HW06: R Functions

Jocelyn Olvera (A12459601)

10/23/2021

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

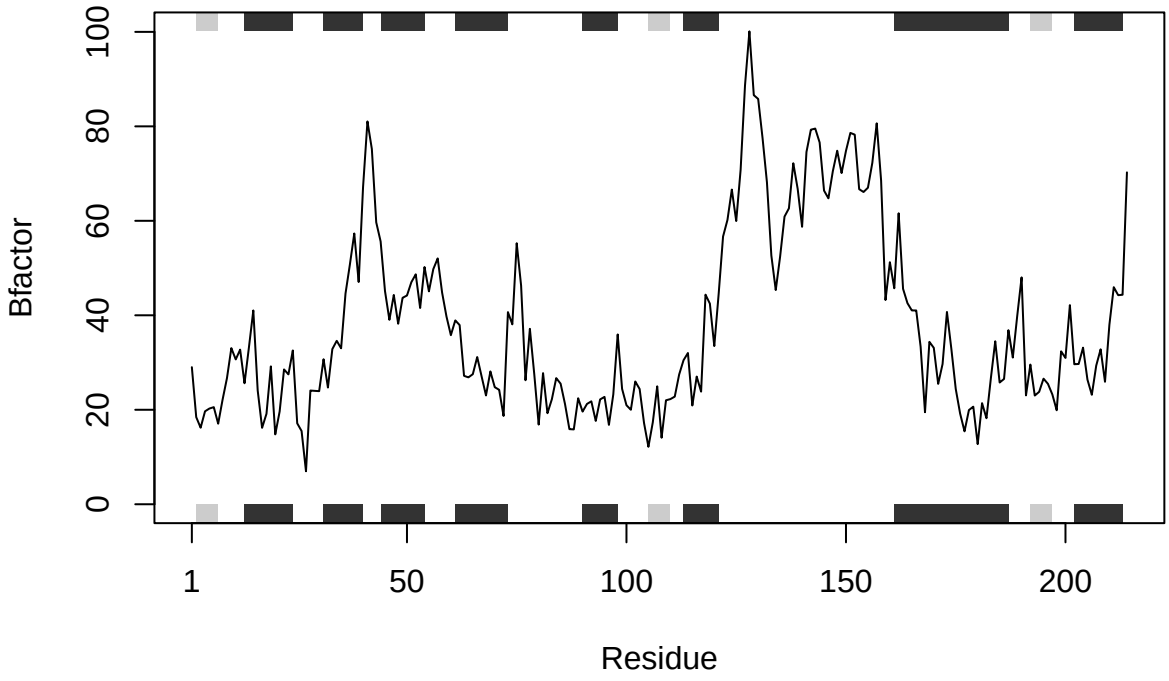
Q6: **My function to view any protein using its pdb code name.**

```
# As per usual, we load the package we will use.
library(bio3d)
```

Writing my own function to produce a plot from the protein 'pdb' code. This function adopts the code from the workbook and modifies it to generate a function easy to work with.

