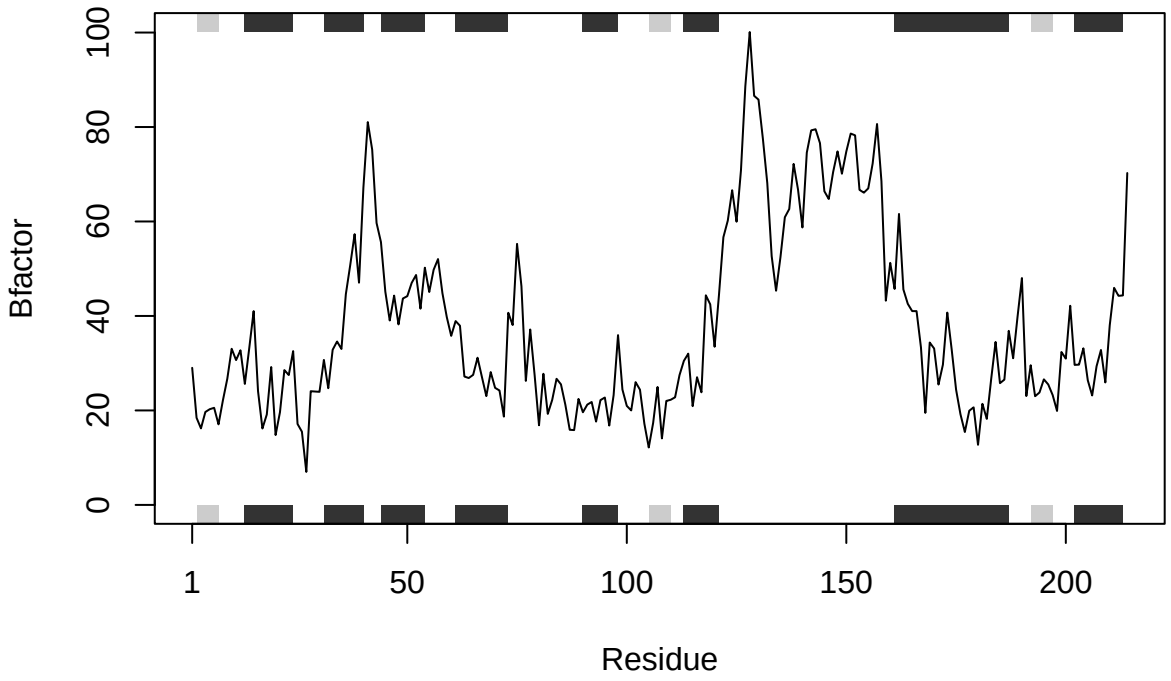
Q2. What does the `trim.pdb()` function do?

```
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")
```

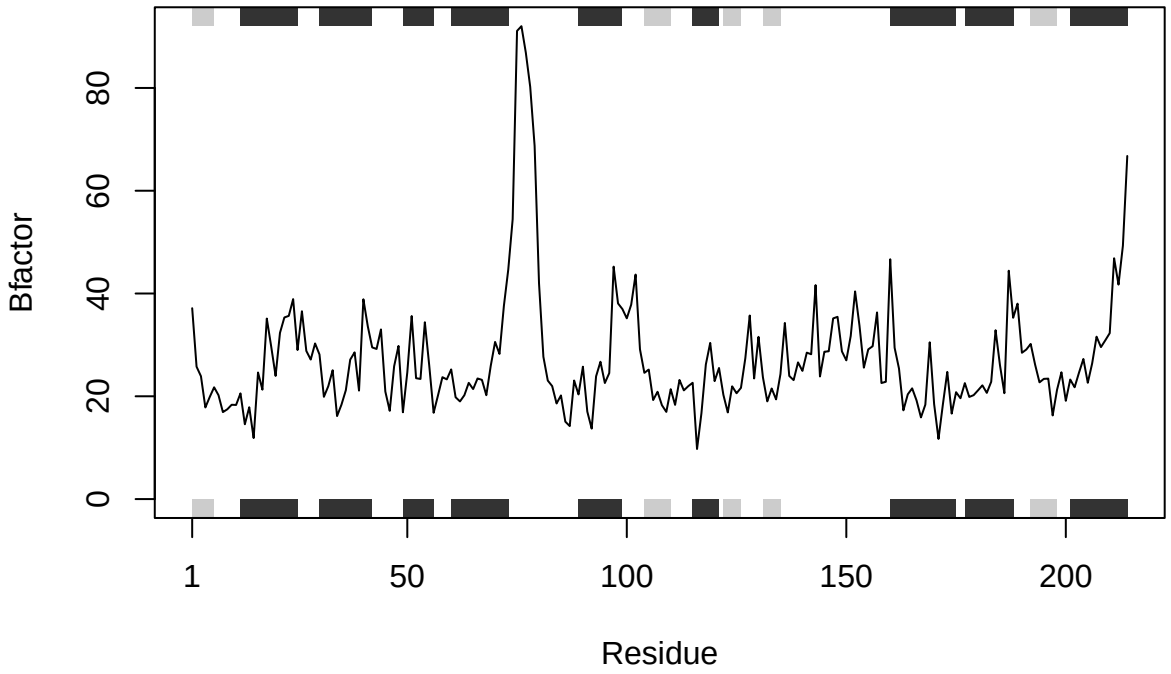
The `trim.pdb()` function extract a subset of atoms from a PDB object. In this case, it selects the atoms in Chain A of the protein (from `chain = "A"` argument) and