Package Documentations-normfluodbf

NORM\_TIDY\_DBF (SHOULD HAVE BEEN NORM\_DIRTY\_DBF)

Title: Cleans and Normalizes ".dbf" data files obtained from experiments using the FLUOstar Omega (Ω) microplate reader (from [BMG LABTECH](https://www.bmglabtech.com/en/about-us/)).

Description: The simplest function utilization scenario entails an input of the path to a ".dbf" file obtained from the FLUOstar microplate (usually a 96-well microplate) reader; In a single step, this function will create a data frame, clean the data frame, normalize the data frame, append a "Cycle\_Number" attribute, perform an adjustment to the “time” attribute and return a data frame that is ready for analysis.

Since the initial publication of this package, several changes have been made to improve the user experience and to give the user more options to fine-tune the output from the package to meet the users’ aesthetic needs. Users who decide to move past the simplest utility scenario have been given more options to customize the output based on the users’ needs. Notably, several normalization sub-parameters have been provided in the package which yields different outputs based on what the user is used to seeing.

Just as the FLUOstar instrument is built to handle an array of assays, this function is designed to be multi-dimensional (meaning it can handle data with the same DBF extension from other assay types), on the condition that the data from assay types other than liposome flux assays follow the same data format this package was designed to handle. Of course, users of this package are advised to pre-analyze DBF files from other assay types to ensure they are compliant with this package (compliance in this scenario is simple meaning DBF files from other assays should be like DBF files from liposome flux assays).

Parameters: Not placed here. No need to eat up space. Proofread in the package.

Note:

* The default normalization sub-parameter outputs values in the 0-1 range. Unless a “norm\_scale” level is specified by the user, the default output is in the 0-1 range.
* The “norm\_scale” sub-parameter “decimal” is a machine-learning tool and should be avoided; it also provides no advantage for basic research analysis as its output operates on a sliding scale just like the raw data.
* Logarithmic transformation provides a minuscule advantage in data analysis and could/should be avoided.
* Backward compatibility is maintained in all updates, so there should be no issues with using the package the way the user was used to.
* The last NOTE is that if the user has no real clue how to analyze their data and just inputs a link to the dbf file, the program returns a 0-1 scale normalized data frame.
* Example (Check out in a more relaxing state as I enjoy the learning process)
* The favorite "norm\_scale" level is "z-score" since it divides the axis into negative and positive, thus facilitating interpretation.

Unique\_identifier function

Title: A function to append a unique identifier attribute to any data frame within the normfluodbf package.

Description: The function in the context of normfluodbf creates an attribute called Cycle\_Number and appends this attribute to the cleaned or wrangled data frame derived from the dirty DBF file.

Param: df A data frame with n number of rows

Return: A data frame with the Cycle\_Number attribute appended to the end of the data frame.

Note: The function operates in a closed system, meaning it is primarily designed to work with this package ONLY. Other use cases are simply a coincidence.

Generic Identifier function

Title: A function to append a generic identifier attribute to any data frame, but users supply a name for said attribute.

Description: A function that creates an attribute of seq(numrows) with a step size of 1, where the user provides the attribute name.

Param: The number of rows the user intends to have in the created data frame.

Param: The desired attribute name.

Return: A user-named single attribute data frame with nrow = numrows.

@examples generic\_identifier (40, col\_name="Cycle\_No")

Min Max function

Title: Min-Max normalization on a 0-1 scale.

#' @param x Attribute value(s)

#' @return A normalized value (between 0 and 1) when applied to a single value or a normalized attribute with values between 0 and 1.

#' @note The lapply function is required to apply this function across several attributes.

Min Max Percent function

Title: Min-Max normalization on a 0-100 scale.

#' @param x Attribute value(s)

#' @return A normalized value (between 0 and 100) when applied to a single value or a normalized attribute with values between 0 and 100.

#' @note The lapply function is required to apply this function across several attributes.

Norm z function

#' Title: Z-score standardization or normalization function

#' @param x Attribute value(s)

#' @return A standardized value (Z = N (0,1)) when applied to a single value or a standardized attribute with values (Z = N (0,1)).

#' @note The lapply function is required to apply this function across several attributes.

Decimal scaling function

Title: A decimal scaling function (a machine learning tool).

@param x Attribute value(s)

#' @return A decimal scaled value when applied to a single value or a decimal scaled attribute(s).

#' @note The lapply function is required to apply this function across several attributes.

#' This is NOT a normalization function, so data obtained from the decimal scaling function exists on a sliding scale and SHOULD NOT be used for meaningful analysis.

Log transformation function

Title: A log transformation function.

@param x Attribute value(s)

#' @return A log-transformed value when applied to a single value or an attribute with log-transformed values.

#' @note The lapply function is required to apply this function across several attributes.