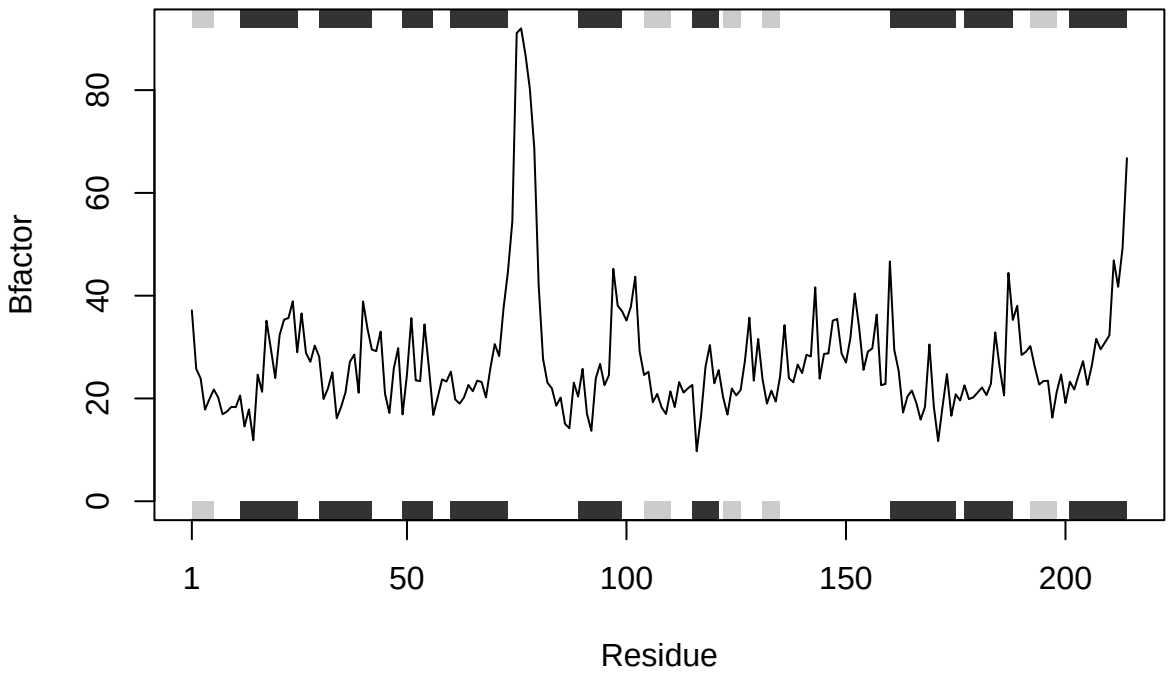
```
Protein_plot <- function(prot_pdb) {
   pdb.data <- read.pdb(prot_pdb) #first we input the data by reading in the file.
   # Next, trim data to the calpha atoms.
   pdb.data.chainA <- trim.pdb(pdb.data, chain="A", elety="CA")
   pdb.data.b <- pdb.data.chainA$atom$b #
   # Finally, this generates the plot with alpha helixes
   # and beta strands represented as grey and black boxes.
   plotb3(pdb.data.b, sse=pdb.data.chainA, typ="l", ylab="Bfactor")

}
# Output: Protein plots.
Protein_plot("4AKE") # kinase w/ drug
```

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



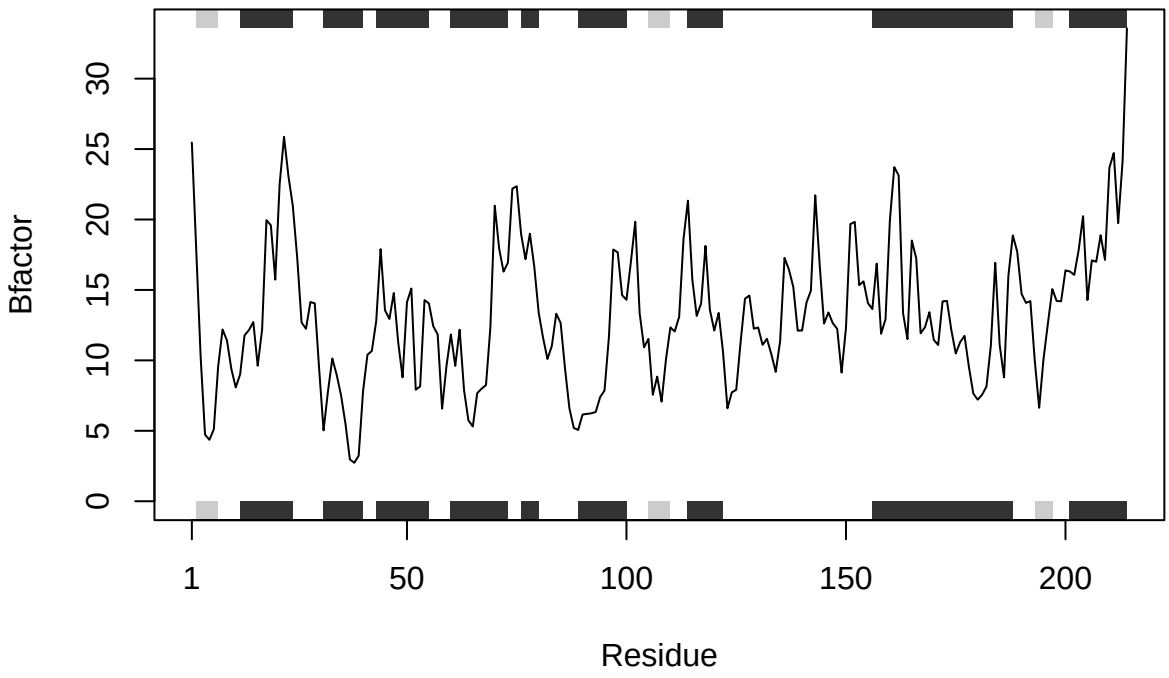

```
Protein_plot("1AKE") # kinase w/out drug
```

