GRmetrics: an R package for calculation and visualization of Growth Rate Metrics

Nicholas Clark

30 August 2016

Background

Quantifying drug response is the cornerstone of many pharmacological experiments ranging from pharmacogenomics studies to small-scale analyses of drug resistance. In general, cells are grown in the presence or absence of drugs for a few days and the endpoint cell count values (or a surrogate) is compared. From the relative cell count, metrics of drug sensitivity such as IC50 or Emax values are evaluated. In cases where the untreated control cells grow over the course of the assay, these traditional metrics are confounded by the number of divisions that take place over the course of an assay. In particular, for drugs that impact growth rate and block cell division, slow-growing cell lines will appear more resistant than fast-growing lines although the biological effect on a per-division basis may be the same.

Hafner et al. recently proposed alterative drug-response metrics based on growth rate inhibition (GR metrics) that are robust to differences in nominal division rate and assay duration. Using these metrics requires only to know the number of cells (or a surrogate) at the time of treatment; note that this value may also be inferred from the nominal division rate and the value of an untreated sample at the endpoint.

To facilitate the use of these GR metrics, we have developed an R package that provides functions to analyze and visualize drug response data with these new metrics across multiple conditions and cell lines.

Installation

The **GRmetrics** package can be installed through Bioconductor

```
source("http://bioconductor.org/biocLite.R")
biocLite("GRmetrics")
### if this doesn't work, use "devel" version of Bioconductor ###
library(biocInstaller)
useDevel()
```

References:

Hafner, M., Niepel, M. Chung, M. and Sorger, P.K., *Growth Rate Inhibition Metrics Correct For Confounders In Measuring Sensitivity To Cancer Drugs.* Nature Methods 13.6 (2016): 521-527. (http://dx.doi.org/10.1038/nmeth.3853)

Corresponding MATLAB and Python scripts available on repo:

https://github.com/sorgerlab/gr50_tools.

Note: Most, but not all of these scripts have been reproduced in this R package. Namely, this package does not contain code for "Case B" of the MATLAB scripts nor does it contain an R script to generate the example input data. The python script for generating the example data can be found in the "inst/scripts/" directory.

Browser interface and online tools: http://www.grcalculator.org

Much of the description below has been adapted from:

https://github.com/sorgerlab/gr50_tools/blob/master/README.md.

Input data

The main function of the package is **GRfit**, which takes in a data frame containing information about concentration, cell counts over time, and additional grouping variables for a dose-response assay and calculates growth-rate inhibition (GR) metrics for each experiment in the dataset.

There are two cases of data input accepted by the **GRfit** function. They are described in detail below. Case "A" is the default option.

Case A: a single file with control values assigned to treated measurements

The control values (both control and time 0 cell counts) are pre-computed by the user and assigned to each treatment (row) in appropriate columns in the input file. Control cell counts should be in a column labeled *cell_count___ctrl* and the time 0 cell counts in a column labeled *cell_count___time0*.

An example input data frame for "Case A", named "inputCaseA", is contained within the package. To access it, use the following code:

```
library(GRmetrics)
data(inputCaseA)
```

The mandatory inputs for Case "A" are:

- **inputData** - the name of an input data frame with the following columns 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*)

Case C: a single file with control values stacked with treated measurements

In the most general case, the control cell counts are in the same file and format as the treated cell counts. Control cell counts will be averaged (using a 50%-trimmed mean) and automatically matched to the treated cell counts based on the keys (columns in the data file). The control cell count values must have a value of 0 for *concentration* and a value for *time* that matches the treated measurements. The time 0 cell count values must have value of 0 for *time*. If the structure of the data is complex, the provided scripts may inappropriately match control and treated cell counts, so users instead should format their data as described in case A.

An example input data frame for "Case C", named "inputCaseC", is contained within the package. To access it, use the following code:

```
library(GRmetrics)
data(inputCaseC)
```

The mandatory inputs for Case "C" are:

- **inputData** - the name of an input data frame with the following columns 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*)

Functions

The package contains 3 visualization functions: **GRdrawDRC**, **GRscatter**, and **GRbox**.

All of these functions take in an object created by **GRfit** as well as additional arguments. The results can be viewed in a static ggplot image or an interactive plotly (turned on/off by the *plotly* parameter).

- **GRdrawDRC** this function draws the (growth-rate inhibition) dose-response curve using the parameters calculated by the **GRfit** function. If *points* is set to TRUE, then it will also plot the points used to fit each curve.
- **GRscatter** this function draws a scatterplot of a given GR metric (GR50, GRmax, etc.) with the *xaxis* value(s) plotted against the *yaxis* value(s).
- **GRbox** this function draws boxplots of a given GR metric (GR50, GRmax, etc.) for values of a given grouping variable. It overlays the points used to make these boxplots and can color them according to another grouping variable.

