Import SMARTTESTER datasets

Joao Marreiros and David Nora

2021-07-14 16:08:08

# Goal of the script

This script imports and merges all single TXT-files (strokes + sensors) produced with the Inotec Smarttester. The experiment involved 12 samples (3 samples from each 4 raw materials) which have been used in four cycles (0-250, 250-500, and 500-1000 strokes) respectively. The script will:

1. Read in the original TXT-files
2. Format and merge the data for each sample
3. Combine the data from the 12 samples into one
4. Write an XLSX-file and save an R object ready for further analysis in R

This script is an adapted from…

dir\_in <- "analysis\_inotec/raw\_data/"  
dir\_out <- "analysis\_inotec/derived\_data/"

Raw data must be located in “analysis\_inotec/raw\_data/”.  
Formatted data will be saved in “analysis\_inotec/derived\_data/”. The knit directory for this script is the project directory.

# Load packages

library(tidyverse)  
library(R.utils)  
library(openxlsx)  
library(tools)

# List all files and get names of the files

# List all CSV files in dir\_in  
TXT\_files <- list.files(dir\_in, pattern = "\\.txt$", recursive = TRUE, full.names = TRUE)  
  
# Extract sample names from paths  
samples\_names <- dirname(dirname(dirname(TXT\_files))) %>% # Path of folder 3 levels higher  
 basename() %>% # Name of folder 3 levels higher  
 unique() # Unique names

# Define sensors

sensors <- data.frame(mess = paste0("Messung", 1:5),   
 meas = c("Force", "Friction", "Depth", "Position", "Velocity"),   
 unit = c("N", "N", "mm", "mm", "mm/s"))

# Merge all files and format the data

# Create named list, 1 element for each sample  
sampl <- vector(mode = "list", length = length(samples\_names))   
names(sampl) <- samples\_names  
  
# For each sample  
for (s in seq\_along(samples\_names)) {  
   
 # Gets information through the path name and defines the cycle, raw material and   
 # contact material  
 folder <- paste0(samples\_names[s], "/") %>%   
 grep(TXT\_files, value = TRUE) %>%   
 dirname() %>%   
 dirname() %>%   
 unique() %>%   
 basename() %>%   
 strsplit(., "\_")   
   
 cycles <- sapply(folder, FUN = function(x) x[[3]])  
 # Defines the number of the first stroke per cycle based on the name from the folders  
 cycle\_start <- gsub("-.\*$", "", x = cycles) %>%   
 # Converts into numeric   
 as.numeric()  
   
 # Orders the cycles  
 order\_cycles <- order(cycle\_start)  
 cycle\_start <- cycle\_start[order\_cycles]  
 cycle\_start[1] <- 1  
 cycles <- cycles[order\_cycles]  
   
 # Takes the information about the contact material  
 cont\_mat <- sapply(folder, FUN = function(x) x[[2]]) %>%   
 unique()  
   
 # Takes the information about the raw material  
 raw\_mat <- ifelse(grepl("FLT", names(sampl)[s]), "Flint", "Lydite")  
  
   
 # Create named list, 1 element for each sensor ("Messung")  
 sampl[[s]] <- vector(mode = "list", length = nrow(sensors))  
 names(sampl[[s]]) <- sensors [["meas"]]  
   
 # For each sensor ("Messung")  
 for (m in seq\_along(sampl[[s]])) {  
   
 # Extract file names of all strokes for the given sensor  
 # Paste sample name and slash to avoid partial matching  
 s\_m <- paste0(samples\_names[[s]], "/") %>%   
 # Extract sample "s" from all files  
 grep(TXT\_files, value = TRUE) %>%   
 # Extract sensor "m" from sample "s"  
 grep(sensors[["mess"]][m], ., value = TRUE)   
   
 # Create named list, 1 element for each stroke bin  
 sampl[[s]][[m]] <- vector(mode = "list", length = length(cycles))  
 names(sampl[[s]][[m]]) <- cycles  
   
 # For each cycle  
 for (cy in seq\_along(sampl[[s]][[m]])) {  
   
 # Extract file names of all strokes for each cycle  
 s\_m\_cy <- grep(cycles[cy], s\_m, value = TRUE)  
   
 # Create named list, 1 element for each stroke  
 sampl[[s]][[m]][[cy]] <- vector(mode = "list", length = length(s\_m\_cy))  
 names(sampl[[s]][[m]][[cy]]) <- paste0("Stroke", seq\_along(s\_m\_cy))  
   
