

Class06 extension

Michael Preston (PID: A53310268)

Section 1: Improving analysis code by writing functions

A.

Improve this regular R code by abstracting the main activities in your own new function. Note, we will go through this example together in the formal lecture. The main steps should entail running through the code to see if it works, simplifying to a core working code snippet, reducing any calculation duplication, and finally transferring your new streamlined code into a more useful function for you.

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

	a	b	c	d
1	1	200.0000	11	NA
2	2	222.2222	12	NA
3	3	244.4444	13	NA
4	4	266.6667	14	NA
5	5	288.8889	15	NA
6	6	311.1111	16	NA
7	7	333.3333	17	NA
8	8	355.5556	18	NA
9	9	377.7778	19	NA
10	10	400.0000	20	NA

```
norm <- function(data){
  nord_data <- (data - min(data)) / (max(data) - min(data))
  return(nord_data)
}
```

```
df[] <- lapply(df, norm)
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
```

B.

Next improve the below example code for the analysis of protein drug interactions by abstracting the main activities in your own new function. Then answer questions 1 to 6 below. It is recommended that you start a new Project in RStudio in a new directory and then install the bio3d package noted in the R code below (N.B. you can use the command `install.packages("bio3d")` or the RStudio interface to do this). Then run through the code to see if it works, fix any copy/paste errors before simplifying to a core working code snippet, reducing any calculation duplication, and finally transferring it into a more useful function for you.

```
# install the bio3d package
# install.packages("bio3d")
```

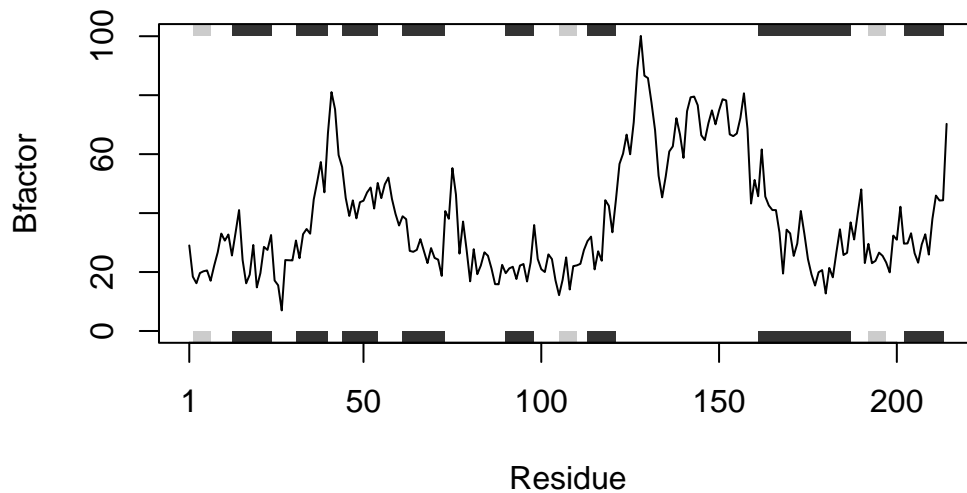
```
# Can you improve this analysis code?
library(bio3d)

run_analysis <- function(pdb_name){
  s <- read.pdb(pdb_name)
  s.chainA <- trim.pdb(s, chain="A", elety="CA")
  s.b <- s.chainA$atom$b
  plotb3(s.b, sse=s.chainA, typ="l", ylab="Bfactor")
  return(s.b)
}

#for (pdb_name in c("4AKE", "1AKE", "1E4Y")){
```

```
# run_analysis(pdb_name)
#}
s1.b <- run_analysis("4AKE")
```