Examples

Load example data (Case A)

```
data(inputCaseA)
```

```
head(inputCaseA)
```

```
##   cell_line agent perturbation replicate time concentration cell_count
## 1    MCF10A drugA            0         1   48      0.001000       1131
## 2    MCF10A drugA            0         1   48      0.003162       1205
## 3    MCF10A drugA            0         1   48      0.010000       1021
## 4    MCF10A drugA            0         1   48      0.031620        743
## 5    MCF10A drugA            0         1   48      0.100000        459
## 6    MCF10A drugA            0         1   48      0.316200        318
##   cell_count__ctrl cell_count__time0
## 1           1212.5             299.5
## 2           1212.5             299.5
## 3           1212.5             299.5
## 4           1212.5             299.5
## 5           1212.5             299.5
## 6           1212.5             299.5
```

Calculate GR values and solve for GR metrics parameters (i.e. fit curves)

```
drc_output = GRfit(inputCaseA, groupingVariables = c('cell_line','agent'))
```

See overview of output data (SummarizedExperiment object)

```
drc_output
```

```
## class: SummarizedExperiment
## dim: 9 12
## metadata(2): '' ''
## assays(1): ''
## rownames(9): GR50 GRmax ... pval flat_fit
## rowData names(2): Metrics Description
## colnames(12): BT20 drugA BT20 drugB ... MCF7 drugC MCF7 drugD
## colData names(4): cell_line agent fit experiment
```

Review output table of GR metrics parameters

```
assay(drc_output)
```

```
##             BT20 drugA   BT20 drugB    BT20 drugC    BT20 drugD
## GR50      9.080477e-02          Inf  2.480054e-01  2.215168e-02
## GRmax    -1.143156e-02  0.885341630 -7.751600e-01 -1.348661e-01
## GR_AOC    5.092626e-01  0.022095089  5.622640e-01  6.764782e-01
## GEC50     8.715113e-02  0.000000000  5.851218e-01  2.363744e-02
## GRinf     2.267627e-02  0.975788529 -7.853376e-01 -4.032884e-02
## Hill      1.130151e+00  0.010000000  1.099956e+00  1.194882e+00
## r2        9.686214e-01 -0.004378014  9.774727e-01  9.768774e-01
## pval      2.096868e-80  1.000000000  4.938309e-88  1.967703e-87
## flat_fit            NA  0.975788529            NA            NA
##           MCF10A drugA  MCF10A drugB MCF10A drugC MCF10A drugD
## GR50      3.964474e-02  1.735020e+00 1.226689e+00 2.363344e-01
## GRmax    -1.130333e-01 -3.456799e-02 1.670275e-01 3.459029e-01
## GR_AOC    6.255923e-01  2.207611e-01 2.040813e-01 2.959000e-01
## GEC50     4.496553e-02  2.414599e+00 9.816886e-01 6.853793e-02
## GRinf    -6.702646e-02 -1.876440e-01 2.057860e-01 4.537141e-01
## Hill      9.988893e-01  9.641441e-01 2.380163e+00 1.922488e+00
## r2        9.866108e-01  9.713640e-01 9.715094e-01 9.623065e-01
## pval      2.731585e-66  9.815911e-55 8.213775e-55 1.476783e-50
## flat_fit            NA            NA           NA           NA
##            MCF7 drugA    MCF7 drugB    MCF7 drugC   MCF7 drugD
## GR50     8.610127e-02  1.892181e-01  8.907388e-02          Inf
## GRmax    1.077187e-01 -7.884164e-01 -8.887942e-01 4.375892e-01
## GR_AOC   4.499196e-01  6.314117e-01  8.867484e-01 1.568364e-01
## GEC50    6.645387e-02  3.674380e-01  1.300352e-01 4.506807e-01
## GRinf    1.631132e-01 -7.775313e-01 -8.786271e-01 5.355470e-01
## Hill     1.524465e+00  1.413502e+00  2.680758e+00 2.105139e+00
## r2       9.722825e-01  9.916603e-01  9.951465e-01 9.002763e-01
## pval     3.135971e-55  1.738556e-73  1.028104e-81 9.077155e-36
## flat_fit           NA            NA            NA           NA
```

