Package 'NACHO'

April 29, 2019

Type Package

Title NanoString Quality Control Dashboard

Version 0.5.6

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://mcanouil.github.io/NACHO

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

Depends R (>= 3.5.0)

Imports tibble, dplyr, tidyr, purrr, shiny, scales, ggplot2, ggbeeswarm, ggrepel, ggpubr, gtools

Suggests covr, Biobase, GEOquery, utils, sessioninfo, knitr, rmarkdown, testthat

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 2019-04-29 19:30:13 UTC

R topics documented:

`GSE74821` *Presummarised data from GSE74821 (20 samples).*

Description

NanoString nCounter RUO-PAM50 Gene Expression Custom CodeSet

Usage

```
GSE74821
```

Format

A 'list' object

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74821

`normalise` *normalise*

Description

normalise

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`	[vector(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
)
}
```

`summarise` *summarise*

Description

summarise

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/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` [vector(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.

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*

Description

visualise

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