Package ‘GRmetrics’

April 14, 2017

Type Package
Title Calculate growth-rate inhibition (GR) metrics
Version 1.0.0
Date 2016-10-01
Author Nicholas Clark
Maintainer Nicholas Clark <nicholas.clark00@gmail.com>
biocViews CellBasedAssays, CellBiology, Software, TimeCourse, Visualization
Description Functions for calculating and visualizing growth-rate inhibition (GR) metrics.
License GPL-3
URL https://github.com/uc-bd2k/GRmetrics
BugReports https://github.com/uc-bd2k/GRmetrics/issues
LazyData TRUE
Depends R (>= 3.3), SummarizedExperiment
Imports drc, plotly, ggplot2, S4Vectors
RoxygenNote 5.0.1
Suggests knitr, rmarkdown, BiocStyle
VignetteBuilder knitr
NeedsCompilation no

R topics documented:

GRbox .................................................. 2
GRdrawDRC .......................................... 3
GRfit .................................................. 4
GRgetDefs ........................................... 7
GRgetGroupVars ..................................... 8
GRgetMetrics ....................................... 9
GRgetValues ........................................ 10
GRscatter ........................................... 11
inputCaseA .......................................... 12
inputCaseC .......................................... 13

Index 15
**Description**

Given a SummarizedExperiment object created by `GRfit`, this function creates boxplots according to the parameters below.

**Usage**

```r
GRbox(fitData, metric, groupVariable, pointColor, factors = "all", wilA = NULL, wilB = NULL, plotly = TRUE)
```

**Arguments**

- `fitData`: a SummarizedExperiment object, generated by the GRfit function.
- `metric`: the GR metric (GR50, GRmax, etc.) or traditional metric (IC50, Emax, etc.) that will be used for the boxplot.
- `groupVariable`: the name of the variable from data (e.g. drug, cell-line, etc.) to select factors from.
- `pointColor`: a variable that defines the coloring of the points overlayed on the boxplot.
- `factors`: a vector of values of `groupVariable` of data that define which variables to make boxplots for. By default, a separate boxplot is made for each unique value of groupVariable.
- `wilA`: one value or a vector of values from `factors`, i.e. a subset of the boxplots. If specified, a one-sided Wilcoxon rank sum test (wilcox.test) will be performed between "wilA" and "wilB" and the results will be displayed on the figure. The null hypothesis that the values from "wilA" and "wilB" have the same mean will be tested against the alternative hypothesis that the mean of the "wilB" values is greater than that of the "wilA" values.
- `wilB`: one value or a vector of values from `factors`, i.e. a subset of the boxplots (not overlapping "wilA").
- `plotly`: a logical value indicating whether to output a ggplot2 graph or an interactive ggplotly graph.

**Details**

Given a SummarizedExperiment object created by `GRfit`, this function creates boxplots of a given GR metric (GR50, GRmax, etc.) or traditional metric (IC50, Emax, etc.) for values of the grouping variable. The results can be viewed in a static ggplot image or an interactive plotly graph.

By default, a boxplot is created for all unique values of the grouping variable. The "factors" parameter can be used to specify a smaller subset of values for which to create boxplots. Points are overlayed on the boxplots and they can be colored by the variable specified in the `pointColor` parameter. If `pointColor` is set to NULL, the points will all be black. The results can be viewed in a static ggplot image or an interactive plotly graph.
Value

ggplot2 or ggplotly boxplots of the factors along the x-axis, with points colored by the given variable.

Author(s)
Nicholas Clark

See Also
To create the object needed for this function, see GRfit. For other visualizations, see GRdrawDRC and GRscatter. For online GR calculator and browser, see http://www.grcalculator.org.

Examples

```r
# Load Case A (example 1) input
data("inputCaseA")
head(inputCaseA)

# Run GRfit function with case = "A"
output1 = GRfit(inputData = inputCaseA,
groupingVariables = c('cell_line', 'agent', 'perturbation', 'replicate',
'time'))
GRbox(output1, metric = 'GRinf',
groupVariable = 'cell_line', pointColor = 'agent', factors = c('BT20',
'MCF10A'))
GRbox(output1, metric = 'GRinf',
groupVariable = 'cell_line', pointColor = 'cell_line',
factors = c('BT20', 'MCF10A'), plotly = FALSE)
GRbox(output1, metric = 'GR50', groupVariable = 'cell_line',
pointColor = 'cell_line', wilA = "BT20", wilB = c("MCF7","MCF10A"))
```

---

**GRdrawDRC**

**Dose-Response Curves**

Description

Given a SummarizedExperiment object created by GRfit, this function plots either the growth-rate inhibition (GR) dose response curves or the traditional dose response curves for a given set of data.

