Package 'DRomics'

January 16, 2019

Title Dose Response for Omics

Version 1.0-2

Description Several functions are provided for dose-response (or concentration-response) characterization from omics data. 'DRomics' is especially dedicated to omics data obtained using a typical dose-response design, favoring a great number of tested doses (or concentrations, at least 6, and the more the better) rather than a great number of replicates (no need of three replicates). 'DRomics' provides functions 1) to check and normalize data, 2) to select monotonic or biphasic significantly responding items (e.g. probes, metabolites), 3) to choose the best-fit model among a predefined family of monotonic and biphasic models to describe each selected item 4) to derive a benchmark dose or concentration and a typology of response from each fitted curve. In the available version data are supposed to be single-channel microarray data transformed in log2, or another type of data that can be directly fitted by least-square regression without any normalization step. In the future this tool will also be able to process RNA-seq data. For further details see Larras et al (2018) <DOI:10.1021/acs.est.8b04752>.

Depends R (>= 3.4.0), limma, utils, grDevices

Imports stats, graphics, ggplot2

Suggests parallel, shiny, shinyBS, shinyjs, testthat

License GPL (>= 2)

Encoding UTF-8

Contact Marie-Laure Delignette-Muller

<marielaure.delignettemuller@vetagro-sup.fr>

NeedsCompilation no

Author Marie-Laure Delignette-Muller [aut],
Elise Billoir [aut],
Floriane Larras [ctb],
Aurelie Siberchicot [aut, cre]

Maintainer Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>

Repository CRAN

Date/Publication 2019-01-16 18:50:11 UTC

R topics documented:

`DRomics-package` *Overview of the* **DRomics** *package*

Description

DRomics provides several functions for dose-response (or concentration-response) characterization from omics data. It is especially dedicated to omics data obtained using a typical dose-response design, favoring a great number of tested doses (or concentrations, at least 6, and the more the better) rather than a great number of replicates (no need of three replicates). DRomics provides four main functions described as follows:

- `omicdata` to check and normalize data,
- `itemselect` to select monotonic or biphasic significant responses,
- `drcfit` to choose the best-fit model among a predefined family of monotonic and biphasic models to describe each significant response and classify it in a typology of response,
- and `bmdcalc` to derive a benchmark dose or concentration from each fitted curve.

In the available version data are supposed to be single-channel microarray data transformed in log2, or another type of data that can be directly fitted by least-square regression without any normalization step. In the future this tool will also be able to process RNA-seq data.

Below is proposed an example including each step or the workflow.

Author(s)

Marie-Laure Delignette-Muller, Elise Billoir, Floriane Larras and Aurelie Siberchicot.

See Also

See `omicdata`, `itemselect`, `drcfit`, `bmdcalc` for details about each function.

Examples

```
# Step 1: importation, check and normalization of data if need
#
## here cyclicloess normalization of a small transcriptomics data set
```

```
## (sample of a real data set)

datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")
(o <- omicdata(datatxt, check = TRUE, norm.method = "cyclicloess"))
plot(o)

# Step 2: item selection using the quadratic method
#
## the quadratic method is the one we preconize to select both
## monotonic and biphasic curves from
## a typical dose-response design (with few replicates per dose)

(s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.001))


# Step 3: fit of dose-response models, choice of the best fit for each curve
# and definition of the typology of response
#

(f <- drcfit(s_quad, progressbar = TRUE))
f$fitres
plot(f)

# Step 4: calculation of x-fold and z-SD benchmark doses
#

(r <- bmdcalc(f, z = 1, x = 10))
plot(r, BMDtype = "zSD", plottype = "ecdf", bytypology = FALSE)
plot(r, BMDtype = "xfold", plottype = "hist", bytypology = TRUE, hist.bins = 10)



# About using the DRomics-shiny app
#

if(interactive()) {
  appDir <- system.file("DRomics-shiny", package = "DRomics")
  shiny::runApp(appDir, display.mode = "normal")
}
```

---

bmdcalc *Computation of benchmark doses for responsive items*

Description

Computes x-fold and z-SD benchmark doses for each responsive item using the best fit dose-reponse model.

