Read cytof data into R

Anja Bråthen Kristoffersen

22 March 2022

This document describes the different lines of the file “readDataToAnalyse.R”. To adjust to your own data save the file “readDataToAnalyse.R” from github <https://github.com/folkehelseinstituttet/cytof> give it a new name and change the necessary paths and values throughout the document to use it on your own data. You also want to save the scripts: “read\_data\_functions.R”, “transformation\_functions.R”, “ploting\_functions.R”, “clustering\_functions.R”, “gating\_functions.R”, “analysis\_functions.R” in the same folder.

You start with defining a variable data\_path that contains the path of the folder where all your cleaned files are saved, one path to the folder where your script are saved, one path to the folder where you want your results to end up and one path to the folder of your metadata. I prefer using the function path from the library fs. When using a function from a given library the way to get the function is by library\_name::function\_name so in this case fs::path. By writing “#” as the first character in a line in R, you start a comment. Here you could write whatever you want it will not be read by R when running the file.

#data\_path <- fs::path("F:", "Forskningsprosjekter", "PDB 2794 - Immune responses aga\_", "Forskningsfiler", "JOBO", "CyTOF","Gating", "Gating fra R\_FINAL")  
data\_path <- fs::path("C:", "CyToF data", "immun\_aga", "panel1")  
#scriptPath <- fs::path("C:", "CyToF data", "fra github")  
scriptPath <- fs::path("H:", "git", "cytof")  
  
outPath <- fs::path("F:", "Forskningsprosjekter", "PDB 2794 - Immune responses aga\_", "Forskningsfiler", "JOBO", "CyTOF", "Analyse i R OUS", "Resultat\_Panel\_1")  
metaDataPath <- fs::path("F:", "Forskningsprosjekter", "PDB 2794 - Immune responses aga\_", "Forskningsfiler", "JOBO", "CyTOF", "Analyse i R OUS")

Then you want to source the different functions that you later will use in to R.

source(fs::path(scriptPath, "read\_data\_functions.R"))  
source(fs::path(scriptPath, "transformation\_functions.R"))  
source(fs::path(scriptPath, "ploting\_functions.R"))  
source(fs::path(scriptPath, "clustering\_functions.R"))  
source(fs::path(scriptPath, "gating\_functions.R"))  
source(fs::path(scriptPath, "analysis\_functions.R"))

Read the matrix posNeg into R. If you have not made this matrix jet, or do not want to make it you could skip this part.

posNegFilnavn <- as.character(readRDS(fs::path(outPath, "Data", "posNegFilnavn.rds")))  
posNeg <- readRDS(fs::path(outPath, "Data", "posNeg.rds"))

This section is especally designed to the data I am analysing now. This have to be designed for your metadata, preferably by only reading a file and then make sure that the order of the rows in that file follows the same order as the posNeg file and later the same order as the cleaned fcs file.

status <- rep("Moderat", length(posNegFilnavn))  
status[grepl("S\_", posNegFilnavn)] <- "Severe"  
status[grepl("Ref", posNegFilnavn)] <- "Ref"  
  
age <- rep(NA, length(posNegFilnavn))  
for(i in 1:length(posNegFilnavn)){  
 age[i] <- strsplit(as.character(posNegFilnavn[i]), "\_")[[1]][4]  
}  
age <- as.numeric(age)  
  
  
sex <- rep("Male", length(posNegFilnavn))  
sex[grepl("Fe", posNegFilnavn)] <- "Female"  
sex[grepl("Ref1", posNegFilnavn)] <- "Ref"  
  
  
dInfo <- data.frame(status = status, age = age, sex = sex, filenames= posNegFilnavn)  
rownames(dInfo) <- dInfo$filenames  
#dInfo$x <- jitter(as.numeric(dInfo$status))  
  
  
tid <- read.csv(fs::path(metaDataPath, "tid.csv"), sep = ";")  
colnames(tid) <- c("fil", "tid")  
tid$pasient <- gsub("T1", "", tid$fil)  
tid$pasient <- gsub("T2", "", tid$pasient)  
  
  
dInfo$tid <- NA  
for(i in 1:nrow(tid)){  
 dInfo$tid[grep(as.character(tid$pasient[i]), dInfo$filenames)] <- as.character(tid$tid[i])  
}  
  
dInfo$tid[grep(as.character("FHI81"), dInfo$filenames)] <- "28.06.2021"  
dInfo$tid[grep(as.character("FHI95"), dInfo$filenames)] <- "28.06.2021"

# read all files in data\_path into one dataset fcs\_data

Here I make sure that the files are read in the same order as the posNeg files by setting files\_to\_open to be the variable posNegFilnavn. If you have not made or will not make the matrix posNeg you should make sure that the fcs data is read in the same order as your metadata by adjusting the files\_to\_open accordently. fcs\_data\_with\_info will be a list with two different variable types. I want them separated, since this is a large variable it is important to remove fcs\_data\_with\_info after assigning to two different variables by using the function rm().

fcs\_data\_with\_info <- read\_specific\_data\_from\_folder(data\_path = data\_path, files\_to\_open = paste0(as.character(posNegFilnavn), ".fcs"))  
fcs\_data <- fcs\_data\_with\_info$fcs\_data  
file\_names <- fcs\_data\_with\_info$file\_names  
rm(fcs\_data\_with\_info)

This section is only to get the names of the channels that you want to use further in the analysis.

params\_fcs <- get\_params\_fcs\_data(fcs\_data[[1]])  
  
kanaler <- params\_fcs$desc[grepl("\_",params\_fcs$desc)]  
kanaler <- kanaler[!grepl("\_DNA", kanaler)]  
kanaler <- kanaler[!grepl("\_Cisp", kanaler)]  
kanalnavn <- params\_fcs$name[params\_fcs$desc %in% kanaler]  
print("antall markører med")  
length(kanalnavn)