Usage

```
GRdrawDRC(fitData, metric = "GR", experiments = "all", min = "auto",
          max = "auto", points = TRUE, curves = TRUE, plotly = TRUE)
```

Arguments

- **fitData**: an element of class SummarizedExperiment, generated by the GRfit function.
- **metric**: either "GR" for GR dose response curves or "IC" for traditional dose response curves.
- **experiments**: the names of the experiments to plot (or "all")
- **min**: the minimum concentration to plot (for curves)
max  the maximum concentration to plot (for curves)
points a logical value indicating whether points (individual GR values) will be plotted
curves a logical value indicating whether sigmoidal dose-response curves will be plotted
plotly a logical value indicating whether to output a ggplot2 graph or a ggplotly graph

Details
Given a SummarizedExperiment object created by GRfit, this function plots these GR values (versus concentration) and/or the sigmoidal curves fitted to the sets of points. The results can be viewed in a static ggplot image or an interactive plotly graph.

The "min" and "max" parameters control the concentration values for which the curves are plotted. They are automatically set to the minimum and maximum concentrations of the data, but can be set by the user as well. "min" and "max" take raw values (not log transformed) for concentration.

By default, curves and points are plotted for all experiments. To specify a smaller set of experiments, use the "experiments" parameter. To see the names of individual experiments, use GRfitObject$parameter_table$experiment. See the examples below.

Value
ggplot2 or ggplotly graphs of Growth-rate inhibition dose-response curves

Author(s)
Nicholas Clark

See Also
To create the object needed for this function, see GRfit. For other visualizations, see GRbox and GRscatter. For online GR calculator and browser, see http://www.grcalculator.org.

Examples
# Load Case A (example 1) input
data("inputCaseA")
# Run GRfit function with case = "A"
drc_output = GRfit(inputCaseA,
groupingVariables = c("cell_line","agent"))
GRdrawDRC(drc_output, experiments = c("BT20 drugA", "MCF10A drugA", "MCF7 drugA"), min = 10^(-4), max = 10^2)
GRdrawDRC(drc_output, plotly = FALSE)
Usage

`GRfit(inputData, groupingVariables, case = "A", force = FALSE, cap = FALSE)`

Arguments

- **inputData**: a data table in one of the specified formats (Case A or Case C). See details below for description. See `data(inputCaseA)` or `data(inputCaseC)` for example input data frames. See help files for `inputCaseA` and `inputCaseC` for description of these examples.
- **groupingVariables**: a vector of column names from inputData. All of the columns in inputData except for those identified here will be averaged over.
- **case**: either "A" or "C", indicating the format of the input data. See below for descriptions of these formats.
- **force**: a logical value indicating whether to attempt to "force" a sigmoidal fit, i.e. whether to allow fits with F-test p-values greater than .05.
- **cap**: a logical value indicating whether to cap GR values (or relative cell counts) at 1. If true, all values greater than 1 will be set to 1.

Details

Calculation of GR values is performed by the function `.GRcalculate` according to the "Online Methods" section of Hafner and Niepel et al. (http://dx.doi.org/10.1038/nmeth.3853).

The fitting of the logistic curve is performed by the `.GRlogisticFit` function, which calls the `drm` function from the `drc` package to solve for the curve parameters. The GR curve fit function is given by \( f(c) = GR_{inf} + \frac{(1 - GR_{inf})}{(1 + (c/GE50)^{h_{GR}})} \) where \( c \) is concentration. The fit is performed under following constraints: \( h_{GR} \) in \([.1, 5]\), \( GR_{inf} \) in \([-1, 1]\), and \( GE50 \) in \([\min(c)\cdot1e-2, \max(c)\cdot1e2]\) (\( c \) is concentration). The initial conditions for the fitting algorithm are \( h_{GR} = 2 \), \( GR_{inf} = 0.1 \) and \( GE50 = \text{median}(c) \). The fitting of the traditional dose response curve is done using the same formula, replacing \( GR_{inf} \) with \( E_{inf} \), \( GE50 \) with \( EC50 \), and \( h_{GR} \) with \( h \). The fit is performed on the relative cell counts instead of GR values. Also, since the traditional dose response curve is bounded between 0 and 1 whereas the GR dose response curve is bounded between -1 and 1, we restrict \( E_{inf} \) to the range \([0, 1]\).

