Package ‘NACHO’

January 9, 2020

Type       Package
Title      NanoString Quality Control Dashboard
Version    1.0.0
Description NanoString nCounter data are gene expression assays
where there is no need for the use of enzymes or amplification
protocols and work with fluorescent barcodes (Geiss et al. (2018)
<doi:10.1038/nbt1385>). Each barcode is assigned a
messenger-RNA/micro-RNA (mRNA/miRNA) which after bonding with its
target can be counted. As a result each count of a specific barcode
represents the presence of its target mRNA/miRNA. ‘NACHO’ (NAnoString
quality Control dasHbOard) is able to analyse the exported NanoString
nCounter data and facilitates the user in performing a quality
control. ‘NACHO’ does this by visualising quality control metrics,
expression of control genes, principal components and sample specific
size factors in an interactive web application.

License  GPL-3 | file LICENSE

URL https://github.com/mcanouil/NACHO,
       https://mcanouil.github.io/NACHO

BugReports https://github.com/mcanouil/NACHO/issues

Depends  R (>= 3.5.0)
Imports   cli (>= 1.1.0), crayon (>= 1.3.4), dplyr (>= 0.8.3),
          ggbeeswarm (>= 0.6.0), ggforce (>= 0.3.1), ggplot2 (>= 3.2.1),
          ggrepel (>= 0.8.1), knitr (>= 1.25), magrittr (>= 1.5), purrr
          (>= 0.3.3), rlang (>= 0.1.2), rmarkdown (>= 1.16), rstudioapi
          (>= 0.10), scales (>= 1.0.0), sessioninfo (>= 1.1.1), shiny (>=
          1.4.0), shinyWidgets (>= 0.4.9), tibble (>= 2.1.3), tidyR (>=
          1.0.0), utils
Suggests  Biobase, covr (>= 3.3.2), GEOquery, limma, roxygen2 (>=
          7.0.2), testthat (>= 2.2.1)

VignetteBuilder knitr
Encoding  UTF-8
LazyData true
** autoplot.nacho  

** Plot quality-control metrics and thresholds of a "nacho" object **

** Description **

This function allows to plot any quality-control figures available within the shiny app using `visualise()` or in the HTML report from `render()`.

** Usage **

```r
## S3 method for class 'nacho'
autoplot(
  object,
  x,
  colour = "CartridgeID",
  size = 0.5,
  show_legend = TRUE,
  show_outliers = TRUE,
  outliers_factor = 1,
  outliers_labels = NULL,
  ...
)
```
Arguments

- **object** [list] List obtained from `load_rcc()` or `normalise()`.
- **x** [character] Character string naming the quality-control metrics to plot from `nacho_object`. The possible values are:
  - "BD" (Binding Density)
  - "FoV" (Imaging)
  - "PCL" (Positive Control Linearity)
  - "LoD" (Limit of Detection)
  - "Positive" (Positive Controls)
  - "Negative" (Negative Controls)
  - "Housekeeping" (Housekeeping Genes)
  - "PN" (Positive Controls vs. Negative Controls)
  - "ACBD" (Average Counts vs. Binding Density)
  - "ACMC" (Average Counts vs. Median Counts)
  - "PCA12" (Principal Component 1 vs. 2)
  - "PCAi" (Principal Component scree plot)
  - "PCA" (Principal Components planes)
  - "PFNF" (Positive Factor vs. Negative Factor)
  - "HF" (Housekeeping Factor)
  - "NORM" (Normalisation Factor)

- **colour** [character] Character string of the column in `sheet_csv` or more generally in `nacho_object$nacho` to be used as grouping colour.
- **size** [numeric] A numeric controlling point size (ggplot2::geom_point() or #’ggbeeswarm::geom_beeswarm() or line size (ggplot2::geom_line())).
- **show_legend** [logical] Boolean to indicate whether the plot legends should be plotted (TRUE) or not (FALSE). Default is TRUE.
- **show_outliers** [logical] Boolean to indicate whether the outliers should be highlighted in red (TRUE) or not (FALSE). Default is TRUE.
- **outliers_factor** [numeric] Size factor for outliers compared to size. Default is 1.
- **outliers_labels** [character] Character to indicate which column in `nacho_object$nacho` should be used to be printed as the labels for outliers or not. Default is NULL.

... Other arguments (Not used).