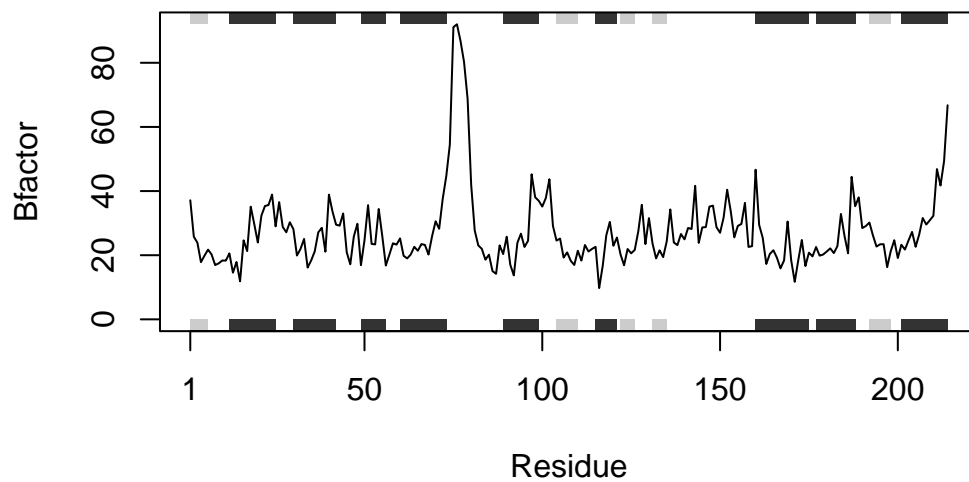
Note: Accessing on-line PDB file



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

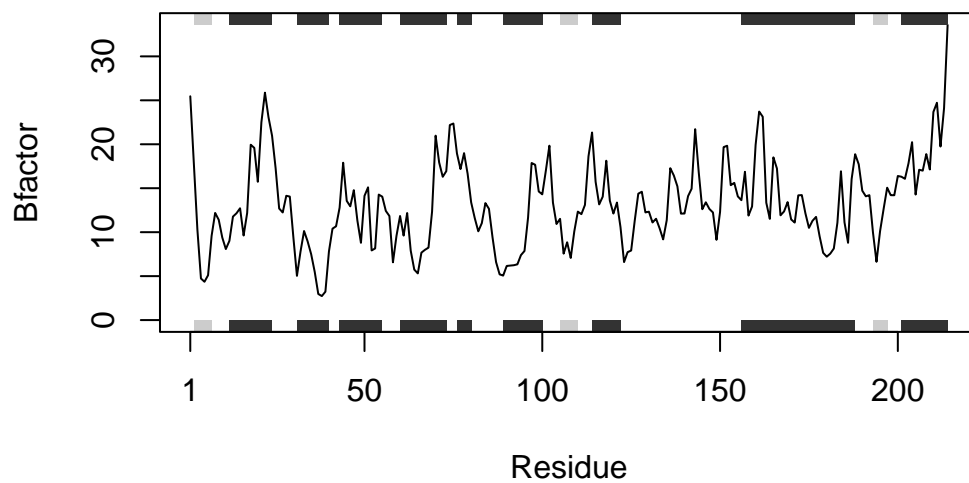
Note: Accessing on-line PDB file

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



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

Note: Accessing on-line PDB file



Q1.

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

```
s <- read.pdb("4AKE")
```

Note: Accessing on-line PDB file

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):  
C:\Users\micha\AppData\Local\Temp\Rtmp8AA5DW\4AKE.pdb exists. Skipping download
```

```
typeof(s)
```

```
[1] "list"
```

list

Q2.

What does the trim.pdb() function do?

```
?trim.pdb()
```

```
starting httpd help server ... done
```

trim.pdb() produces a new smaller PDB object, containing a subset of atoms. Here, trimmming to chain A.

Q3.

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

```
?plotb3
```

sse. they represent where the protein's secondary structures elements occur.

Q4.

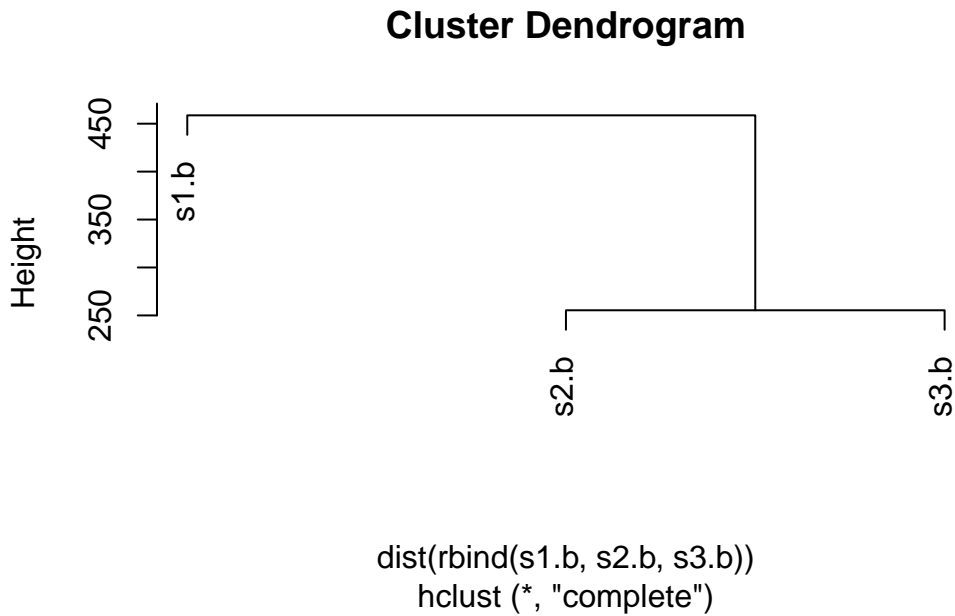
What would be a better plot to compare across the different proteins?

A cluster dendrogram.

Q5.

Which proteins are more similar to each other in their B-factor trends. How could you quantify this? HINT: try the `rbind()`, `dist()` and `hclust()` functions together with a resulting dendrogram plot. Look up the documentation to see what each of these functions does.

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



S2 and S3 are more similar