The parameters of the GR dose response curves (and traditional dose response curves) for each experiment are fitted separately. An F-test is used to compare the sigmoidal fit to a flat line fit. If the p-value of the F-test is less than .05, the sigmoidal fit is accepted. If the p-value is greater than or equal to .05, a flat horizontal line fit is given, with y equal to the mean of the GR values (or relative cell counts in the case of the traditional dose response curve). For each flat fit, \( GE50 \) (or \( EC50 \)) is set to 0, \( h_{GR} \) (or \( h \)) is set to 0.01, \( GR_{inf} \) (or \( E_{inf} \)) is set to the y value of the flat fit, and \( GR_{50} \) (or \( IC_{50} \)) is set to +/-Inf depending on whether \( GR_{inf} \) (or \( E_{inf} \)) is greater or less than .5.

The mandatory columns for `inputData` for Case "A" are the following as well as other grouping columns.

1. concentration - column with concentration values (not log transformed) of the perturbagen on which dose-response curves will be evaluated
2. cell_count - column with the measure of cell number (or a surrogate of cell number) after treatment
3. cell_count__time0 - column with initial (Time 0) cell counts - the measure of cell number in untreated wells grown in parallel until the time of treatment
4. cell_count__ctrl - column with the Control cell count: the measure of cell number in control (e.g. untreated or DMSO-treated) wells from the same plate

All other columns will be treated as additional keys on which the data will be grouped (e.g. cell_line, drug, time, replicate)

The mandatory columns for inputData for Case "C" are the following as well as other grouping columns.
1. concentration - column with concentration values (not log transformed) of the perturbagen on which dose-response curves will be evaluated
2. cell_count - column with the measure of cell number (or a surrogate of cell number)
3. time - column with the time at which a cell count is observed

All other columns will be treated as additional keys on which the data will be grouped (e.g. cell_line, drug, replicate)

Value

A SummarizedExperiment object containing GR metrics (GR50, GRmax, etc.) and traditional metrics (IC50, Emax, etc.) as well as goodness of fit measures is returned. The object also contains, in its metadata, a table of the original data converted to the style of "Case A" (with calculated GR values and relative cell counts for each row) and a vector of the grouping variables used for the calculation.

Note

To see the underlying code, use (getAnywhere(.GRlogistic_3u)), (getAnywhere(.IClogistic_3u)), (getAnywhere(.GRcalculate)), and (getAnywhere(.GRlogisticFit))

Author(s)

Nicholas Clark

References


See Also

See drm for the general logistic fit function that solves for the parameters GRinf, GEC50, and h_GR. See drmc for options of this function. Use the functions GRdrawDRC, GRbox, and GRscatter to create visualizations using the output from this function. For online GR calculator and browser, see http://www.grcalculator.org.

Examples

# Load Case A (example 1) input
data("inputCaseA")
head(inputCaseA)
# Run GRfit function with case = "A"
output1 = GRfit(inputData = inputCaseA,
groupingVariables = c('cell_line', 'agent', 'perturbation', 'replicate', 'time'))
# Overview of SummarizedExperiment output data

## Not run:

# View GR metrics table
View(GRgetMetrics(output1))

# View descriptions of each metric (or goodness of fit measure)
View(GRgetDefs(output1))

# View table of original data (converted to style of Case A) with GR values # and relative cell counts
View(GRgetValues(output1))

# View vector of grouping variables used for calculation
GRgetGroupVars(output1)

## End(Not run)

# Load Case C (example 4) input
# Same data, different format

data("inputCaseC")

head(inputCaseC)

output4 = GRfit(inputData = inputCaseC,
groupingVariables = c("cell_line","agent","perturbation","replicate","time"),
case = "C")

# Extract data tables and export to .tsv or .csv

## Not run:

# Write GR metrics parameter table to tab-separated text file
write.table(GRgetMetrics(output1), file = "filename.tsv", quote = FALSE, sep = "\t", row.names = FALSE)

# Write original data plus GR values to comma-separated file
write.table(GRgetValues(output1), file = "filename.csv", quote = FALSE, sep = ",", row.names = FALSE)

## End(Not run)

---

**GRgetDefs**

**GR metrics definitions**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a SummarizedExperiment object created by <strong>GRfit</strong>, this function returns a table with the definition of each GR metric and traditional metric that is calculated.</td>
</tr>
</tbody>
</table>

**Usage**

GRgetDefs(fitData)

**Arguments**

- `fitData` a SummarizedExperiment object, generated by the GRfit function.