#' This is NOT a normalization function, so data obtained from the decimal scaling function exists on a sliding scale and SHOULD NOT be used for meaningful analysis.

Round fluor function

Title: A value rounding function.

@param x Attribute value(s).

#' @return A rounded value with three decimal places when applied to a single value or an attribute with log-transformed values.

Clean odd cc function

Title: DAT file data frame cleaner.

Description:

The function takes the dirty data frame obtained from reading the FLUOstar DAT file and applies a function called comma\_cleaner () to the dirty data frame, which automatically inserts NAs in place of the special characters, and rows with NAs only are removed.

#' @param df A dirty data frame obtained from the FLUOstar DAT file.

#' @return A clean data frame with clean NA values retained.

Clean odddat optimus function

Title: DAT file data frame cleaner.

Description:

The function takes the dirty data frame obtained from reading the FLUOstar DAT file, applies an original algorithm that inserts NAs in place of the special characters, and then applies a function called comma\_cleaner() to the dirty data frame for the removal of commas, and finally, rows with NAs only are removed.

#' @param df A dirty data frame obtained from the FLUOstar DAT file.

#' @return A clean data frame with clean NA values retained.

Comma cleaner function

Title: Comma Cleaner function.

Description: This modular function, in the context of this package, is responsible for removing commas from attribute(s) values. Removal of commas facilitates the conversion of attributes into the numeric class.

#' @param df A dirty data frame obtained from the FLUOstar DAT file.

#' @return A clean data frame with numeric no-comma values for attribute(s).

Dat col names\_optimus function

Title: Attribute(s) naming function.

Description: This function is used to name attribute(s). Attribute(s) names, in this case, are equivalent to the well labels found on the microplate reader. An attribute for a sample loaded into row A - column 1 will be named A1. In short, the function takes a clean data frame and returns attribute names that match the FLUOstar plate layout often presented as an Excel file.

#' @param df A data frame that requires attribute labels.

#' @param rows\_used A character vector indicating the rows or tuples used on the microplate (usually a 96-well microplate). Initialized as NULL.

#' @param cols\_used A numeric vector indicating the plate columns or attributes used. Initialized as NULL.

#' @param user\_specific\_labels A character vector where the user manually enters the used microplate wells based on the FLUOstar plate layout.

#' @param read\_direction A string input with two choices, “vertical” or “horizontal.” The user indicates “vertical” if the user intends to have a final data frame with samples arranged as sample type triplets (A1, B1, C1, A1, B1, C1) OR “horizontal” if the user intends to have a final data frame with samples arranged as clusters per sample type (A1, A2, A3, B1, B2, B3).

#' @return Returns a character or numeric vector of attribute(s) names for the normalized data frame.

#' @note Users are advised to input rows used but won’t be penalized for not doing so. If the user provides the rows used, then attribute names are generated for the user. The user must check to ensure that the names match the microplate layout.

The user can leave the columns used as NULL if the user loaded samples from column 1 and did so in sequence. If the user fails to load in sequence from the first position, then the user must provide a numeric vector of columns used.

For instance, where the user skips columns, the user will be prompted to interact with the program in order to ensure the final data frame has the correct attribute names.

The user can bypass the rows used and columns used parameters if the user supplies a manually created character vector of the wells used in an experiment.

The read direction parameter is used to determine the presentation of the samples in the final data frame.

Title: A fluorescence quantification Quality Control (QC) function.

description: A function designed to check that fluorescence values do not exceed the upper limit (2^15 or 32768) OR fall below the lower limit (2^11 or 2048). Fluorescence values that exceed these thresholds are considered noisy and lead to incorrect interpretation of analysis results.

#' @param clean\_df A cleaned data frame.

#' @param fun A parameter used for Boolean expressions.

return: A polite warning message to the data analyst or researcher.

Notes:

Experimental issues should be investigated at very high or very low fluorescence values.

The most common experimental issues arise when ACMA concentrations are out of the tolerated range. Based on my experience, ACMA concentrations between 2 and 5 Micromolar will suffice to get fluorescence values within the tolerance threshold. ACMA concentrations as low as 0.2 Micromolar or as high as 20 Micromolar have proven problematic based on my research experience.

A second issue linked to the FLUOstar instrument revolves around setting the right “gain” to ensure the right level of sensitivity in machine readings. A very high “gain” setting results in increased machine sensitivity even at the right ACMA concentrations and vice versa. In short, we want the machine to be primed to read exactly what we feed it, no more, no less.

This function provides the attribute(s) and tuple(s) for the values that need investigation.

These deductions were obtained from my experimental hiccups and my characterization of the liposome flux assay system.

I have a slide deck on the characterization of the liposome flux assay system.