View details of each experiment

```
colData(drc_output)
```

```
## DataFrame with 12 rows and 4 columns
##                cell_line       agent         fit   experiment
##              <character> <character> <character>  <character>
## BT20 drugA          BT20       drugA     sigmoid   BT20 drugA
## BT20 drugB          BT20       drugB        flat   BT20 drugB
## BT20 drugC          BT20       drugC     sigmoid   BT20 drugC
```

```
## BT20 drugD          BT20       drugD     sigmoid   BT20 drugD
## MCF10A drugA      MCF10A       drugA     sigmoid MCF10A drugA
## ...                  ...         ...         ...          ...
## MCF10A drugD      MCF10A       drugD     sigmoid MCF10A drugD
## MCF7 drugA          MCF7       drugA     sigmoid   MCF7 drugA
## MCF7 drugB          MCF7       drugB     sigmoid   MCF7 drugB
## MCF7 drugC          MCF7       drugC     sigmoid   MCF7 drugC
## MCF7 drugD          MCF7       drugD     sigmoid   MCF7 drugD
```

View descriptions of each GR metric (or goodness of fit measure)

```
View(rowData(drc_output))
```

Review output table of GR values

```
head(metadata(drc_output)[[1]])
```

```
##   cell_line agent perturbation replicate time concentration cell_count
## 1    MCF10A drugA            0         1   48      0.001000       1131
## 2    MCF10A drugA            0         1   48      0.003162       1205
## 3    MCF10A drugA            0         1   48      0.010000       1021
## 4    MCF10A drugA            0         1   48      0.031620        743
## 5    MCF10A drugA            0         1   48      0.100000        459
## 6    MCF10A drugA            0         1   48      0.316200        318
##   cell_count__ctrl cell_count__time0 log10_concentration         GR
## 1           1212.5             299.5          -3.0000000 0.93219264
## 2           1212.5             299.5          -2.5000381 0.99385806
## 3           1212.5             299.5          -2.0000000 0.83663626
## 4           1212.5             299.5          -1.5000381 0.56891172
## 5           1212.5             299.5          -1.0000000 0.23569217
## 6           1212.5             299.5          -0.5000381 0.03015641
##     experiment
## 1 MCF10A drugA
## 2 MCF10A drugA
## 3 MCF10A drugA
## 4 MCF10A drugA
## 5 MCF10A drugA
## 6 MCF10A drugA
```

View grouping variables used for calculation

```
metadata(drc_output)[[2]]
```

```
## [1] "cell_line" "agent"
```

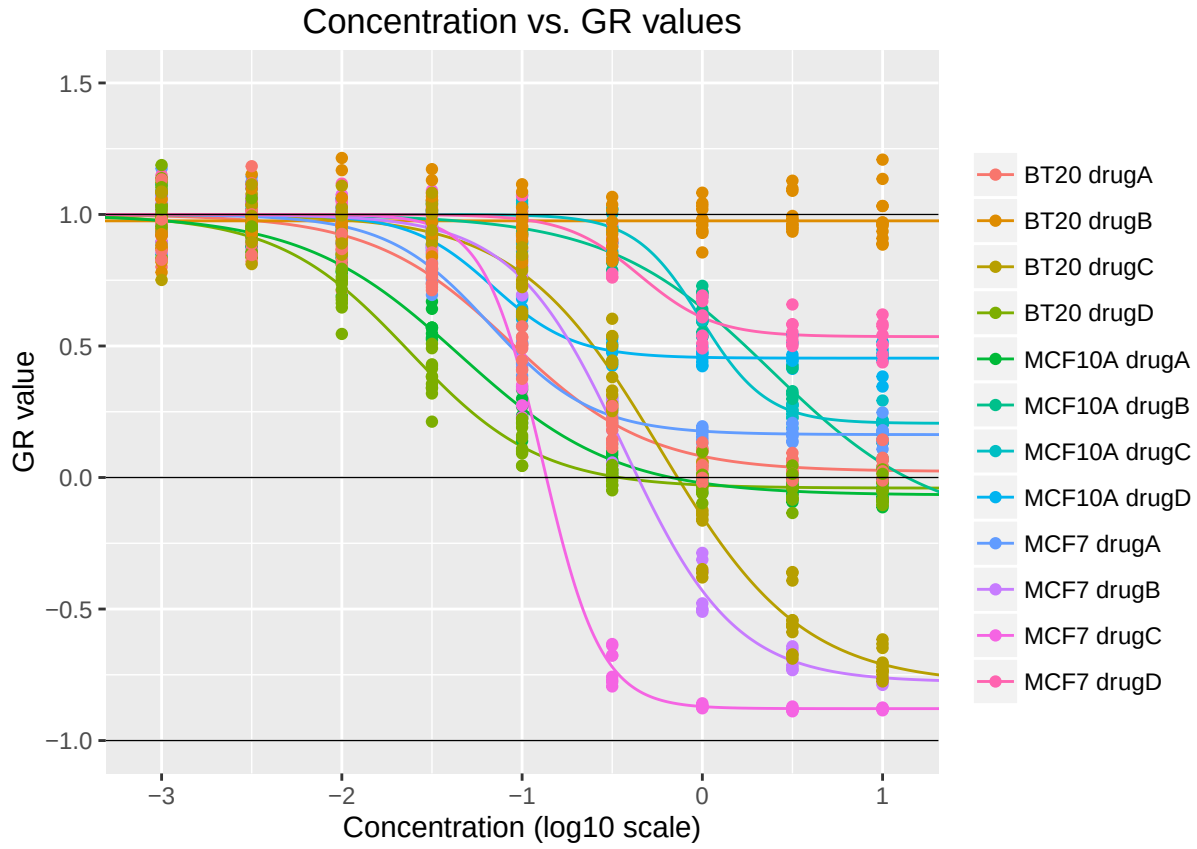
You can also export your results. Here are two examples:

```
# Write GR metrics parameter table to tab-separated text file
write.table(assay(drc_output), file = "filename.tsv", quote = FALSE,
sep = "\t", row.names = FALSE)
# Write original data plus GR values to comma-separated file
write.table(metadata(drc_output)[[1]], file = "filename.csv", quote = FALSE,
sep = ",", row.names = FALSE)
```