Usage

```
bmdcalc(f, z = 1, x = 10)

## S3 method for class 'bmdcalc'
print(x, ...)

## S3 method for class 'bmdcalc'
plot(x, BMDtype = c("zSD", "xfold"),
            plottype = c("ecdf", "hist", "density"), bytypology = FALSE,
            hist.bins = 30, ...)
```

Arguments

f	An object of class "drcfit" returned by the function drcfit.
z	Value of z defining the BMD-zSD as the dose at which the response is reaching y0 +/- z * SD, with y0 the level at the control given by the dose-response fitted model and SD the residual standard deviation of the dose-response fitted model.
x	Value of x given as a percentage and defining the BMD-xfold as the dose at which the response is reaching y0 +/- (x/100) * y0, with y0 the level at the control given by the dose-response fitted model. For print and plot functions, an object of class "bmdcalc".
BMDtype	The type of BMD to plot, "zSD" (default choice) or "xfold".
plottype	The type plot, "ecdf" for an empirical cumulative distribution plot (default choice), "hist" for a histogram or "density" for a density plot.
bytypology	If TRUE the plot is split by typology.
hist.bins	The number of bins, only used for histogram(s).
...	further arguments passed to graphical or print functions.

Details

Two types of benchmark doses (BMD) were computed for each responsive item using the best fit dose-reponse model previously obtained using the drcfit function :

- the BMD-zSD defined as the dose at which the response is reaching y0 +/- z * SD, with y0 the level at the control given by the dose-response model, SD the residual standard deviation of the dose response model fit and z given as an input (z fixed to 1 by default),
- the BMD-xfold defined as the dose at which the response is reaching y0 +/- (x/100) * y0, with y0 the level at the control given by the dose-response fitted model and x the percentage given as an input (x fixed at 10 by default.)

When there is no analytical solution for the BMD, it is numerically searched along the fitted curve using the uniroot function.

In cases where the BMD cannot be reached due to the asymptote at high doses, NaN is returned. In cases where the BMD is not reached at the highest tested dose, NA is returned.

Value

bmdcalc returns an object of class "bmdcalc", a list with 4 components:

res	a data frame reporting the results of the fit and BMD computation on each selected item sorted in the ascending order of the adjusted p-values returned by function itemselect. The different columns correspond to the identifier of each item (id), the row number of this item in the initial data set (irow), the adjusted p-value of the selection step (adjpvalue), the name of the best fit model (model), the number of fitted parameters (nbpar), the values of the parameters b, c, d, e and f, (NA for non used parameters), the residual standard deviation (SDres), the typology of the curve (typology, (twelve class typology described in the help of the drcfit function)), the rough trend of the curve (trend) defined with four classes (U, bell, increasing or decreasing shape), the theoretical value at the control (y0), the theoretical y range for x within the range of tested doses (yrange) and for biphasic curves the x value at which their extremum is reached (xextrem) and the corresponding y value (yextrem), the BMD-zSD value (BMD.zSD) and the BMD-xfold value (BMD.xfold).
z	Value of z given in input to define the BMD-zSD.
x	Value of x given in input as a percentage to define the BMD-xfold.
omicdata	The corresponding object of class "omicdata" given in input (component of itemselect).

Author(s)

Marie-Laure Delignette-Muller and Elise Billoir

References

Larras F, Billoir E, Baillard V, Siberchicot A, Scholz S, Wubet T, Tarkka M, Schmitt-Jansen M and Delignette-Muller ML (2018). DRomics: a turnkey tool to support the use of the dose-response framework for omics data in ecological risk assessment. Environmental science & technology.https://doi.org/10.1021/acs.est.8b04752

See Also

See uniroot for details about the function used for the numerical search of the benchmark dose for cases where there is no analytical solution.

Examples

```
# (1) a toy example (a very small subsample of a transcriptomics data set)
#
datatxt <- system.file("extdata", "transcripto_very_small_sample.txt", package="DRomics")

# to test the package on a small (for a quick calculation) but not very small data set
# use the following commented line
# datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")
```

```
(o <- omicdata(datatxt, check = TRUE, norm.method = "cyclicloess"))
(s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.01))
(f <- drcfit(s_quad, progressbar = TRUE))
(r <- bmdcalc(f))
plot(r)

# changing the values of z and x for BMD calculation
(rb <- bmdcalc(f, z = 2, x = 50))
plot(rb)

# (2) an example on a transcriptomics data set (a subsample of a greater data set)
#

datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")

# to test the package on a small (for a quick calculation) but not very small data set
# use the following commented line
# datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")

(o <- omicdata(datatxt, check = TRUE, norm.method = "cyclicloess"))
(s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.01))
(f <- drcfit(s_quad, progressbar = TRUE))
(r <- bmdcalc(f))
plot(r)

# different plots of BMD-zSD

plot(r, plottype = "hist", bytypology = FALSE)
plot(r, plottype = "density", bytypology = FALSE)
plot(r, plottype = "hist", bytypology = TRUE)

# a plot of BMD-xfold (by default BMD-zSD is plotted)
plot(r, BMDtype = "xfold", plottype = "hist", bytypology = TRUE, hist.bins = 10)
```

---

drcfit *Dose response modelling for responsive items*

Description

Fits dose reponse models to responsive items.

