Package 'NACHO'

October 12, 2019

Type Package

Title NanoString Quality Control Dashboard

Version 0.6.1

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/microRNA (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 utils, tibble, dplyr, tidyr, shiny, scales, ggplot2, ggbeeswarm, ggrepel, ggpubr, gtools, magrittr, knitr, rmarkdown, sessioninfo

Suggests covr, Biobase, GEOquery, testthat

SystemRequirements pandoc (>= 1.12.3), pandoc-citeproc

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Mickaël Canouil [aut, cre] (<https://orcid.org/0000-0002-3396-4549>),
Roderick Slieker [aut] (<https://orcid.org/0000-0003-0961-9152>),
Gerard Bouland [aut]

Maintainer Mickaël Canouil <mickael.canouil@cnrs.fr>

Repository CRAN

Date/Publication 2019-10-12 07:30:10 UTC

R topics documented:

autoplot.nacho 2
GSE74821 3
normalise 4
print.nacho 6
render 7
summarise 8
visualise 10

Index **13**

`autoplot.nacho` *Plot quality-control metrics and thresholds*

Description

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

Usage

```
## S3 method for class 'nacho'
autoplot(object, x, colour = "CartridgeID", size = 0.5,
  show_legend = TRUE, attribute = NULL, ...)
```

Arguments

`object`	[list] List obtained from summarise 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) • `"PC"` (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 ssheet_csv or more generally in nacho_object$nacho to be used as grouping colour.
size	[numeric] A numeric controlling point size (geom_point or geom_beeswarm) or line size (geom_line).
show_legend	[logical] Boolean to indicate whether the plot legends should be plotted (TRUE) or not (FALSE). Default is TRUE.
attribute	[character] A character string to indicate which RCC attributes should be used
...	Other arguments (Not used).

Examples

```
data(GSE74821)
autoplot(GSE74821, x = "BD")
autoplot(GSE74821, x = "PCA12")
autoplot(GSE74821, x = "NORM")
```

GSE74821 *Presummarised data from GSE74821 (20 samples)*

Description

NanoString nCounter RUO-PAM50 Gene Expression Custom CodeSet

Usage

```
GSE74821
```

Format

A [list] object

Source

GSE74821

normalise *(re)Normalise a dataset read from summarise*

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 summarise.

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] List obtained from summarise 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`):

- Binding Density (`BD`) < 0.1
- Binding Density (`BD`) > 2.25
- Imaging (`FoV`) < 75
- Positive Control Linearity (`PC`) < 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 summarise in `id_colname`.

`housekeeping_genes` [character] Value passed to summarise or normalise.

`housekeeping_predict` [logical] Value passed to summarise.

`housekeeping_norm` [logical] Value passed to summarise or normalise.

`normalisation_method` [character] Value passed to summarise or normalise.

`remove_outliers` [logical] Value passed to normalise.

`n_comp` [numeric] Value passed to summarise.

`data_directory` [character] Value passed to summarise.

`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 <- summarise(
    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 `summarise()` or `normalise()`. It is intended to be used in a `Rmarkdown` chunk.

Usage

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

Arguments

`x`	[list] List obtained from summarise 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 (geom_point or geom_beeswarm) or line size (geom_line).
`show_legend`	[logical] Boolean to indicate whether the plot legends should be plotted (`TRUE`) or not (`FALSE`). Default is `TRUE`.
`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` *Render a HTML report from summarise or normalise*

Description

This function create a `Rmarkdown` script and render it as a HTML document. The HTML document is a quality-control report using all the metrics from visualise based on recommendations from NanoString.

Usage

```
render(nacho_object, colour = "CartridgeID",
  output_file = "NACHO_QC.html", output_dir = ".", size = 0.5,
  show_legend = TRUE, clean = TRUE)
```

Arguments

nacho_object	[list] List obtained from summarise 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 (geom_point or geom_beeswarm) or line size (geom_line).
show_legend	[logical] Boolean to indicate whether the plot legends should be plotted (TRUE) or not (FALSE). Default is TRUE.
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)
}
```

---

summarise *Summarise data from RCC NanoString files (and normalise them)*

Description

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

Usage

```
summarise(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 containing parameters and data:

`access` [character] Value passed to summarise in `id_colname`.

`housekeeping_genes` [character] Value passed to summarise.

`housekeeping_predict` [logical] Value passed to summarise.

`housekeeping_norm` [logical] Value passed to summarise.

`normalisation_method` [character] Value passed to summarise.

`remove_outliers` [logical] `FALSE`.

`n_comp` [numeric] Value passed to summarise.

`data_directory` [character] Value passed to summarise.

`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 <- summarise(
    data_directory = paste0(tempdir(), "/GSE74821"),
    ssheet_csv = paste0(tempdir(), "/GSE74821/Samplesheet.csv"),
    id_colname = "IDFILE"
  )
}
```

`visualise` *Visualise quality-control metrics using a shiny app*

Description

This function allows to visualise several quality-control metrics in an interactive shiny application, in which thresholds can be customised and exported to the global environment.

Usage

```
visualise(nacho_object)
```

Arguments

`nacho_object`	[list] List obtained from summarise or normalise.

Examples

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

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 <- summarise(
    data_directory = paste0(tempdir(), "/GSE74821"),
    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, 3

autoplot.nacho, 2

character, *2–5*, *7–9*

data.frame, *5*, *9*

geom_beeswarm, *3*, *7*, *8*
geom_line, *3*, *7*, *8*
geom_point, *3*, *7*, *8*
GSE74821, 3

list, *2–5*, *7–10*
logical, *3–5*, *7–9*

normalise, *2*, *4*, 4, *5*, *7*, *8*, *10*
normalize *(*normalise*)*, 4
numeric, *3–5*, *7–9*

print.nacho, 6

render, *2*, 7

shiny, *10*
summarise, *2*, *4*, *5*, *7*, *8*, 8, *9*, *10*
summarize *(*summarise*)*, 8

visualise, *2*, *7*, 10
visualize *(*visualise*)*, 10