06-HW-!.R

Angela

2025-10-20

```r
#Section 1A
normalize <- function(x) {
  rng <- range(x, na.rm = TRUE)
  (x - rng[1]) / (rng[2] - rng[1])
}

df <- data.frame(a=1:10, b=seq(200,400,length=10), c=11:20, d=NA)
df[] <- lapply(df, normalize)
```

```
## Warning in min(x, na.rm = na.rm): no non-missing arguments to min; returning
## Inf
```

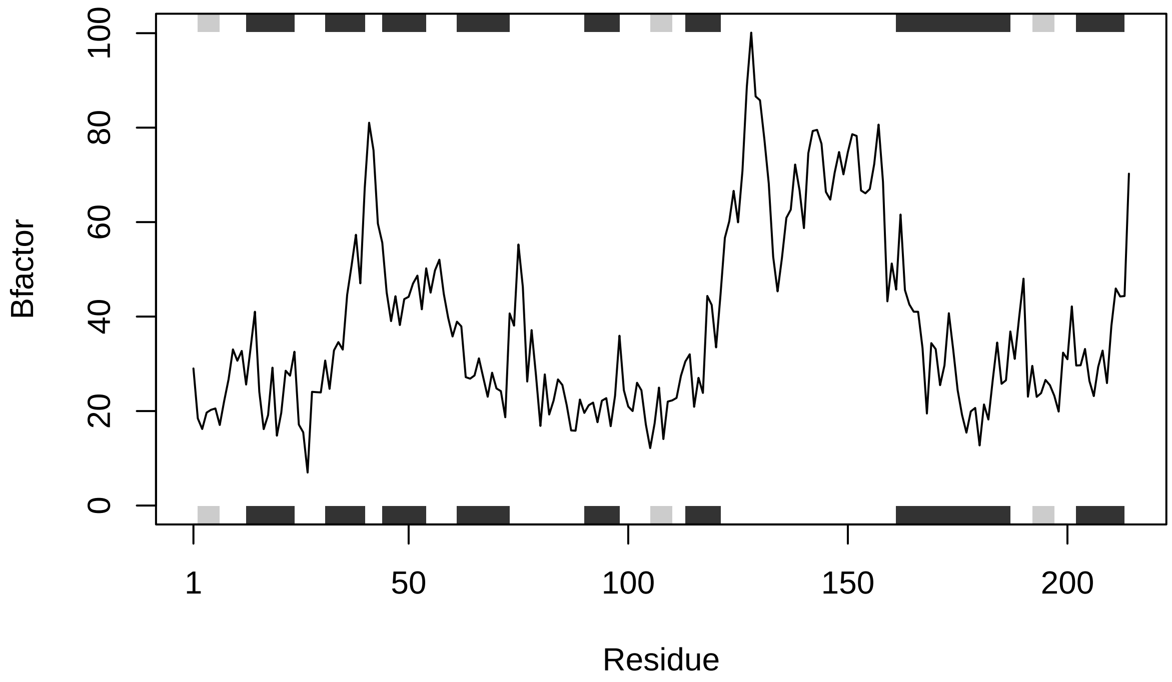
```
## Warning in max(x, na.rm = na.rm): no non-missing arguments to max; returning
## -Inf
```

```r
print(df)
```

```
##            a         b         c  d
## 1  0.0000000 0.0000000 0.0000000 NA
## 2  0.1111111 0.1111111 0.1111111 NA
## 3  0.2222222 0.2222222 0.2222222 NA
## 4  0.3333333 0.3333333 0.3333333 NA
## 5  0.4444444 0.4444444 0.4444444 NA
## 6  0.5555556 0.5555556 0.5555556 NA
## 7  0.6666667 0.6666667 0.6666667 NA
## 8  0.7777778 0.7777778 0.7777778 NA
## 9  0.8888889 0.8888889 0.8888889 NA
## 10 1.0000000 1.0000000 1.0000000 NA
```