Usage

```
drcfit(itemselect, sigmoid.model = c("Hill", "log-probit"),
                   progressbar = TRUE, saveplot2pdf = TRUE,
                   parallel = c("no", "snow", "multicore"), ncpus)

## S3 method for class 'drcfit'
```

```
print(x, ...)

## S3 method for class 'drcfit'
plot(x, items, ...)
```

Arguments

itemselect	An object of class "itemselect" returned by the function itemselect.
sigmoid.model	The chosen sigmoid model, "Hill" (default choice) or "log-probit".
progressbar	If TRUE a progress bar is used to follow the fitting process.
saveplot2pdf	If TRUE a pdf file named drcfitplot.pdf is saved containing all the fitted dose-response curves sorted by adjusted p-values of the selection step.
parallel	The type of parallel operation to be used, "snow" or "multicore" (the second one not being available on Windows), or "no" if no parallel operation.
ncpus	Number of processes to be used in parallel operation : typically one would fix it to the number of available CPUs.
x	An object of class "drcfit".
items	Argument of the plot.drcfit function : the number of the first fits to plot (20 items max) or the character vector specifying the identifiers of the items to plot (20 items max).
...	further arguments passed to graphical or print functions.

Details

For each selected item, five dose-response models (linear, Hill, exponential, Gauss-probit and log-Gauss-probit, see Larras et al. 2018 for their definition) were fitted by non linear regression, using the nls function. The best one was chosen as the one giving the lowest AIC value. Items with the best AIC value not lower than the AIC value of the null model (constant model) minus 2 were eliminated. Items with the best fit showing a global significant quadratic trend of the residuals as a function of the dose (in rank-scale) were also eliminated (the best fit is considered as not reliable in such cases). Each retained item is classified in a twelve class typology depending of the chosen model and of its parameter values :

- H.inc for increasing Hill curves (or lP.inc if sigmoid.model = "log-probit"),
- H.dec for decreasing Hill curves (or lP.dec if sigmoid.model = "log-probit"),
- L.inc for increasing linear curves,
- L.dec for decreasing linear curves,
- E.inc.convex for increasing convex exponential curves,
- E.dec.concave for decreasing concave exponential curves,
- E.inc.concave for increasing concave exponential curves,
- E.dec.convex for decreasing convex exponential curves,
- GP.U for U-shape Gauss-probit curves,
- GP.bell for bell-shape Gauss-probit curves,
- lGP.U for U-shape log-Gauss-probit curves,

- lGP.bell for bell-shape log-Gauss-probit curves.

Each retained is also classified in four classes by its global trend :

- inc for increasing curves,
- dec for decreasing curves ,
- U for U-shape curves,
- bell for bell-shape curves.

Some curves fitted by a Gauss-probit model can be classified as increasing or decreasing when the dose value at which their extremum is reached is at zero.

Value

`drcfit` returns an object of class `"drcfit"`, a list with 4 components:

`fitres`	a data frame reporting the results of the fit on each selected item (one line per item) sorted in the ascending order of the adjusted p-values returned by function `itemselect`. The different columns correspond to the identifier of each item (`id`), the row number of this item in the initial data set (`irow`), the adjusted p-value of the selection step (`adjpvalue`), the name of the best fit model (`model`), the number of fitted parameters (`nbpar`), the values of the parameters `b`, `c`, `d`, `e` and `f`, (`NA` for non used parameters), the residual standard deviation (`SDres`), the typology of the curve (`typology`), the rough trend of the curve (`trend`) defined with four classes (U, bell, increasing or decreasing shape), the theoretical value at the control `y0`), the theoretical y range for x within the range of tested doses (`yrange`), for biphasic curves the x value at which their extremum is reached (`xextrem`) and the corresponding y value (`yextrem`).
`omicdata`	The corresponding object of class `"omicdata"` given in input (component of itemselect).
`n.failure`	The number of previously selected items on which the workflow failed to fit an acceptable model.
`AIC.val`	a data frame reporting AIC values for each selected item (one line per item) and each fitted model (one colum per model with the AIC value fixed at `Inf` when the fit failed).