**Details**

Given a SummarizedExperiment object created by **GRfit**, this function returns a table containing the definition of each GR metric, traditional metric, and goodness of fit measure in the object. This is simply a convenient accessor function, equivalent to rowData(fitData).
Value

A table of definitions of GR metrics and traditional metrics

Author(s)

Nicholas Clark

See Also

To create the object needed for this function, see GRfit.

Examples

```r
# Load Case A (example 1) input
data("inputCaseA")
head(inputCaseA)
# Run GRfit function with case = "A"
output1 = GRfit(inputData = inputCaseA,
groupingVariables = c("cell_line","agent"))
defs = GRgetDefs(output1)
# See all definitions (some will be truncated)
## Not run: View(defs)
# See the first six descriptions
head(defs)
# Look at a specific definition
defs[15,]
```

---

<table>
<thead>
<tr>
<th>GRgetGroupVars</th>
<th>Grouping Variables</th>
</tr>
</thead>
</table>

Description

Given a SummarizedExperiment object created by GRfit, this function returns a vector of the grouping variables used to create the object.

Usage

`GRgetGroupVars(fitData)`

Arguments

fitData a SummarizedExperiment object, generated by the GRfit function.

Details

Given a SummarizedExperiment object created by GRfit, this function returns a vector of the grouping variables used to create the object. These are the variables in the dataset that are not averaged over. This is simply a convenient accessor function, equivalent to `metadata(fitData)[[2]]`.

Value

A vector of grouping variables
GRgetMetrics

Author(s)
Nicholas Clark

See Also
To create the object needed for this function, see GRfit.

Examples
# Load Case A (example 1) input
data("inputCaseA")
head(inputCaseA)
# Run GRfit function with case = "A"
output1 = GRfit(inputData = inputCaseA,
groupingVariables = c(’cell_line’,’agent’))
groupVars = GRgetGroupVars(output1)
groupVars

GRgetMetrics

Description
Given a SummarizedExperiment object created by GRfit, this function returns a table of GR metrics (as well as traditional metrics) for each experiment in the dataset.

Usage
GRgetMetrics(fitData)

Arguments
fitData a SummarizedExperiment object, generated by the GRfit function.

Details
Given a SummarizedExperiment object created by GRfit, this function returns a table of GR metrics and traditional metrics along with goodness of fit measures. It also identifies each fit as flat or sigmoidal. This is simply a convenient accessor function, equivalent to cbind(as.data.frame(colData(fitData)),as.data.frame(assay(fitData))).

Value
A table with GR metrics and goodness of fit measures

Author(s)
Nicholas Clark

See Also
To create the object needed for this function, see GRfit.
Examples

```r
# Load Case A (example 1) input
data("inputCaseA")
head(inputCaseA)
# Run GRfit function with case = "A"
output1 = GRfit(inputData = inputCaseA,
groupingVariables = c("cell_line","agent"))
metrics = GRgetMetrics(output1)
## Not run: View(metrics)
head(metrics)
```

Description

Given a SummarizedExperiment object created by `GRfit`, this function returns a table of the original data (in the form of "Case A") with columns for GR values and relative cell counts.

Usage

`GRgetValues(fitData)`

Arguments

- `fitData`: a SummarizedExperiment object, generated by the GRfit function.

Details

Given a SummarizedExperiment object created by `GRfit`, this function returns a table of the original data (in the form of "Case A") with columns for GR values and relative cell counts. This is simply a convenient accessor function, equivalent to `as.data.frame(metadata(fitData)[[1]])`.

Value

A table with GR values and relative cell counts

Author(s)

Nicholas Clark

See Also

To create the object needed for this function, see `GRfit`. 

---

**GRgetValues**

<table>
<thead>
<tr>
<th>GRgetValues</th>
<th>GR values</th>
</tr>
</thead>
</table>

---
Examples

```r
# Load Case A (example 1) input
data("inputCaseA")
head(inputCaseA)
# Run GRfit function with case = "A"
output1 = GRfit(inputData = inputCaseA,
    groupingVariables = c('cell_line', 'agent'))
GRvalues = GRgetValues(output1)
head(GRvalues)
## Not run: View(GRvalues)
```

---

**GRscatter**

*Scatterplot of a given GR metric*

**Description**

Given a SummarizedExperiment object created by `GRfit`, this function creates a scatterplot according to the parameters below.

**Usage**

```r
GRscatter(fitData, metric, variable, xaxis, yaxis, plotly = TRUE)
```

**Arguments**

- `fitData`:
  a SummarizedExperiment object, generated by the `GRfit` function.