Examples

data(GSE74821)

autoplot(GSE74821, x = "BD")
check_outliers  

Annotate a "nacho" object for outliers

Description

Add or update "is_outlier" column in the "nacho" field of an object from a call to load_rcc() or normalise() (nacho_object$nacho), using the current quality-control thresholds.

Usage

check_outliers(nacho_object)

Arguments

nacho_object  

[object] A list object of class "nacho" obtained from load_rcc() or normalise().

Value

A [object] object of class "nacho".

Examples

data(GSE74821)
nacho_object <- check_outliers(GSE74821)
head(nacho_object$nacho)

deploy  

Deploy (copy) the shiny application to the specified directory

Description

Deploy (copy) the shiny application to the specified directory

Usage

deploy(directory = "/srv/shiny-server", app_name = "NACHO")

Arguments

directory  

[object] A character vector of one path to the new location.

app_name  

[object] A character vector defining the shiny application name in the new location.
Value

logical A logical indicating whether the deployment is successful (TRUE) or not (FALSE).

Examples

deploy(directory = ".")
if (interactive()) {
    shiny::runApp("NACHO")
}

GSE74821 A "nacho" object containing 20 samples of GSE74821 dataset

Description

NanoString nCounter RUO-PAM50 Gene Expression Custom CodeSet

Usage

GSE74821

Format

A [list] object of class "nacho".

Source

GSE74821

load_rcc Produce a "nacho" object from RCC NanoString files

Description

This function is used to preprocess the data from NanoString nCounter.
Usage

`load_rcc(
    data_directory,
    ssheet_csv,
    id_colname,
    housekeeping_genes = NULL,
    housekeeping_predict = FALSE,
    housekeeping_norm = TRUE,
    normalisation_method = "GEO",
    n_comp = 10
)
`

Arguments

data_directory  [character] A character string of the directory where the data are stored.
ssheet_csv       [character] or [data.frame] Either a string with the name of the CSV of the samplesheet or the samplesheet as a data.frame. Should contain a column that matches the file names in the folder.
id_colname       [character] Character string of the column in ssheet_csv that matches the file names in data_directory.
housekeeping_genes [character] A vector of names of the miRNAs/mRNAs that should be used as housekeeping genes. Default is NULL.
housekeeping_predict [logical] Boolean to indicate whether the housekeeping genes should be predicted (TRUE) or not (FALSE). Default is FALSE.
housekeeping_norm [logical] Boolean to indicate whether the housekeeping normalisation should be performed. Default is TRUE.
normalisation_method [character] Either "GEO" or "GLM". Character string to indicate normalisation using the geometric mean ("GEO") or a generalized linear model ("GLM"). Default is "GEO".
n_comp            [numeric] Number indicating the number of principal components to compute. Cannot be more than n-1 samples. Default is 10.

Value

list  A list object of class "nacho":

  access  [character] Value passed to `load_rcc()` in id_colname.
  housekeeping_genes  [character] Value passed to `load_rcc()`.
  housekeeping_predict [logical] Value passed to `load_rcc()`.
  housekeeping_norm [logical] Value passed to `load_rcc()`.
  normalisation_method [character] Value passed to `load_rcc()`.
  remove_outliers [logical] FALSE.
load_rcc

n_comp [numeric] Value passed to load_rcc().
data_directory [character] Value passed to load_rcc().

pc_sum [data.frame] A data.frame with n_comp rows and four columns: “Standard deviation”, “Proportion of Variance”, “Cumulative Proportion” and “PC”.
nacho [data.frame] A data.frame with all columns from the sample sheet ssheet.csv and all computed columns, i.e., quality-control metrics and counts, with one sample per row.
outliers_thresholds [list] A list of the (default) quality-control thresholds used.
raw_counts [data.frame] Raw counts with probes as rows and samples as columns. With "CodeClass" (first column), the type of the probes and "Name" (second column), the Name of the probes.
normalised_counts [data.frame] Normalised counts with probes as rows and samples as columns. With "CodeClass" (first column), the type of the probes and "Name" (second column), the name of the probes.