 # For each stroke  
 for (st in seq\_along(s\_m\_cy)) {  
   
 # Read in TXT file  
 sampl[[s]][[m]][[cy]][[st]] <- read.table(s\_m\_cy[st], skip = 4, sep = ";") %>%   
   
 # Add columns Step based on V2 and Stroke based on "st"  
 mutate(Step = V2/100000+1, Stroke = st -1 + cycle\_start[cy]) %>%   
   
 # Select columns stroke, step, V1  
 select(Stroke, Step, V1)  
   
 # Rename column V1 based on "m"  
 names(sampl[[s]][[m]][[cy]][[st]])[3] <- sensors[m, "meas"]   
 }  
   
 # rbind all files per cycle  
 sampl[[s]][[m]][[cy]] <- do.call(rbind, sampl[[s]][[m]][[cy]])  
 }  
   
 # rbind all cycles per sensor  
 sampl[[s]][[m]] <- do.call(rbind, sampl[[s]][[m]])  
 }  
   
 # rbind all sensors per sample  
 sampl[[s]] <- full\_join(sampl[[s]][[1]], sampl[[s]][[2]]) %>%   
 full\_join(sampl[[s]][[3]]) %>%   
 full\_join(sampl[[s]][[4]]) %>%  
 full\_join(sampl[[s]][[5]]) %>%   
 mutate(Sample = names(sampl)[s], Raw\_material = raw\_mat,   
 Contact\_material = cont\_mat) %>%  
   
 select(Sample, Raw\_material, Contact\_material, everything())  
}  
  
# rbind all samples   
sampl <- do.call(rbind, sampl)

# Save data

## Format name of output file

file\_out <- "sampl"

## Write to XLSX

write.xlsx(list(data = sampl, units = sensors), file = paste0(dir\_out, file\_out, ".xlsx"))

## Save R object

saveObject(sampl, file = paste0(dir\_out, file\_out, ".Rbin"))

Warning in gzfile(file, "wb"): cannot open compressed file 'analysis\_inotec/  
derived\_data/sampl.Rbin.tmp', probable reason 'No such file or directory'

Error in gzfile(file, "wb"): cannot open the connection

# sessionInfo() and RStudio version

sessionInfo()

R version 4.0.4 (2021-02-15)  
Platform: x86\_64-apple-darwin17.0 (64-bit)  
Running under: macOS Catalina 10.15.7  
  
Matrix products: default  
BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib  
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib  
  
locale:  
[1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
  
attached base packages:  
[1] tools stats graphics grDevices utils datasets methods   
[8] base   
  
other attached packages:  
 [1] openxlsx\_4.2.4 R.utils\_2.10.1 R.oo\_1.24.0 R.methodsS3\_1.8.1  
 [5] forcats\_0.5.1 stringr\_1.4.0 dplyr\_1.0.7 purrr\_0.3.4   
 [9] readr\_1.4.0 tidyr\_1.1.3 tibble\_3.1.2 ggplot2\_3.3.5   
[13] tidyverse\_1.3.1   
  
loaded via a namespace (and not attached):  
 [1] tidyselect\_1.1.1 xfun\_0.24 haven\_2.4.1 colorspace\_2.0-2   
 [5] vctrs\_0.3.8 generics\_0.1.0 htmltools\_0.5.1.1 yaml\_2.2.1   
 [9] utf8\_1.2.1 rlang\_0.4.11 pillar\_1.6.1 glue\_1.4.2   
[13] withr\_2.4.2 DBI\_1.1.1 dbplyr\_2.1.1 modelr\_0.1.8   
[17] readxl\_1.3.1 lifecycle\_1.0.0 munsell\_0.5.0 gtable\_0.3.0   
[21] cellranger\_1.1.0 zip\_2.2.0 rvest\_1.0.0 evaluate\_0.14   
[25] knitr\_1.33 fansi\_0.5.0 broom\_0.7.8 Rcpp\_1.0.6   
[29] scales\_1.1.1 backports\_1.2.1 jsonlite\_1.7.2 fs\_1.5.0   
[33] hms\_1.1.0 digest\_0.6.27 stringi\_1.6.2 grid\_4.0.4   
[37] cli\_2.5.0 magrittr\_2.0.1 crayon\_1.4.1 pkgconfig\_2.0.3   
[41] ellipsis\_0.3.2 xml2\_1.3.2 reprex\_2.0.0 lubridate\_1.7.10   
[45] assertthat\_0.2.1 rmarkdown\_2.9 httr\_1.4.2 rstudioapi\_0.13   
[49] R6\_2.5.0 compiler\_4.0.4

END OF SCRIPT