the alpha carbon atoms (from `elety = "CA"` argument).

Q3. What input parameter would turn off the marginal black and grey rectangles in the plots and what do they represent in this case?

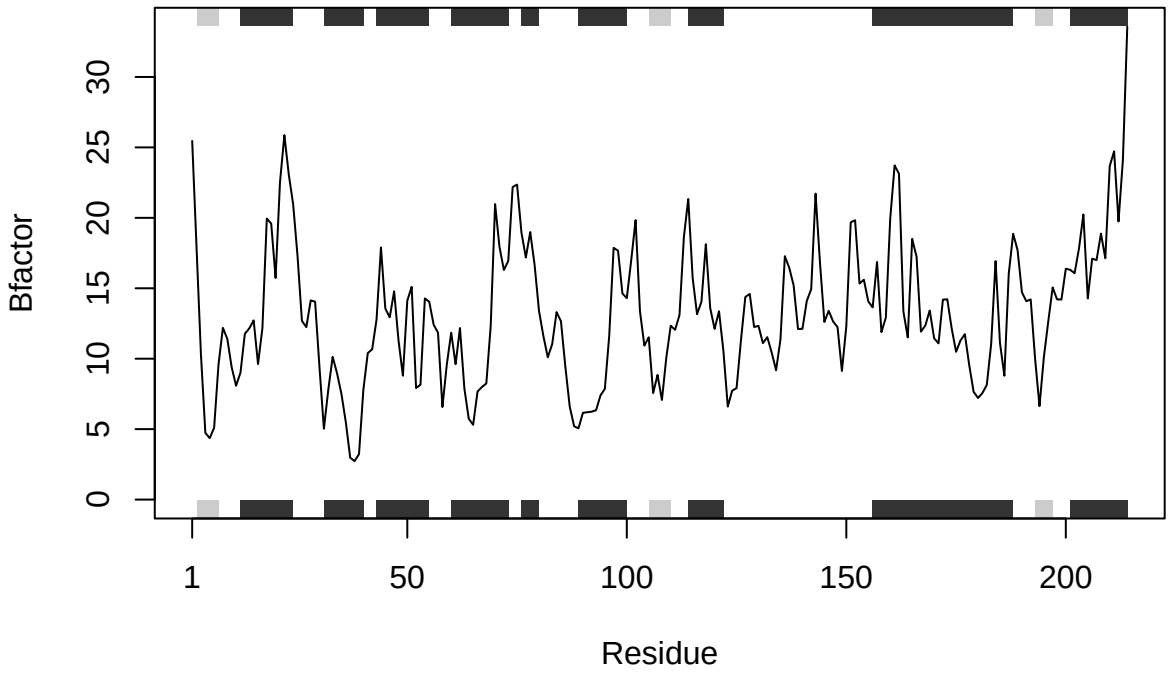
```
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```




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