Examples

if (interactive()) {
  library(GEOquery)
  library(NACHO)
  # Import data from GEO
  gse <- GEOquery::getGEO(GEO = "GSE74821")
  targets <- Biobase::pData(Biobase::phenoData(gse[[1]]))
  GEOquery::getGEOSuppFiles(GEO = "GSE74821", baseDir = tempdir())
  utils::untar(
    tarfile = paste0(tempdir(), "/GSE74821/GSE74821_RAW.tar"),
    exdir = paste0(tempdir(), "/GSE74821")
  )
  targets$IDFILE <- list.files(
    path = paste0(tempdir(), "/GSE74821"),
    pattern = ".RCC.gz$"
  )
  targets[] <- lapply(X = targets, FUN = iconv, from = "latin1", to = "ASCII")
  utils::write.csv(
    x = targets,
    file = paste0(tempdir(), "/GSE74821/Samplesheet.csv"
  )
  )
  # Read RCC files and format
  nacho <- load_rcc(
    data_directory = paste0(tempdir(), "/GSE74821"),
    ssheet_csv = paste0(tempdir(), "/GSE74821/Samplesheet.csv"),
    id_colname = "IDFILE"
  )
}
nacho_conflicts

Conflicts between the NACHO and other packages

Description

This function lists all the conflicts between packages in the NACHO and other packages that you have loaded.

Usage

nacho_conflicts()

Details

There are four conflicts that are deliberately ignored: intersect(), union(), setequal(), and setdiff() from dplyr. These functions make the base equivalents generic, so shouldn’t negatively affect any existing code.

Examples

nacho_conflicts()

normalise

(re)Normalise a "nacho" object

Description

This function creates a list in which your settings, the raw counts and normalised counts are stored, using the result from a call to load_rcc().

Usage

normalise(
  nacho_object,
  housekeeping_genes = nacho_object[["housekeeping_genes"]],
  housekeeping_predict = nacho_object[["housekeeping_predict"]],
  housekeeping_norm = nacho_object[["housekeeping_norm"]],
  normalisation_method = nacho_object[["normalisation_method"]],
  n_comp = nacho_object[["n_comp"]],
  remove_outliers = nacho_object[["remove_outliers"]],
  outliers_thresholds = nacho_object[["outliers_thresholds"]]
)
Arguments

nacho_object  [list] A list object of class "nacho" obtained from `load_rcc()` or `normalise()`.
housekeeping_genes
[character] A vector of names of the miRNAs/mRNAs that should be used as housekeeping genes. Default is NULL.
housekeeping_predict
[logical] Boolean to indicate whether the housekeeping genes should be predicted (TRUE) or not (FALSE). Default is FALSE.
housekeeping_norm
[logical] Boolean to indicate whether the housekeeping normalisation should be performed. Default is TRUE.
normalisation_method
[character] Either "GEO" or "GLM". Character string to indicate normalisation using the geometric mean ("GEO") or a generalized linear model ("GLM"). Default is "GEO".
n_comp
[numeric] Number indicating the number of principal components to compute. Cannot be more than n-1 samples. Default is 10.
remove_outliers
[logical] A boolean to indicate if outliers should be excluded.
outliers_thresholds
[list] List of thresholds to exclude outliers.

Details

Outliers definition (remove_outliers = TRUE):

- Binding Density (BD) < 0.1
- Binding Density (BD) > 2.25
- Field of View (FoV) < 75
- Positive Control Linearity (PCL) < 0.95
- Limit of Detection (LoD) < 2
- Positive normalisation factor (Positive_factor) < 0.25
- Positive normalisation factor (Positive_factor) > 4
- Housekeeping normalisation factor (house_factor) < 1/11
- Housekeeping normalisation factor (house_factor) > 11

Value

list  A list containing parameters and data.

access  [character] Value passed to `load_rcc()` in id_colname.
housekeeping_genes  [character] Value passed to `load_rcc()` or `normalise()`.
housekeeping_predict  [logical] Value passed to `load_rcc()`.
housekeeping_norm  [logical] Value passed to `load_rcc()` or `normalise()`.
normalisation_method [character] Value passed to load_rcc() or normalise().

remove_outliers [logical] Value passed to normalise().

n_comp [numeric] Value passed to load_rcc().

data_directory [character] Value passed to load_rcc().

pc_sum [data.frame] A data.frame with n_comp rows and four columns: "Standard deviation", "Proportion of Variance", "Cumulative Proportion" and "PC".

nacho [data.frame] A data.frame with all columns from the sample sheet ssheet_csv and all computed columns, i.e., quality-control metrics and counts, with one sample per row.

outliers_thresholds [list] A list of the quality-control thresholds used.

raw_counts [data.frame] Raw counts with probes as rows and samples as columns. With "CodeClass" (first column), the type of the probes and "Name" (second column), the Name of the probes.

normalised_counts [data.frame] Normalised counts with probes as rows and samples as columns. With "CodeClass" (first column), the type of the probes and "Name" (second column), the name of the probes.