Author(s)

Marie-Laure Delignette-Muller

References

Larras F, Billoir E, Baillard V, Siberchicot A, Scholz S, Wubet T, Tarkka M, Schmitt-Jansen M and Delignette-Muller ML (2018). DRomics: a turnkey tool to support the use of the dose-response framework for omics data in ecological risk assessment. Environmental science & technology.https://doi.org/10.1021/acs.est.8b04752

See Also

See nls for details about the non linear regression function.

Examples

```
# (1) a toy example (a very small subsample of a transcriptomics data set)
#
datatxt <- system.file("extdata", "transcripto_very_small_sample.txt", package="DRomics")

# to test the package on a small (for a quick calculation) but not very small data set
# use the following commented line
# datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")

(o <- omicdata(datatxt, check = TRUE, norm.method = "cyclicloess"))
(s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.05))
(f <- drcfit(s_quad, progressbar = TRUE))

# Default plot
plot(f)


# (2) an example on a transcriptomics data set (a subsample of a greater data set)
#

datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")

(o <- omicdata(datatxt, check = TRUE, norm.method = "cyclicloess"))
(s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.05))
(f <- drcfit(s_quad, progressbar = TRUE))

# Default plot
plot(f)

# Plot of the first 12 most responsive items
plot(f, items = 12)

# Plot of the chosen items in the chosen order
plot(f, items = c("301.2", "363.1", "383.1"))


# (3) Comparison of parallel and non paralell implementations on a
#     larger selection of items
#

   s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.05)
    system.time(f1 <- drcfit(s_quad, progressbar = TRUE))
   system.time(f2 <- drcfit(s_quad, progressbar = FALSE, parallel = "snow", ncpus = 2))
```

`itemselect` *Selection of significantly responsive items*

Description

Significantly responsive items are selected using one of the three proposed methods: a quadratic trend test, a linear trend test or an ANOVA-based test.

Usage

```
itemselect(omicdata, select.method = c("quadratic", "linear", "ANOVA"), FDR = 0.05)

## S3 method for class 'itemselect'
print(x, ...)
```

Arguments

`omicdata`	An object of class `"omicdata"` returned by the function `omicdata`.
`select.method`	The chosen method for selecting items using the `limma` package : `"quadratic"` for a quadratic trend test on dose ranks, `"linear"` for a linear trend test on dose ranks and `"ANOVA"` for an ANOVA-type test (see details for further explaination).
`FDR`	The threshold in term of FDR (False Discovery Rate) for selecting responsive items.
`x`	An object of class `"itemselect"`.
`...`	further arguments passed to print function.

Details

The selection of responsive items is performed using the `limma` package. Three methods are proposed, all implemented using functions `lmFit`, `eBayes` and `topTable` with p-values ajusted for multiple testing using the Benjamini-Hochberg method, with the false discovery rate given in input (argument `FDR`).

- The ANOVA_based test (`"ANOVA"`) is classically used for selection of omics data in the general case but it requires many replicates per dose to be efficient, and is thus not really suited for a dose-response design.
- The linear trend test (`"linear"`) aims at detecting monotonic trends from dose-response designs, whatever the number of replicates per dose. As proposed by Tukey (1985), it tests the global significance of a linear model describing the response as a function of the dose in rank-scale.
- The quadratic trend test (`"quadratic"`) tests the global significance of a quadratic model describing the response as a function of the dose in rank-scale. It is a variant of the linear trend method that aims at detecting monotonic and non monotonic trends from a dose-response designs, whatever the number of replicates per dose (default chosen method).

Value

`itemselect` returns an object of class `"itemselect"`, a list with 5 components:

`adjpvalue`	the vector of the p-values adjusted by the Benjamini-Hochberg method for selected items (adjpvalue inferior to FDR) sorted in ascending order
`selectindex`	the corresponding vector of row indices of selected items in the object omicdata
`omicdata`	The corresponding object of class "omicdata" given in input.
`select.method`	The selection method given in input.
`FDR`	The threshold in term of FDR given in input.

The print of a `"itemselect"` object gives the number of selected items and the identifiers of the 20 most responsive items.

Author(s)

Marie-Laure Delignette-Muller

References

Tukey JW, Ciminera JL and Heyse JF (1985), *Testing the statistical certainty of a response to increasing doses of a drug*. Biometrics, 295-301.