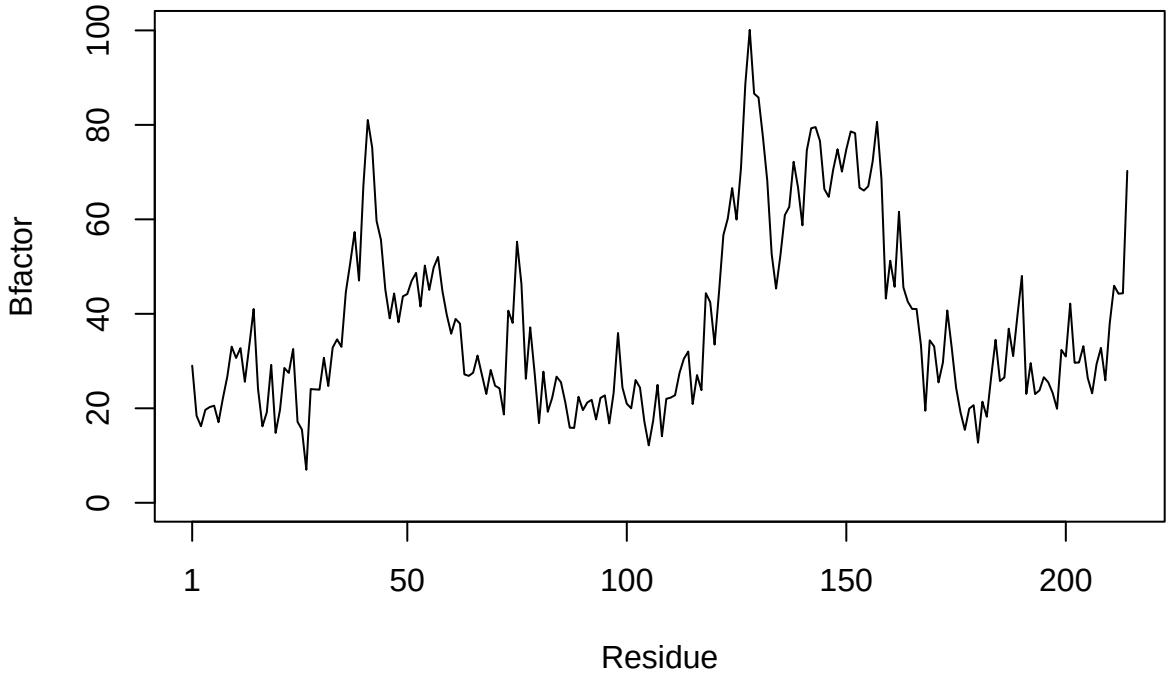



```
plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")
```

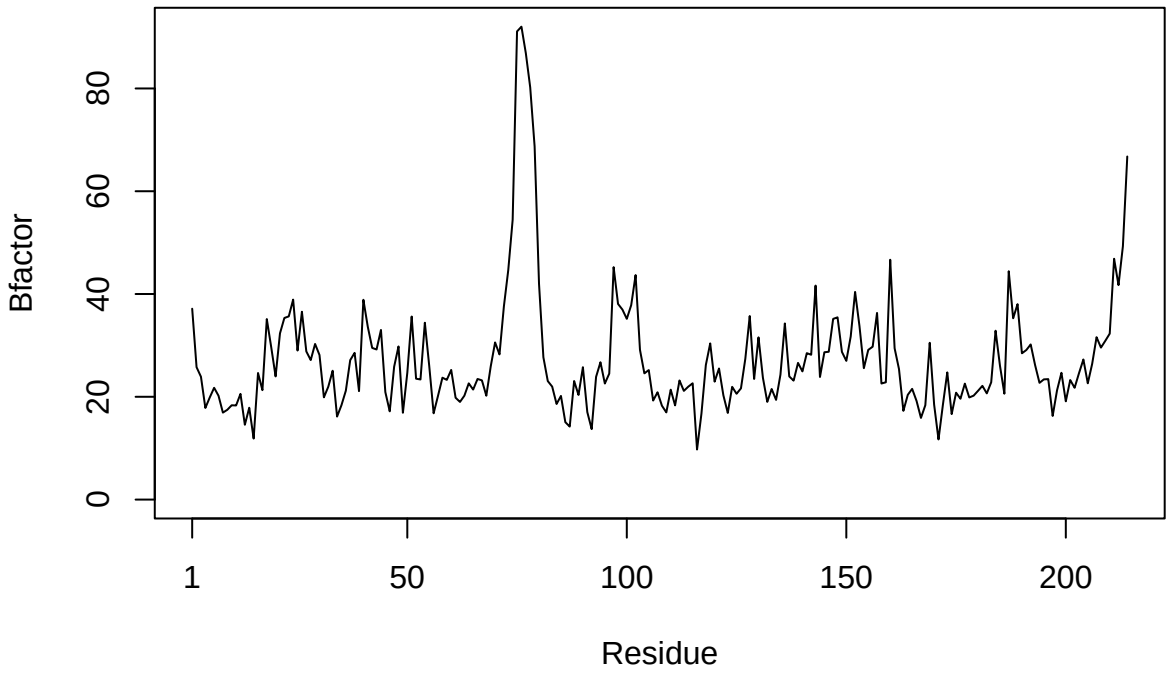


The marginal black and grey rectangles in the plots represent the secondary structures of the proteins. Black rectangles indicate regions of alpha helices. Grey rectangles indicate regions of beta sheets.