#' fpath <- system.file("extdata", "dat\_1.dat", package = "normfluodbf", mustWork = TRUE)

#' dat\_df <- read.table(file=fpath)

#' nocomma\_dat <- clean\_odd\_cc(dat\_df)

#' resampled\_scaled <- resample\_dat\_scale(nocomma\_dat, tnp=3, cycles=40)

#' resampled\_scaled <- resampled\_scaled[,c(1:4)]

#' fluor\_threshold\_check(resampled\_scaled)

#' Title: Cleans normalizes DAT files obtained from experiments using the FLUOstar Omega microplate reader (from BMG LABTECH).

Description:

#' The simplest case scenario entails inputting the name or directory of a DAT file as a string, the number of rows denoted by the tnp (test, negative, positive) parameter, and the number of cycles (selected by the user when running the FLUOstar instrument). The program takes these three baseline parameters, performs cleaning and normalization of the DAT file, and then appends an attribute called “Cycle\_Number” to the normalized data frame.

#' @author Tingwei Adeck

#' @param A string ("dat\_1.dat") if the file is found within the present working directory (pwd) OR a path pointing directly to a ".dat" file.

#' @param tnp A numeric value indicating the number of rows used. TNP is used as an acronym for Test, Negative, Positive.

#' @param cycles A numeric value indicating the number of cycles selected by the user when running the FLUOstar instrument.

#' @param rows\_used A character vector of the rows used.

#' @param cols\_used A numeric vector of the columns used.

#' @param user\_specific\_labels A character vector manually prepared by the user to denote the wells used on the microplate reader.

#' @param read\_direction A string input with two choices, “vertical” or “horizontal.” The user indicates “vertical” if the user intends to have a final data frame with samples arranged as sample type triplets (A1, B1, C1, A1, B1, C1) OR “horizontal” if the user intends to have a final data frame with samples arranged as clusters per sample type (A1, A2, A3, B1, B2, B3).

@param norm\_scale This parameter takes sub-parameters: 'raw’ , hundred’ , 'one’ , 'z-score' , or 'decimal’ , which denotes the normalization type or scale; Initialized as NULL.

#' @return A normalized data frame with an appended "Cycle\_Number" attribute. The “Cycle\_Number” attribute is the x-variable.

#' @examples fpath <- system.file("extdata", "dat\_1.dat", package = "normfluodbf", mustWork = TRUE)

#' n <- c('A','B','C')

#' normalized\_fluo\_dat <- normfluodat(dat=fpath, tnp = 3, cycles = 40, n, read\_direction = 'vertical')

#' normalized\_fluo\_dat <- normfluodat(dat=fpath, tnp = 3, cycles = 40, n, read\_direction = 'vertical')

#' #' normalized\_fluo\_dat <- normfluodat(dat=fpath, tnp = 3, cycles = 40)

#' Title: A function to create an attribute or column for each sample loaded into the microplate wells.

#' @description: Designed as a prototype function to take a single attribute or column consisting of several samples and perform a putative resampling to yield another data frame with a separate attribute for each sample.

#' @author Tingwei Adeck

#' @param df A clean data frame with attributes or tuples containing a mixture of samples.

#' @param tnp A numeric value indicating the number of rows used. TNP is used as an acronym for Test, Negative, Positive.

#' @param cycles A numeric value indicating the number of cycles selected by the user when running the FLUOstar instrument.

#' @return A new data frame where separated samples are assigned a separate attribute or column.

#' @examples fpath <- system.file("extdata", "dat\_1.dat", package = "normfluodbf", mustWork = TRUE)

#' dat\_df <- read.table(file=fpath)

#' nocomma\_dat <- clean\_odd\_cc(dat\_df)

#' col\_1 <- nocomma\_dat[,1]

#' col\_1 <- as.data.frame(col\_1)

#' samples\_delineated <- resample\_dat(col\_1, tnp=3, cycles=40)

#' Title: A function to create an attribute or column for each sample loaded into the microplate wells.

#' @description: Designed as a prototype function to take a single tuple or row consisting of several samples and perform a putative resampling to yield another data frame with a separate attribute for each sample.

#' @author Tingwei Adeck

#' @param df A clean data frame with attributes or tuples containing a mixture of samples.

#' @param tnp A numeric value indicating the number of rows used. TNP is used as an acronym for Test, Negative, Positive.

#' @param cycles A numeric value indicating the number of cycles selected by the user when running the FLUOstar instrument.

#' @return A new data frame where separated samples are assigned a separate attribute or column.

#' @examples fpath <- system.file("extdata", "dat\_1.dat", package = "normfluodbf", mustWork = TRUE)

#' dat\_df <- read.table(file=fpath)

#' nocomma\_dat <- clean\_odd\_cc(dat\_df)

#' col\_1 <- nocomma\_dat[,1]

#' col\_1 <- as.data.frame(col\_1)

#' samples\_delineated <- resample\_dat\_alt(col\_1, tnp=3, cycles=40)