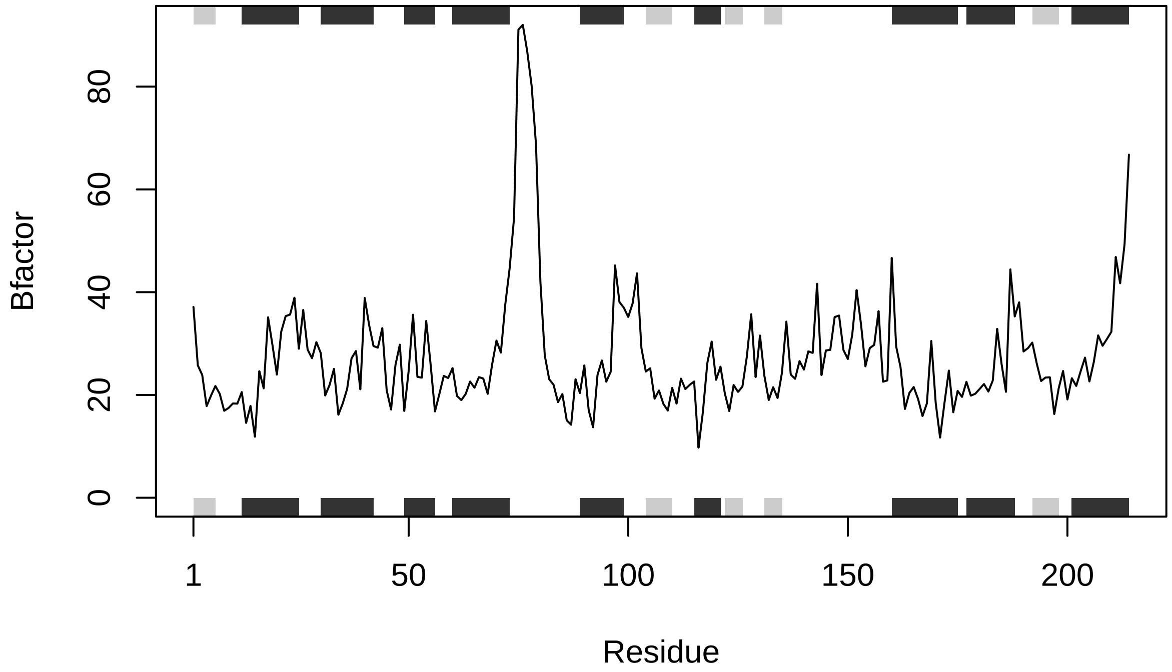
```r
#Section 1B
#install.packages("bio3d")
library(bio3d)

analyze_protein <- function(pdb_code) {
  library(bio3d)

  # Read PDB file
  pdb <- read.pdb(pdb_code)

  # Trim to chain A, C-alpha atoms
  pdb_chain <- trim.pdb(pdb, chain="A", elety="CA")

  # Extract B-factors
  bfactor <- pdb_chain$atom$b

  # Plot
  plotb3(bfactor, sse=pdb_chain, typ="l", ylab="Bfactor")

  return(bfactor)
}

# Example usage:
s1.b <- analyze_protein("4AKE")
```