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




```
Protein_plot("1E4Y") # kinase w/out drug
```

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



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

Response Q1: **Using the read.pdb() function retrieves the 'pdb' objects. To view items in this object, we print 's1', 's2', or 's3', which provides atom number, protein atom number, nuecleic acid number of atoms, non-prot nucleic atoms, and non-prot nucleic residue values. This function also provides the protein sequence. Example below, code provided from workbook.**

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

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

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

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

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

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

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

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

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

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

```
s1 #print 's1' to get objects
```

```
##
##  Call:  read.pdb(file = "4AKE")
##
##    Total Models#: 1
##      Total Atoms#: 3459,  XYZs#: 10377  Chains#: 2  (values: A B)
##
##      Protein Atoms#: 3312  (residues/Calpha atoms#: 428)
##      Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
##
##      Non-protein/nucleic Atoms#: 147  (residues: 147)
##      Non-protein/nucleic resid values: [ HOH (147) ]
##
##    Protein sequence:
##       MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
##       DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
##       VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
##       YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILGMRIILLGAPGA...<cut>...KILG
##
## + attr: atom, xyz, seqres, helix, sheet,
##         calpha, remark, call
```

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

Response Q2: **This function creates a new and smaller pdb object from the selected region of the protein structure (i.e alpha chain, backbone). See example below, code provided from workbook.**

```
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")
```

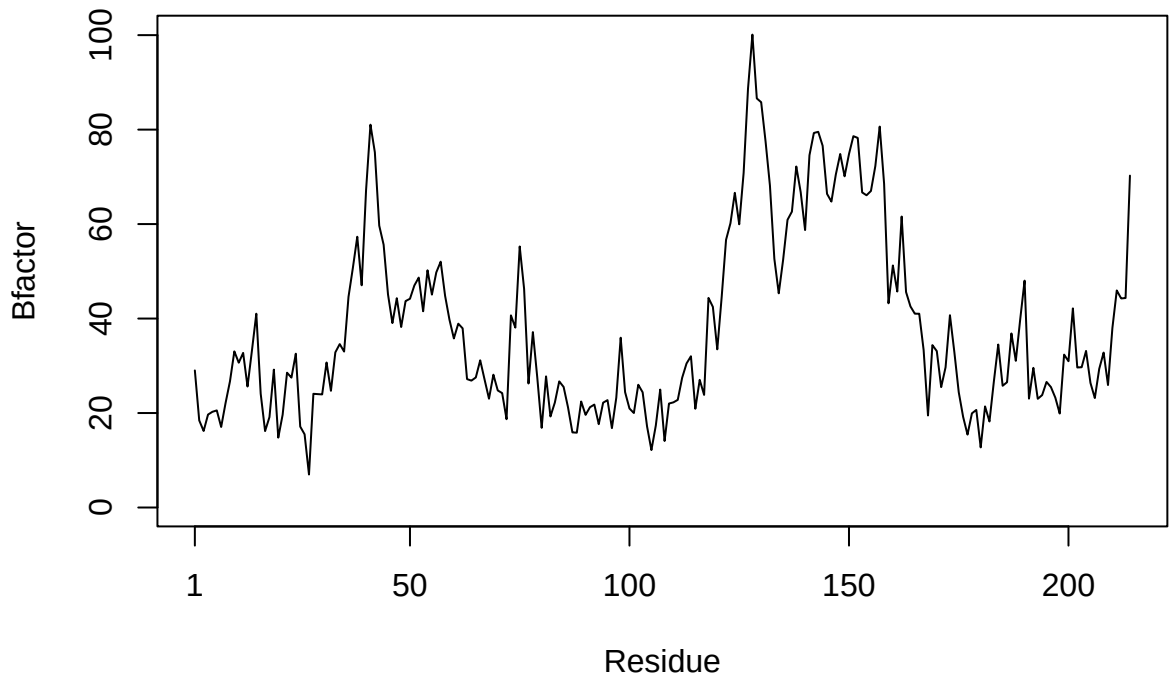
Code adjustment on 's3.chainA' changing s1 to s3.