Visualizations

You can draw GR dose-response curves with plotly or with ggplot2. You can also specify the range of the graph.

```
# Draw dose-response curves
GRdrawDRC(drc_output)
GRdrawDRC(drc_output, experiments = c('BT20 drugA', 'MCF10A drugA',
                                      'MCF7 drugA'))
GRdrawDRC(drc_output, experiments = c('BT20 drugA', 'MCF10A drugA',
                                      'MCF7 drugA'),
          min = 10^(-4), max = 10^2)
GRdrawDRC(drc_output, plotly = FALSE)
```



You can also draw scatterplots and boxplots of GR metrics with plotly or ggplot2. Here is an example using example data in the format of Case C.

```
## Case C (scatterplot and boxplot examples)
data(inputCaseC)
```

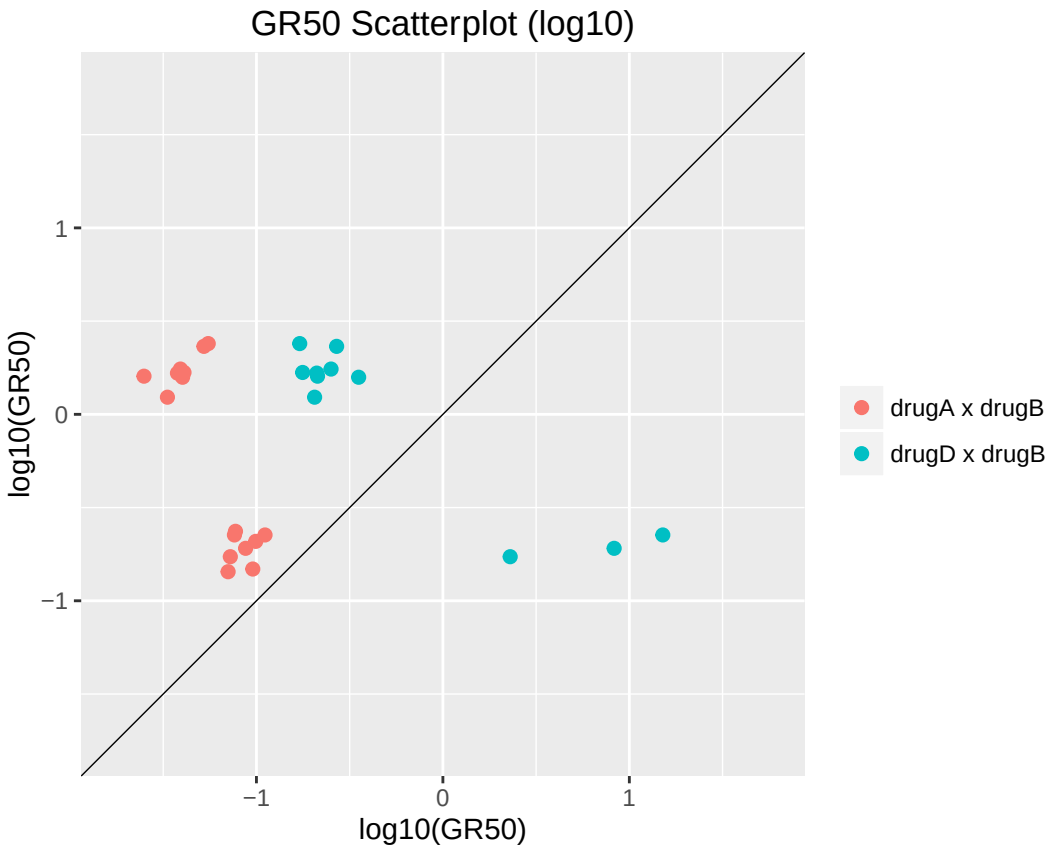
```
head(inputCaseC)
```

```
##   cell_line agent perturbation replicate time concentration cell_count
## 1    MCF10A     -            0       NaN    0             0        294
## 2    MCF10A     -            0       NaN    0             0        318
## 3    MCF10A     -            0       NaN    0             0        287
## 4    MCF10A     -            0       NaN    0             0        296
## 5    MCF10A     -            0       NaN    0             0        291
```