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



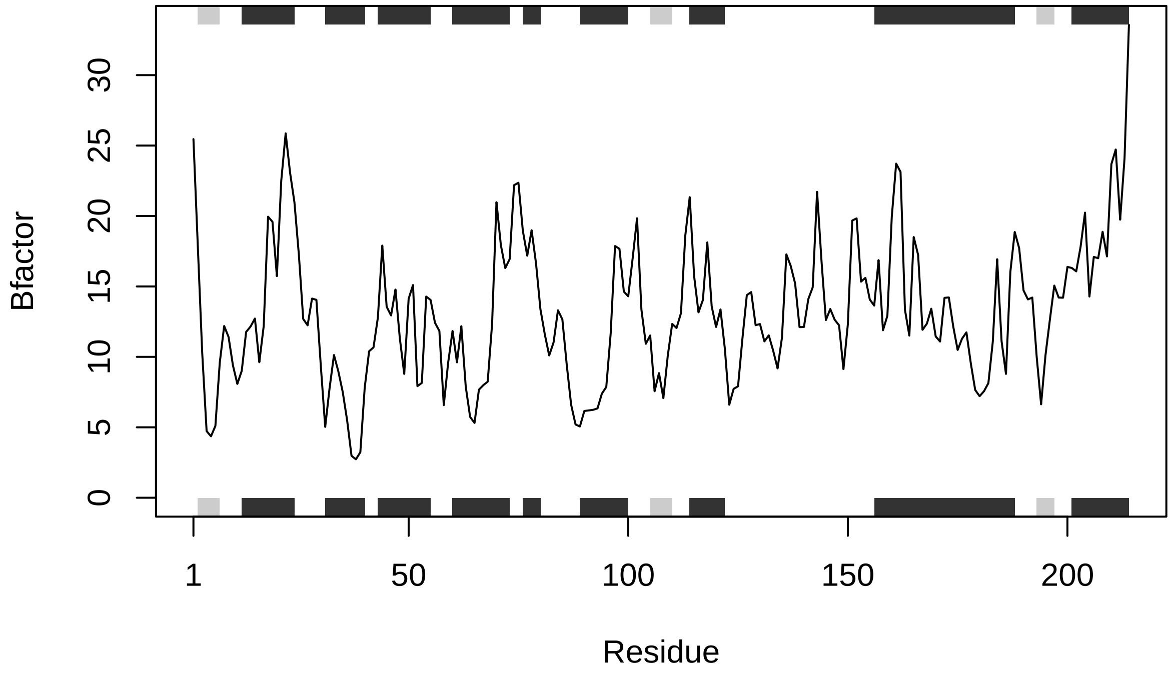
```
s2.b <- analyze_protein("1AKE")
```

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



```
s3.b <- analyze_protein("1E4Y")
```

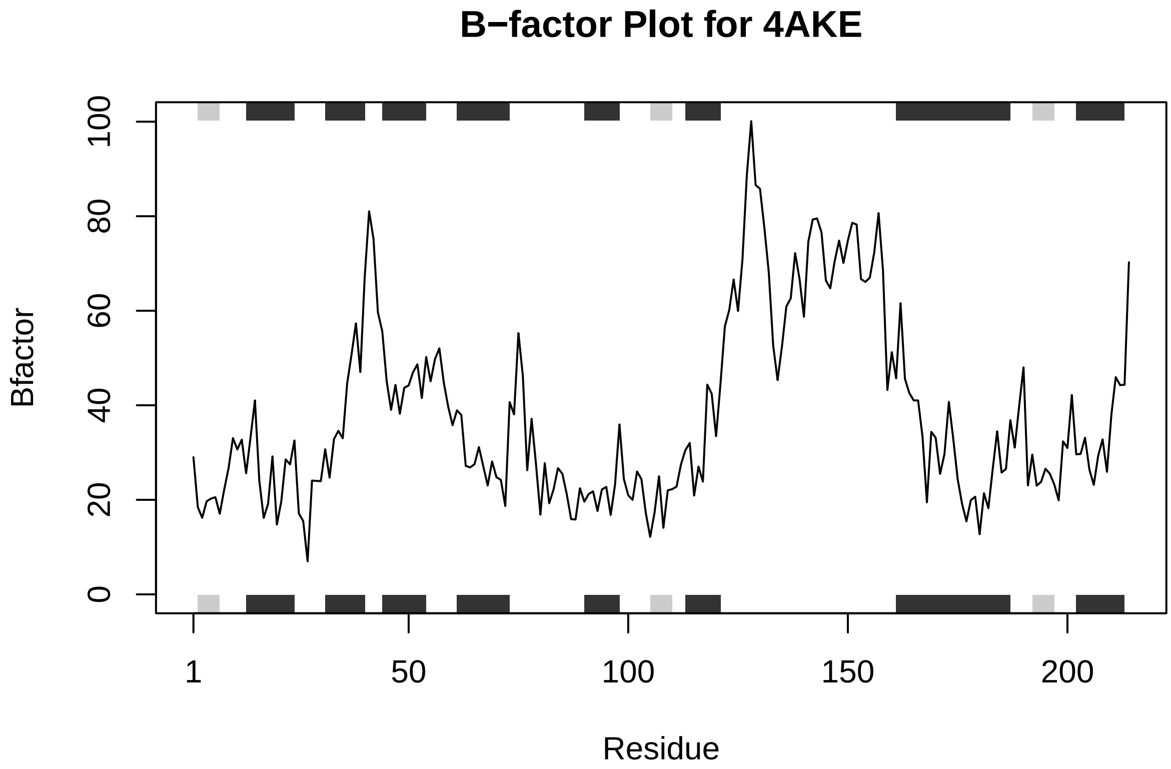
```
##   Note: Accessing on-line PDB file
```



```
#Q1) A list object of class "pdb"
#Q2) Extracts subset of atoms from PDB structure based on specific criteria like chain ID and element t
#Q3) Parameter: sse = NULL; grey rectangles represent alpha helices and black rectangles represent beta
#Q4) Overlay plot showing all 3 proteins on the same axes with different colors and legend
#Q5) The proteins that cluster together first are most similar in their B-factor trends. This can be qu
#Q6)

analyze_protein_general <- function(pdb_code, chain="A", plot=TRUE) {
  # This function analyzes protein drug interactions by reading in any
  # protein PDB data and outputting a plot for the specified protein
  #
  # Inputs:
  #   - pdb_code: A 4-letter PDB code (e.g., "4AKE") or file path
  #   - chain: The protein chain to analyze (default: "A")
  #   - plot: whether to generate a plot (default: TRUE)
  #
  # Output:
  #   - Returns a numeric vector of B-factor values for the protein

  library(bio3d)

  # Read PDB file
  pdb <- read.pdb(pdb_code)

  # Trim to specified chain and C-alpha atoms
  pdb_chain <- trim.pdb(pdb, chain=chain, elety="CA")
  # Extract B-factors
  bfactor <- pdb_chain$atom$b

  # Generate plot if requested
  if (plot) {
    plotb3(bfactor, sse=pdb_chain, typ="l", ylab="Bfactor",
           main=paste("B-factor Plot for", pdb_code))
  }

  # Return B-factor values
  return(bfactor)
}

# Example usage showing the function works:
result1 <- analyze_protein_general("4AKE")
```