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

Response Q3: **The 'sse' parameter turns off the black/grey rectangles in the plot. These rectangles represent the alpha helix and beta strand regions of the protein.**

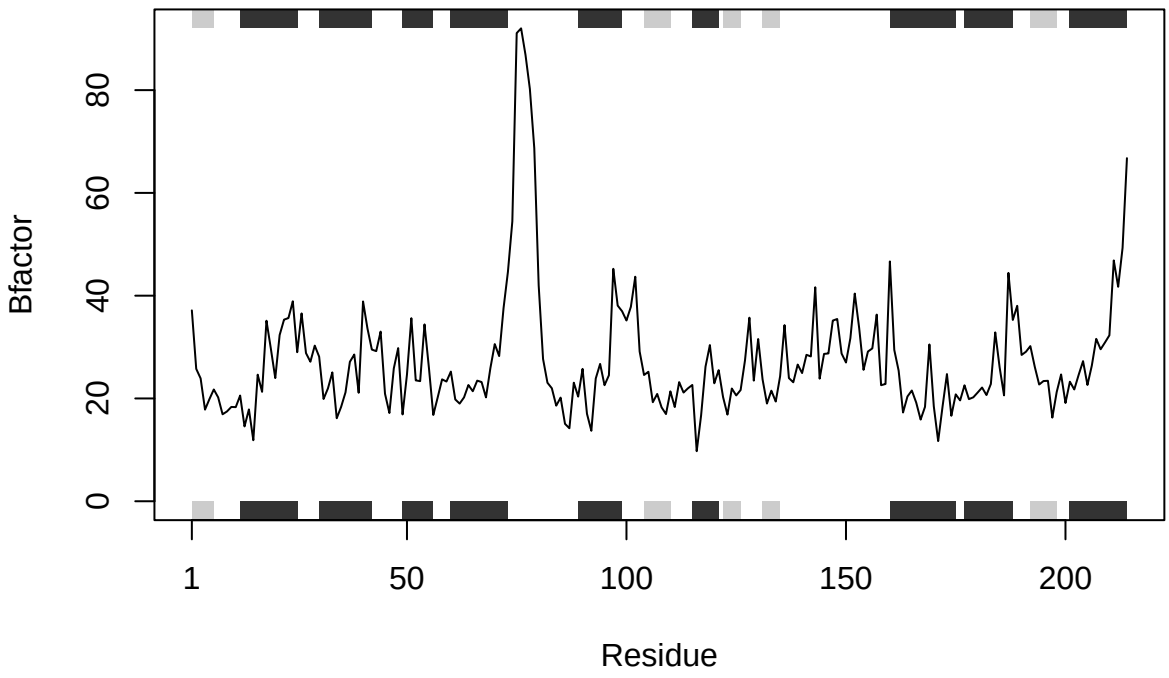
```
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
```

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

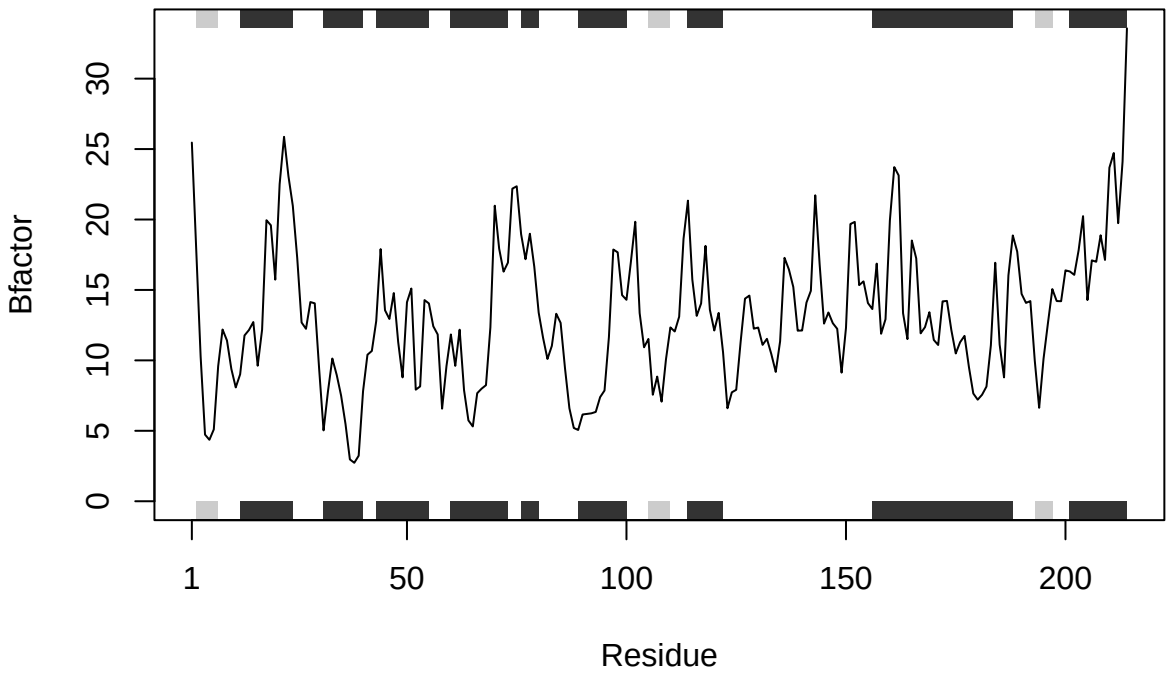



```
# Removing 'sse' to demonstrate that this parameter controls the presence of black/grey boxes.
```