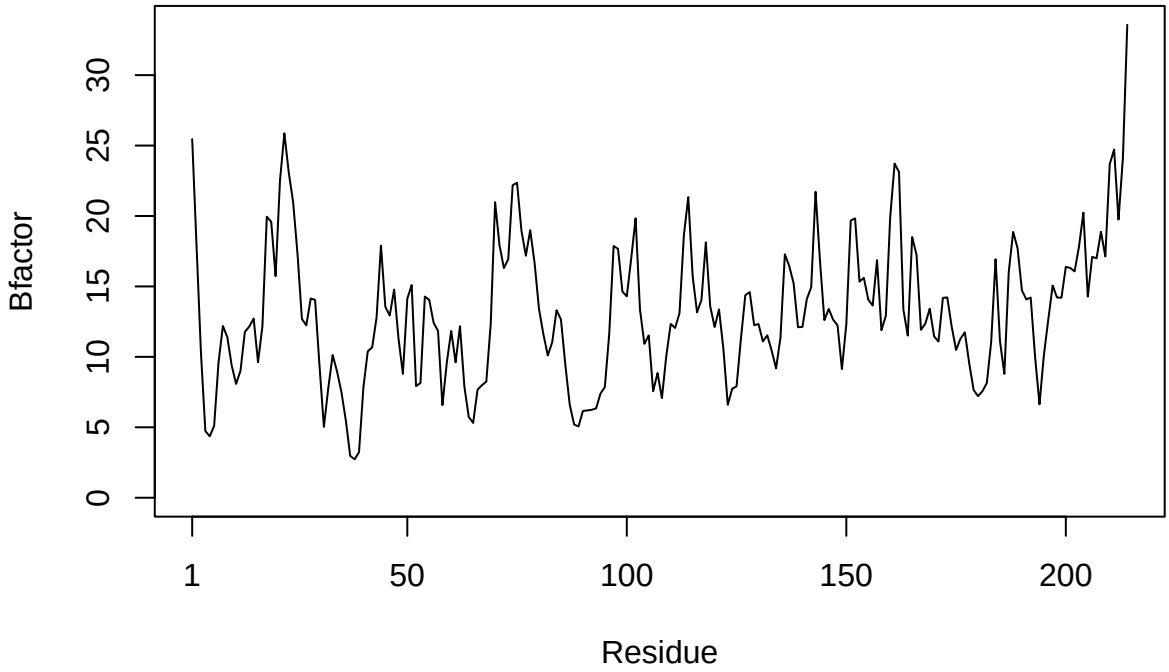
```
plotb3(s1.b, typ="l", ylab="Bfactor")
```




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




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



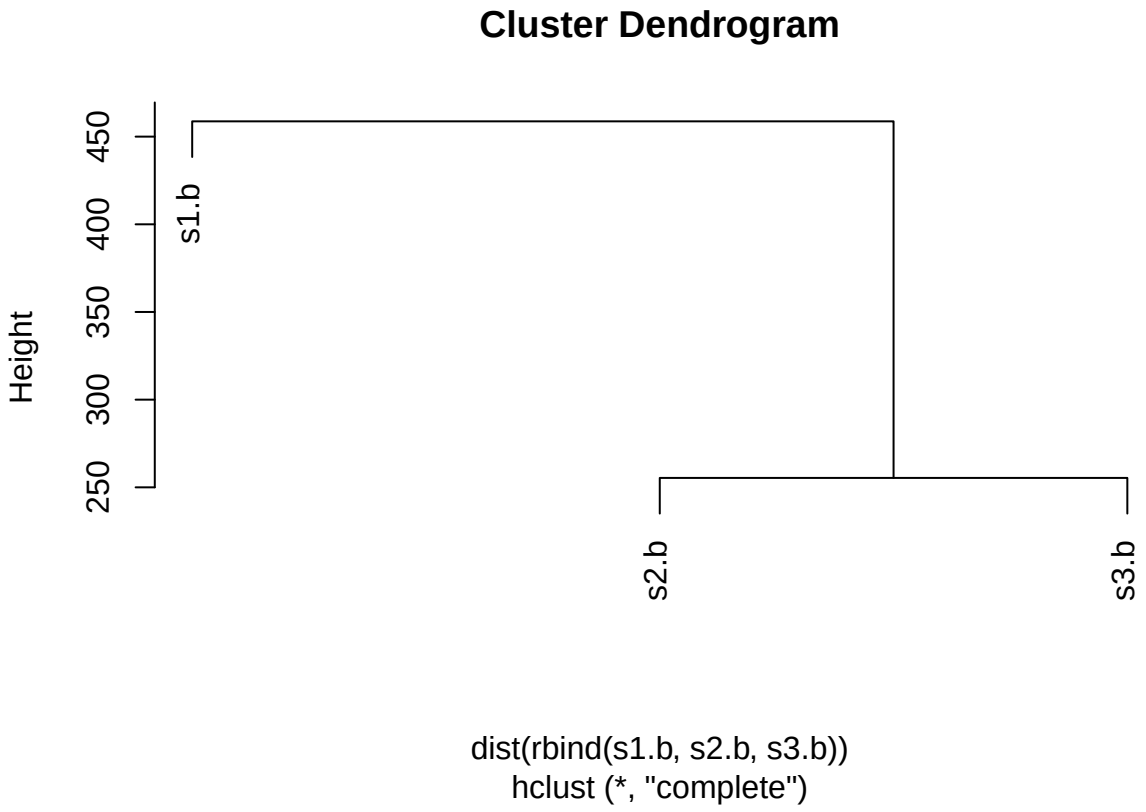
To turn off the marginal black and grey rectangles, we can eliminate the `sse` parameter.

Q4. What would be a better plot to compare across the different proteins?

A single line plot with all 3 proteins B-factor would be a better plot to compare across the different proteins.

Q5. Which proteins are more similar to each other in their B-factor trends. How could you quantify this?

```
hc <- hclust( dist( rbind(s1.b, s2.b, s3.b) ) )
plot(hc)
```



Based on the cluster dendogram plot, proteins 2 and 3 are more similar to each other in their B-factors trends.