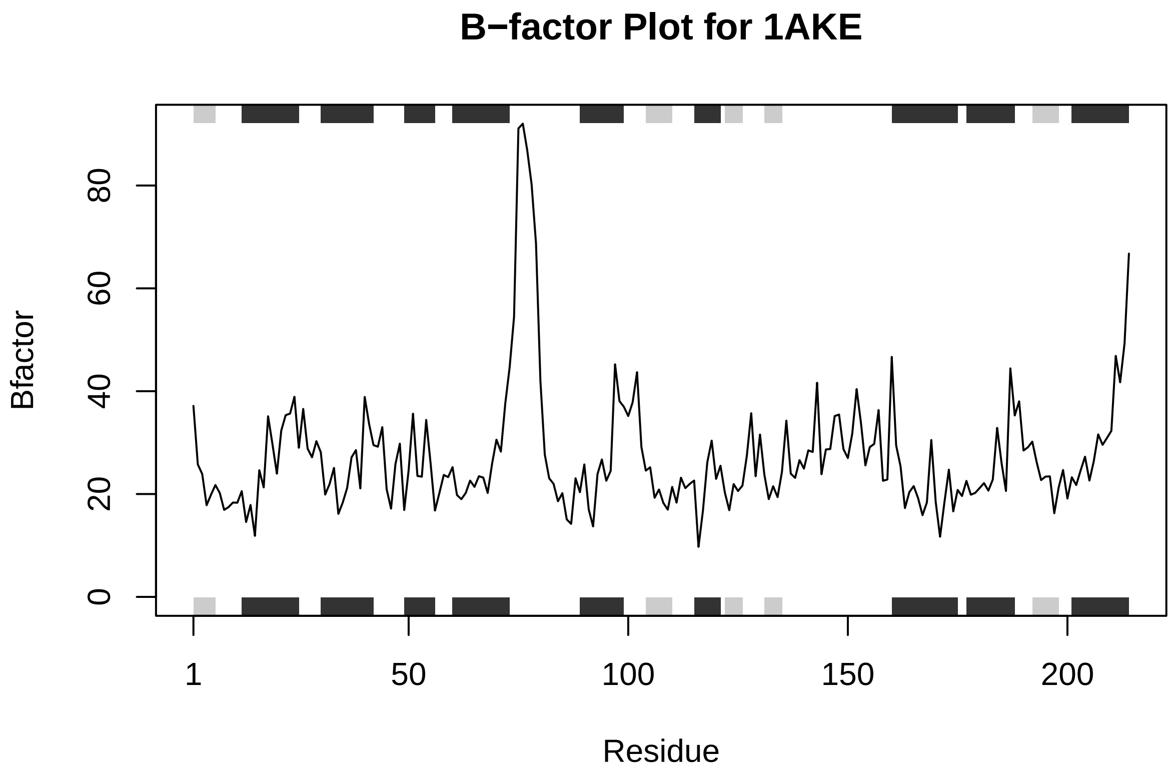
```
##   Note: Accessing on-line PDB file
```

```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\Angela\AppData\Local\Temp\RtmpU1Jwrm/4AKE.pdb exists. Skipping
## download
```