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




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



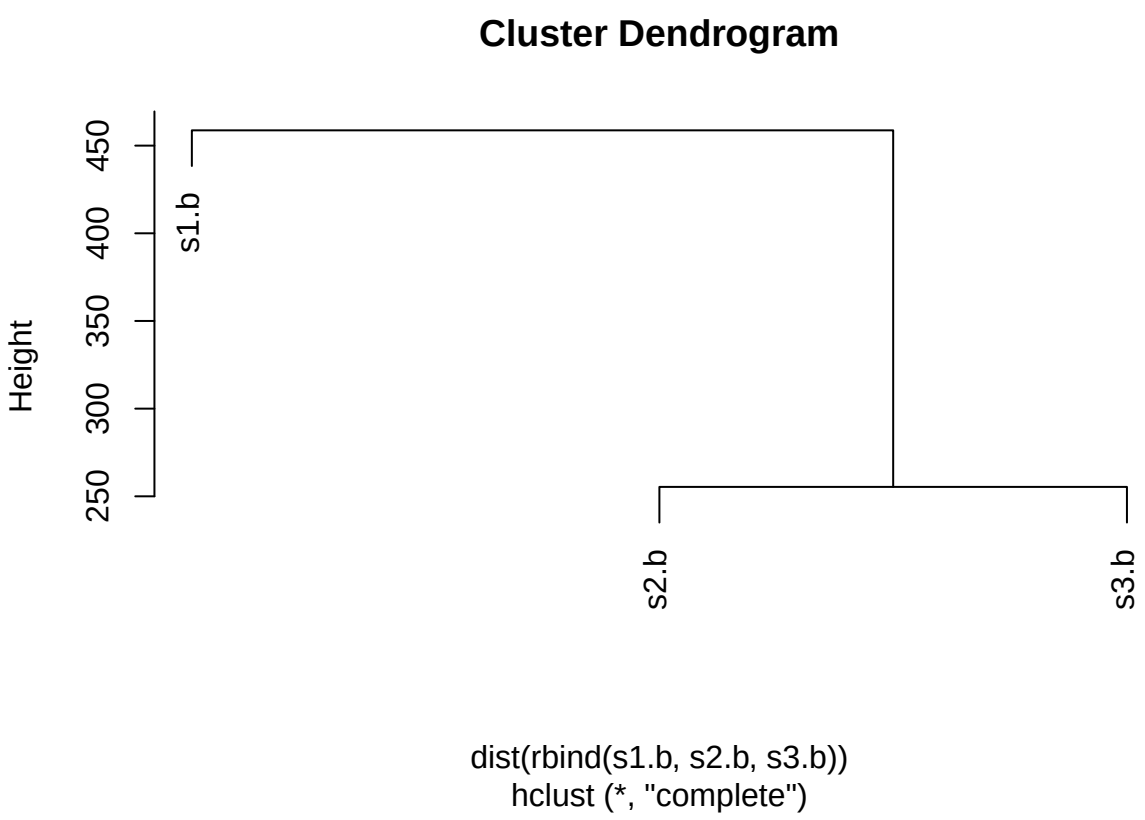
Q4. What would be a better plot to compare across the different proteins?

Response Q4: **We can use a dandogram to compare B-factors as in Q5.**

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

Response Q5: **Proteins 1AKE and 1E4Y are more similar. This can be quantified by finding the distance between the each B-factor of the proteins.**

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


Cluster Dendrogram
Height
450
400
350
300
250
s1.b
s2.b
s3.b
dist(rbind(s1.b, s2.b, s3.b))
hclust (*, "complete")