- `metric`:
  the GR metric (GR50, GRinf, h_GR, GRmax, GEC50, or GR_AOC) or traditional metric (IC50, Einf, h, Emax, EC50, or AUC) that will be used for the scatterplot.

- `variable`:
  The name of the variable from data (e.g. drug, cell-line, etc.) to select factors from.

- `xaxis`:
  a vector of values of "variable" of data to be used for the scatterplot's x-axis

- `yaxis`:
  a vector of values of "variable" of data to be used for the scatterplot's y-axis

- `plotly`:
  a logical value indicating whether to output a ggplot2 graph or a ggplotly graph

**Details**

Given a SummarizedExperiment object created by `GRfit`, this function creates a scatterplot of a given GR metric (GR50, GRmax, etc.) or traditional metric (IC50, Emax, etc.) with the "xaxis" values plotted against the "yaxis" values. The results can be viewed in a static ggplot image or an interactive plotly graph.

The xaxis and yaxis vectors must be of the same length or at least one must be of length one. For each pair of values xaxis[i] and yaxis[j], the function will create a scatterplot (all on one graph) of the specified GR metric. If a vector is of length one, it will be repeated to the length of the other vector.

**Value**

a ggplot2 or ggplotly scatterplot of the x-axis variable(s) versus the y-axis variable(s) for the given GR metric
inputCaseA

Simulated dose-response assay data (Case A)

Description

This is an example dataset corresponding to Case A. The data was generated by the python script "generate_data.py" found in the "inst/scripts/" directory. This object is meant to be used as the "inputData" argument for the GRfit function when "case" is equal to "A" (the default option).

The input data for "Case A" of the GRfit function must be in this format with columns named "concentration", "cell_count", "cell_count__ctrl", and "cell_count__time0" as well as columns for other key variables. The columns "cell_line", "agent", "perturbation", "replicate", and "time" are simply examples of these key variables. It is not necessary that your input data frame include these exact column names or the same number of columns.

Usage

inputCaseA

Format

A data frame with 1008 rows and 9 variables:

- cell_line: the cell-line used (MCF10A, MCF7, BT20)
- agent: the drug used (drugA, drugB, drugC, drugD)
- perturbation: an example of another key variable (e.g. different media, a co-treatment, etc.) (0, 1)
- replicate: replicate number, (1, 2, 3)
- time: time of assay measured in hours (48, 72)
- concentration: concentration of the perturbagen on which dose-response curves will be evaluated (not log transformed)
Input Case C

- cell_count: the measure of cell number (or a surrogate of cell number) after treatment at the end of the assay
- cell_count__ctrl: the measure of cell number in control (e.g. untreated or DMSO-treated) wells from the same plate at the end of the assay
- cell_count__time0: the measure of cell number in untreated wells grown in parallel until the time of treatment

Value

An example dataset in the form of "Case A" generated for use with GRfit

Source

/inst/scripts/generate_data.py

Description

This is an example dataset corresponding to Case C. The dataset is equivalent to that of Case A, but in a different form. The data was generated by the python script "generate_data.py" found in the "inst/scripts/" directory. This object is meant to be used as the "inputData" argument for the GRfit function when "case" is equal to "C". The input data for "Case C" of the GRfit function must be in this format with columns named "concentration", "cell_count", and "time" as well as columns for other key variables. The columns "cell_line", "agent", "perturbation", and "replicate", are simply examples of these key variables. It is not necessary that your input data frame include these exact column names or the same number of columns.

Usage

inputCaseC

Format

A data frame with 1352 rows and 7 variables:

- cell_line: the cell-line used (MCF10A, MCF7, BT20)
- agent: the drug used (-, drugA, drugB, drugC, drugD)
- perturbation: an example of another key variable (e.g. different media, a co-treatment, etc.) (0, 1)
- replicate: replicate number, (1, 2, 3)
- time: time of assay measured in hours (0, 48, 72)
- concentration: concentration of the perturbagen on which dose-response curves will be evaluated (not log transformed)
- cell_count: measure of cell number or a surrogate of the number of cells.

Value

An example dataset in the form of "Case C" generated for use with GRfit
Source

inst/scripts/generate_data.py
Index

*Topic datasets
  inputCaseA, 12
  inputCaseC, 13

drm, 6
drmc, 6

GRbox, 2, 4, 6, 12
GRdrawDRC, 3, 3, 6, 12
GRfit, 2–4, 4, 7–13
GRgetDefs, 7
GRgetGroupVars, 8
GRgetMetrics, 9
GRgetValues, 10
GRscatter, 3, 4, 6, 11

inputCaseA, 5, 12
inputCaseC, 5, 13