Examples

data(GSE74821)
GSE74821_norm <- normalise(
  nacho_object = GSE74821,
  housekeeping_norm = TRUE,
  normalisation_method = "GEO",
  remove_outliers = TRUE
)

if (interactive()) {
  library(GEOquery)
  library(NACHO)

  # Import data from GEO
  gse <- GEOquery::getGEO(GEO = "GSE74821")
  targets <- Biobase::pData(Biobase::phenoData(gse[[1]]))
  GEOquery::getGEOSuppFiles(GEO = "GSE74821", baseDir = tempdir())
  utils::untar(
    tarfile = paste0(tempdir(), "/GSE74821/GSE74821_RAW.tar"),
    exdir = paste0(tempdir(), "/GSE74821"))
  targets$IDFILE <- list.files(
    path = paste0(tempdir(), "/GSE74821"),
    pattern = ".RCC.gz$"
  )
  targets[] <- lapply(X = targets, FUN = iconv, from = "latin1", to = "ASCII")
  utils::write.csv(
    x = targets,
    file = paste0(tempdir(), "/GSE74821/Samplesheet.csv"
  )

  # Read RCC files and format
nacho <- load_rcc(
  data_directory = paste0(tempdir(), "/GSE74821"),
  ssheet_csv = paste0(tempdir(), "/GSE74821/Samplesheet.csv"),
  id_colname = "IDFILE"
)

# (re)Normalise data by removing outliers
nacho_norm <- normalise(
  nacho_object = nacho,
  remove_outliers = TRUE
)

# (re)Normalise data with "GLM" method and removing outliers
nacho_norm <- normalise(
  nacho_object = nacho,
  normalisation_method = "GLM",
  remove_outliers = TRUE
)

---------------------------------------------------------------------

**print.nacho**

*Print method for "nacho" object*

**Description**

This function allows to print text and figures from the results of a call to `load_rcc()` or `normalise()`. It is intended to be used in a Rmarkdown chunk.

**Usage**

```r
## S3 method for class 'nacho'
print(
  x,
  colour = "CartridgeID",
  size = 0.5,
  show_legend = FALSE,
  show_outliers = TRUE,
  outliers_factor = 1,
  outliers_labels = NULL,
  echo = FALSE,
  title_level = 1,
  ...
)
```

**Arguments**

- `x` [list] A list object of class "nacho" obtained from `load_rcc()` or `normalise()`.
colour [character] Character string of the column in ssheet_csv or more generally in nacho_object$nacho to be used as grouping colour.

size [numeric] A numeric controlling point size (ggplot2::geom_point() or ggbeeswarm::geom_beeswarm()) or line size (ggplot2::geom_line()).

show_legend [logical] Boolean to indicate whether the plot legends should be plotted (TRUE) or not (FALSE). Default is TRUE.

show_outliers [logical] Boolean to indicate whether the outliers should be highlighted in red (TRUE) or not (FALSE). Default is TRUE.

outliers_factor [numeric] Size factor for outliers compared to size. Default is 1.

outliers_labels [character] Character to indicate which column in nacho_object$nacho should be used to be printed as the labels for outliers or not. Default is NULL.

echo [logical] A boolean to indicate whether text and plots should be printed. Mainly for use within a Rmarkdown chunk.

title_level [numeric] A numeric to indicate the title level to start with, using markdown style, i.e., the number of "#".

... Other arguments (Not used).

Examples

data(GSE74821)
print(GSE74821)

render(nacho_object, colour = "CartridgeID", output_file = "NACHO_QC.html", output_dir = ".", size = 1, show_legend = TRUE, show_outliers = TRUE)
render

  outliers_factor = 1,
  outliers_labels = NULL,
  clean = TRUE
)