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, and Smyth, GK (2015), *limma powers differential expression analyses for RNA-sequencing and microarray studies*. Nucleic Acids Research 43, e47.

See Also

See `lmFit`, `eBayes` and `topTable` for details about the used functions of the `limma` package.

Examples

```
# (1) an example on a transcriptomics data (a subsample of a greater data set)
#
datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")

(o <- omicdata(datatxt, check = TRUE, norm.method = "cycliclоess"))

# 1.a using the quadratic trend test
#
(s_quad <- itemselect(o, select.method = "quadratic", FDR = 0.05))

# 1.b using the linear trend test
#
(s_lin <- itemselect(o, select.method = "linear", FDR = 0.05))

# 1.c using the ANOVA-based test
#
(s_ANOVA <- itemselect(o, select.method = "ANOVA", FDR = 0.05))
```

```
# 1.d using the quadratic trend test with a smaller false discovery rate
#
(s_quad.2 <- itemselect(o, select.method = "quadratic", FDR = 0.001))
```

`omicdata` *Import, check and normalization of omics data*

Description

Data are imported from a .txt file, checked (see the description of argument `file` for the required format of data) and normalized if required.

Usage

```
omicdata(file, check = TRUE, norm.method = c("none", "cyclicloess", "quantile", "scale"))

## S3 method for class 'omicdata'
print(x, ...)
## S3 method for class 'omicdata'
plot(x, ...)
```

Arguments

`file`	The name of the .txt file containing one row per item, with the first column corresponding to the identifier of each item, in a column named `"item"`, and the other columns giving the responses of the item for each replicate at each dose or concentration. The names of the corresponding columns must correspond to the tested doses or concentrations in a numeric format for the corresponding replicate (for example, if there are triplicates for each treatment, column names can be "item", 0, 0, 0, 0.1, 0.1, 0.1, etc.
`check`	If TRUE the format of the input file is checked.
`norm.method`	If `"none"` no normalization is performed, else a normalization is performed using the function normalizeBetweenArrays of the `limma` package using the specified method.
`x`	An object of class `"omicdata"`.
`...`	further arguments passed to print or plot functions.

Details

This function imports the data from a .txt file, then checks the format of data (see the description of argument `file` for the required format of data) and gives in the `print` information that should help the user to check that the coding of data is correct : the tested doses (or concentrations) the number of replicates for each dose, the number of items, the identifiers of the first 20 items. If the argument `norm.method` is not `"none"`, data are normalized using the function `normalizeBetweenArrays` of the `limma` package using the specified method : `"cyclicloess"` (default choice), `"quantile"` or `"scale"`.

Value

`omicdata` returns an object of class "omicdata", a list with 5 components:

`data`	the numeric matrix of responses of each item in each replicate (one line per item, one column per replicate)
`dose`	the numeric vector of the tested doses or concentrations corresponding to each column of data
`item`	the character vector of the identifiers of the items, corresponding to each line of data
`design`	a table with the experimental design (tested doses and number of replicates for each dose) for control by the user
`data.mean`	the numeric matrix of mean responses of each item per dose (mean of the corresponding replicates) (one line per item, one column per unique value of the dose
`norm.method`	The normalization method specified in input
`data.beforenorm`	the numeric matrix of responses of each item in each replicate (one line per item, one column per replicate) before normalization

The print of a `omicdata` object gives the tested doses (or concentrations) and number of replicates for each dose, the number of items, the identifiers of the first 20 items (for check of good coding of data) and the normalization method. The plot of a `omicdata` object shows the data distribution for each dose or concentration and replicate before and after normalization.

Author(s)

Marie-Laure Delignette-Muller

References

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, and Smyth, GK (2015), *limma powers differential expression analyses for RNA-sequencing and microarray studies*. Nucleic Acids Research 43, e47.

See Also

See `normalizeBetweenArrays` for details about the normalization.

Examples

```
# (1) import, check and normalization of transcriptomics data
# (an example on a subsample of a greater data set)
#
datatxt <- system.file("extdata", "transcripto_sample.txt", package="DRomics")

o <- omicdata(datatxt, check = TRUE, norm.method = "cycliccloess")
print(o)
plot(o)
```

```
# (2) normalization with other methods
(o.2 <- omicdata(datatxt, check = TRUE, norm.method = "quantile"))
(o.3 <- omicdata(datatxt, check = TRUE, norm.method = "scale"))
```

Index