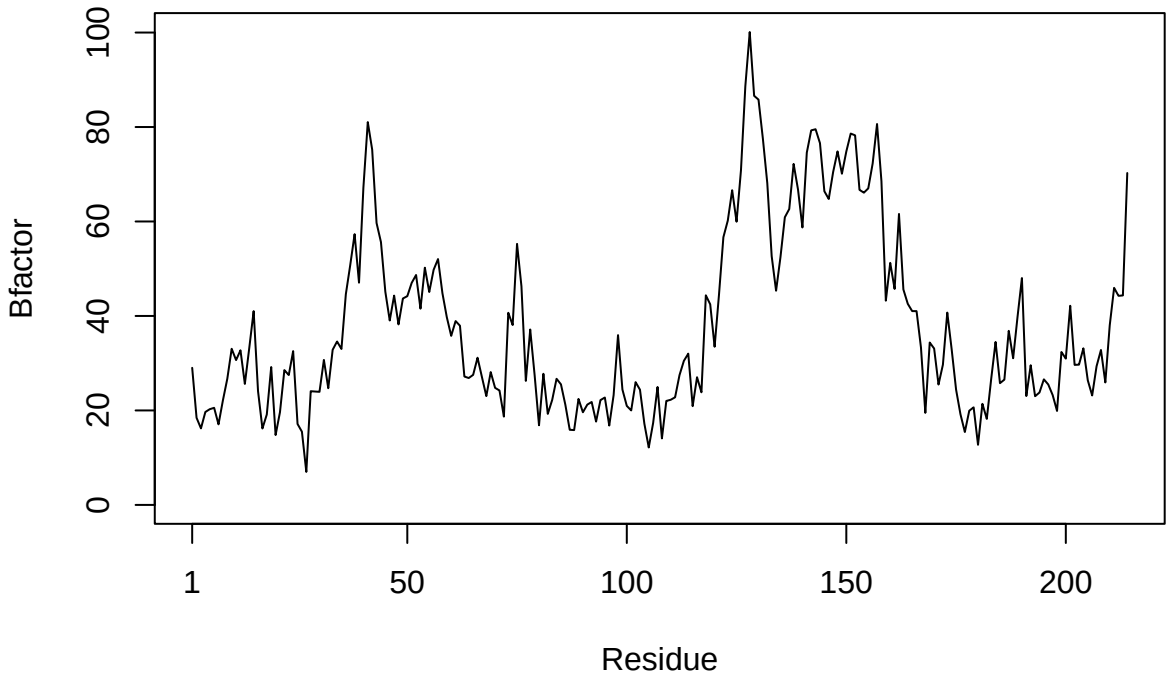
Homework

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

```
plot_function <- function(pdb_id, chain_input, elety_input) {
  structure <- read.pdb(pdb_id)
  structure.chain <- trim.pdb(structure,
                              chain = chain_input,
                              elety = elety_input)
  structure.b <- structure.chain$atom$b
  plotb3(structure.b, typ = "l", ylab = "Bfactor")
}
```

```
plot_function("4AKE", "A", "CA")
```

```
##   Note: Accessing on-line PDB file
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/2p/vsw78bsj1fzdsjd5x8_nvqch0000gn/T//RtmplYuYgk/4AKE.pdb exists.
## Skipping download
```

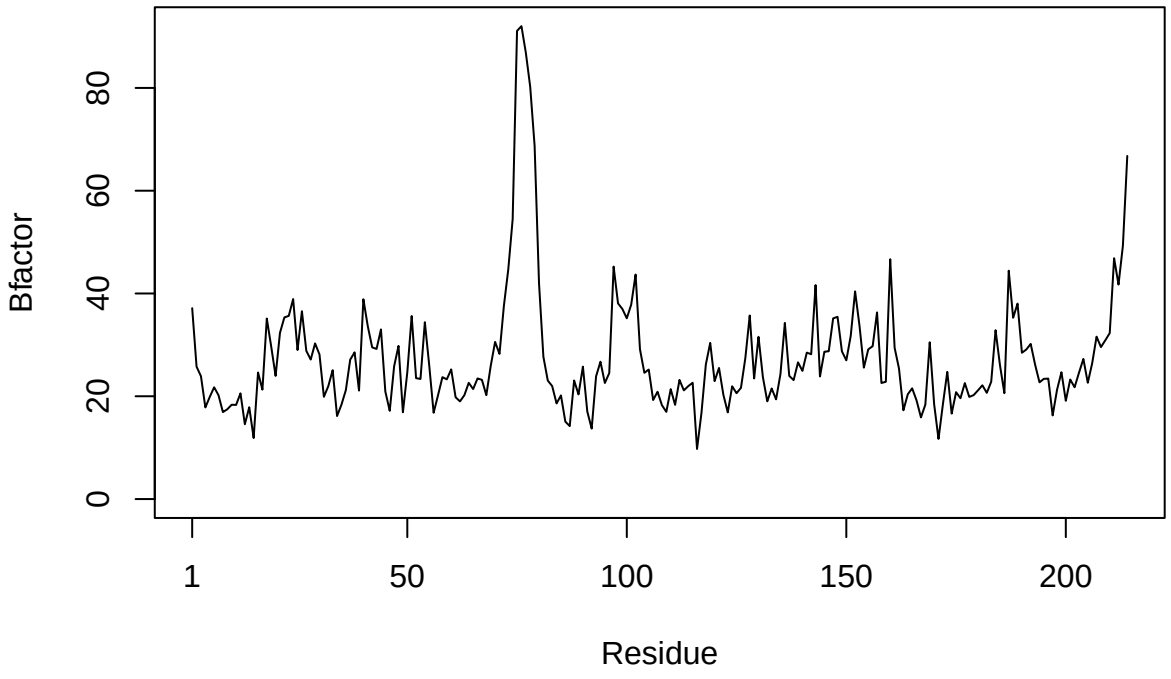



```
plot_function("1AKE", "A", "CA")
```

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/2p/vsw78bsj1fzdsjd5x8_nvqch0000gn/T//RtmplYuYgk/1AKE.pdb exists.