Arguments

nacho_object [list] A list object of class "nacho" obtained from load_rcc() or normalise().
colour [character] Character string of the column in ssheet_csv or more generally in nacho_object$nacho to be used as grouping colour.
output_file [character] The name of the output file.
output_dir [character] The output directory for the rendered output_file. This allows for a choice of an alternate directory to which the output file should be written (the default output directory is the working directory, i.e., .). If a path is provided with a filename in output_file the directory specified here will take precedence. Please note that any directory path provided will create any necessary directories if they do not exist.
size [numeric] A numeric controlling point size (ggplot2::geom_point() or ggbeeswarm::geom_beeswarm()) or line size (ggplot2::geom_line()).
show_legend [logical] Boolean to indicate whether the plot legends should be plotted (TRUE) or not (FALSE). Default is TRUE.
show_outliers [logical] Boolean to indicate whether the outliers should be highlighted in red (TRUE) or not (FALSE). Default is TRUE.
outliers_factor [numeric] Size factor for outliers compared to size. Default is 1.
outliers_labels [character] Character to indicate which column in nacho_object$nacho should be used to be printed as the labels for outliers or not. Default is NULL.
clean [logical] Boolean to indicate whether the Rmd and Rdata file used to produce the HTML report are removed from output_dir. Default is TRUE.

Examples

if (interactive()) {
  data(GSE74821)
  render(GSE74821)
}
visualise

Visualise quality-control metrics of a "nacho" object

Description

This function allows to visualise results from \texttt{load\_rcc()} or \texttt{normalise()} several quality-control metrics in an interactive shiny application, in which thresholds can be customised and exported.

Usage

\begin{verbatim}
visualise(nacho_object)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{nacho\_object} \hspace{1em} [list] A list object of class "nacho" obtained from \texttt{load\_rcc()} or \texttt{normalise()}.
\end{itemize}

Examples

\begin{verbatim}
if (interactive()) {
  data(GSE74821)
  # Must be run in an interactive R session!
  visualise(GSE74821)
}

each (interactive()) {
  library(GEOquery)
  library(NACHO)

  # Import data from GEO
  gse <- GEOquery::getGEO(GEO = "GSE74821")
  targets <- Biobase::pData(Biobase::phenoData(gse[[1]]))
  GEOquery::getGEOSuppFiles(GEO = "GSE74821", baseDir = tempdir())
  utils::untar(
    tarfile = paste0(tempdir(), "/GSE74821/GSE74821_RAW.tar"),
    exdir = paste0(tempdir(), "/GSE74821")
  )
  targets$IDFILE <- list.files(
    path = paste0(tempdir(), "/GSE74821"),
    pattern = ".RCC.gz$"
  )
  targets[] <- lapply(X = targets, FUN = iconv, from = "latin1", to = "ASCII")
  utils::write.csv(
    x = targets,
    file = paste0(tempdir(), "/GSE74821/Samplesheet.csv"
  )
  )

  # Read RCC files and format
  nacho <- load\_rcc(
    data\_directory = paste0(tempdir(), "/GSE74821"),
  

data(GSE74821)
\end{verbatim}

if (interactive()) {
  visualise(GSE74821)
}
visualise

```r
ssheet.csv = paste0(tempdir(), "/GSE74821/Samplesheet.csv"),
id_colname = "IDFILE"
)
visualise(nacho)

# (re)Normalise data by removing outliers
nacho_norm <- normalise(
  nacho_object = nacho,
  remove_outliers = TRUE
)
visualise(nacho_norm)

# (re)Normalise data with "GLM" method and removing outliers
nacho_norm <- normalise(
  nacho_object = nacho,
  normalisation_method = "GLM",
  remove_outliers = TRUE
)
visualise(nacho_norm)
```
Index

*Topic **datasets**
  GSE74821, 5

autoplot.nacho, 2

character, 3, 4, 6, 7, 9, 10, 12, 13
check_outliers, 4

data.frame, 6, 7, 10
deploy, 4

ggbeeswarm::geom_beeswarm(), 3, 12, 13
ggplot2::geom_line(), 3, 12, 13
ggplot2::geom_point(), 3, 12, 13
GSE74821, 5

intersect(), 8

list, 3–7, 9–11, 13, 14
load_rcc, 5
load_rcc(), 3, 4, 6–11, 13, 14
logical, 3, 5, 6, 9, 10, 12, 13

nacho_conflicts, 8
normalise, 8
normalise(), 3, 4, 9–11, 13, 14
normalize(normalise), 8
numeric, 3, 6, 7, 9, 10, 12, 13

print.nacho, 11

render, 12
render(), 2

setdiff(), 8
setequal(), 8
summarize(load_rcc), 5

union(), 8

visualise, 14
visualise(), 2, 12
visualize(visualise), 14