Class 06 HW

Serena Quezada (PID: A18556865)

Section 1: Improving analysis code by writing functions

A

scale a numeric vector to the [0,1] interval

```
scale01 <- function(x) {
  if (all(is.na(x))) return(x)
  (x-min(x, na.rm = TRUE)) /
    max(x, na.rm = TRUE) - min(x, na.rm = TRUE)
}
```

Build the data.frame

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

Apply the df$_to each column

```
df$a <- scale01(df$a)
df$b <- scale01 (df$b)
df$c <- scale01 (df$c)
df$d <- scale01 (df$d)
```

Check the results

```
print(df)
```

```
      a         b      c  d
1  -1.0 -200.0000 -11.00 NA
2  -0.9 -199.9444 -10.95 NA
3  -0.8 -199.8889 -10.90 NA
4  -0.7 -199.8333 -10.85 NA
5  -0.6 -199.7778 -10.80 NA
6  -0.5 -199.7222 -10.75 NA
7  -0.4 -199.6667 -10.70 NA
8  -0.3 -199.6111 -10.65 NA
9  -0.2 -199.5556 -10.60 NA
10 -0.1 -199.5000 -10.55 NA
```

B

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

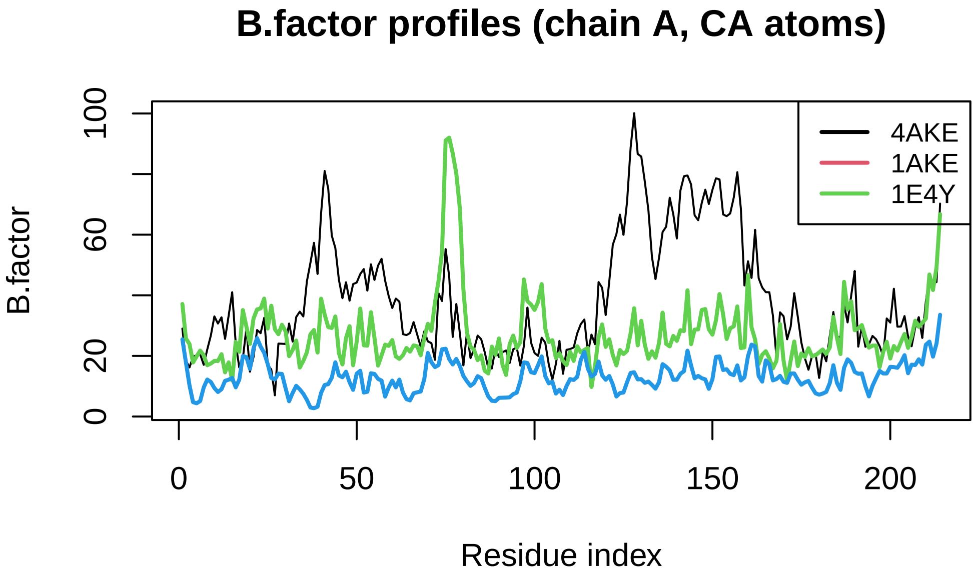
install new package (run once)

```r
options(repos = c(CRAN = "https://cloud.r-project.org"))
install.packages("bio3d")
```

```
The downloaded binary packages are in
    /var/folders/zk/6hldzf5n74scx33n9zx154500000gn/T//RtmpiSN8aP/downloaded_packages
```

```r
library(bio3d)

download.file (
  "https://files.rcsb.org/download/4AKE.pdb",
  destfile = "4AKE.pdb",
  quiet = TRUE)

# This reads, trims and extracts B-factors
get_bfactors <- function(pdb_ids,
                         chain = "A",
                         elety = "CA",
                         plot = TRUE) {

  #Helper that processes one structure
  proc_one <- function(pdb_id) {
    pdb   <- read.pdb(pdb_id)                    # download / read file
    trimmed <- trim.pdb(pdb, chain = chain, elety = elety)
    trimmed$atom$b                               # return B-factor vector
  }


  # Apply to every ID
  bf_list <- lapply(pdb_ids, proc_one)
  names(bf_list) <- pdb_ids


  # Plot one window for all
  if (plot) {
    max_len <- max(sapply(bf_list, length))
    plot(1:max_len,
         bf_list[[1]],
         type = "l",
         col = 1,
         xlab = "Residue index",
         ylab = "B-factor",
         ylim = range(unlist(bf_list), na.rm = TRUE),
         main = "B-factor profiles (chain A, CA atoms)")

    invisible(
      lapply(seq_along(bf_list)[-1], function(i) {
        lines(1:length(bf_list[[i]]), bf_list[[i]],
              col = i + 1, lwd = 2)
      })
    )
    legend("topright",
           legend = pdb_ids,
           col = 1:length(pdb_ids),
           lwd = 2,
           cex = 0.8)
  }
   invisible(bf_list)
}

# Use the function - works for any number of proteins

my_ids <- c("4AKE", "1AKE", "1E4Y")   # add as many as you like
bfactors <- get_bfactors(my_ids)
```

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


B.factor profiles (chain A, CA atoms)
B.factor
Residue index
4AKE
1AKE
1E4Y