## Skipping download
```

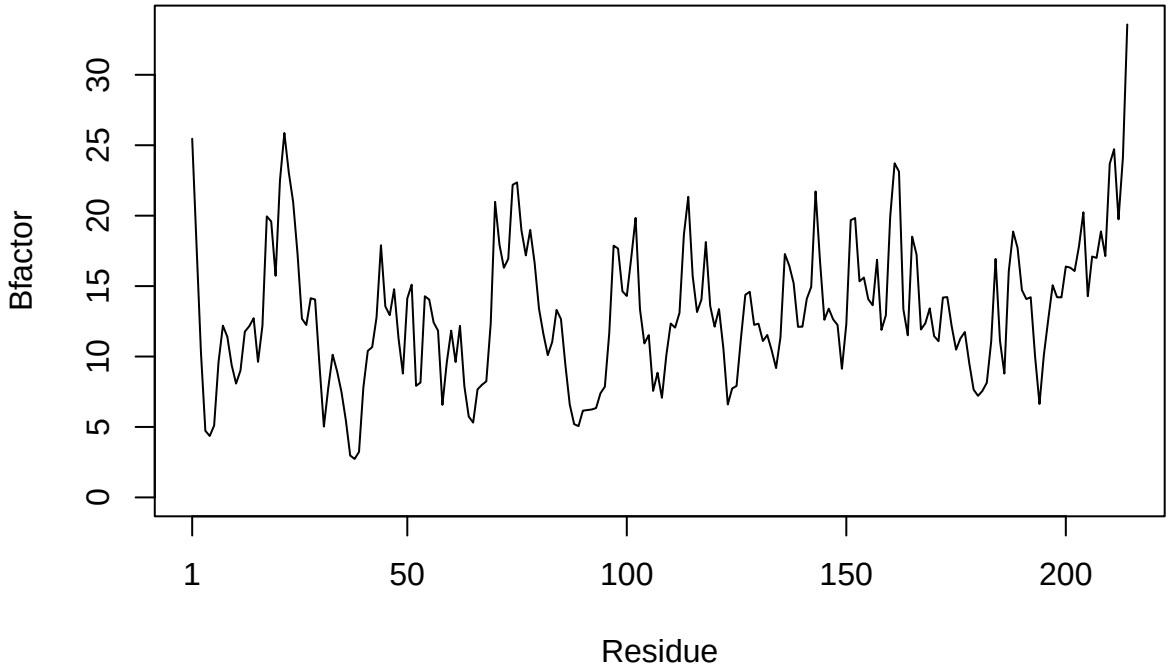
```
##    PDB has ALT records, taking A only, rm.alt=TRUE
```




```
plot_function("1E4Y", "A", "CA")
```

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

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## /var/folders/2p/vsw78bsj1fzdsjd5x8_nvqch0000gn/T//RtmplYuYgk/1E4Y.pdb exists.
## Skipping download
```



There are 3 inputs to the `plot_function()`:

- **pdb__id**: the pdb id of the protein structure
- **chain__input**: slect the chain of protein structure
- **elety__input**: select the alpha carbon of the protein structure

Here is how the `plot_function()` works:

- `read.pdb()` function reads the pdb structure of a protein and assigns it to `structure`
- `trim.pdb()` function selects the atoms from a chain and the alpha carbon from the `structure` and assigns it to `structure.chain`
- Filters the B-factor from `structure.chain` and assigsn it to `structure.b`
- `plotb3()` function plots the B-factor from `structure.b`

The output of the `plot_function` is a plot that shows the relationship between B_factors and Residues in a given protein structure.