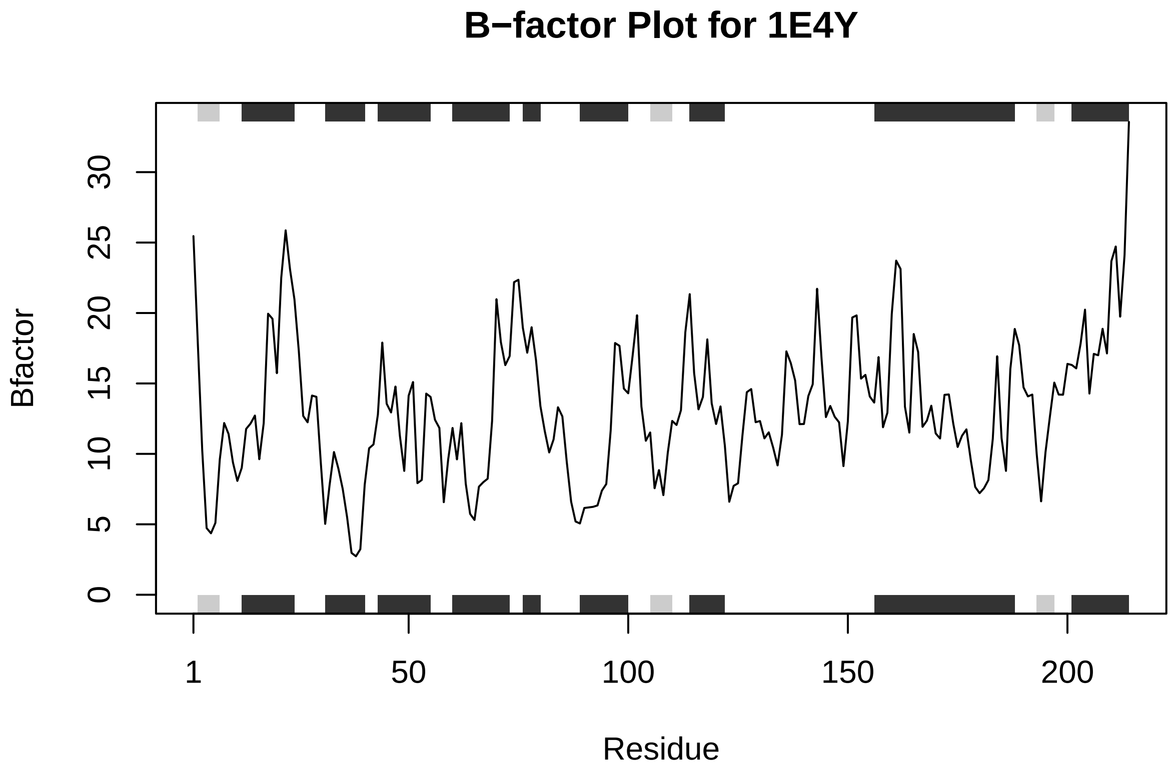
```r
result2 <- analyze_protein_general("1AKE")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\Angela\AppData\Local\Temp\RtmpU1Jwrm/1AKE.pdb exists. Skipping
## download

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



```r
result3 <- analyze_protein_general("1E4Y")
```

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

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
## C:\Users\Angela\AppData\Local\Temp\RtmpU1Jwrm/1E4Y.pdb exists. Skipping
## download
```


B−factor Plot for 1E4Y
Bfactor
0
5
10
15
20
25
30
Residue
1
50
100
150
200