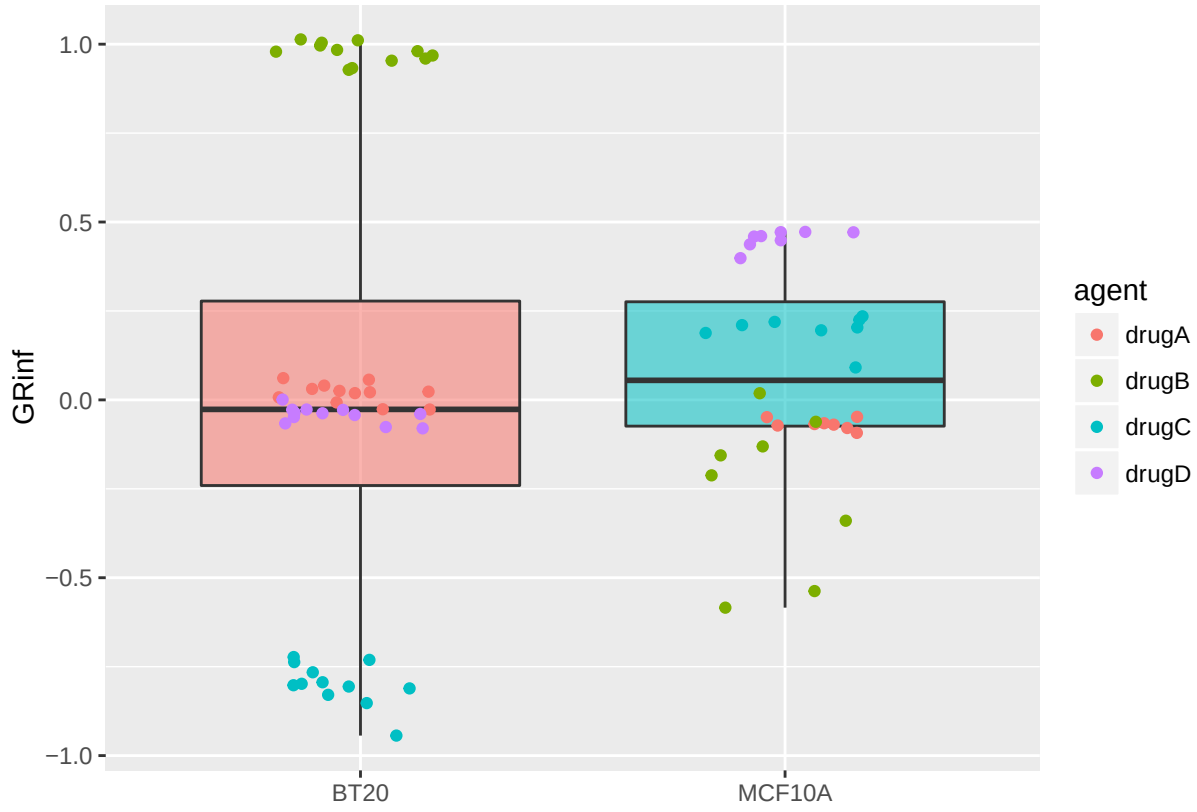
```
## 6    MCF10A     -            0       NaN    0             0        286
```

```
output1 = GRfit(inputData = inputCaseC, groupingVariables =
c('cell_line','agent', 'perturbation', 'replicate', 'time'), case = "C")
```

```
# Draw scatterplots
GRscatter(output1, 'GR50', 'agent', c('drugA','drugD'), 'drugB')
GRscatter(output1, 'GR50', 'agent', c('drugA','drugD'), 'drugB',
          plotly = FALSE)
```




```
# Draw boxplots
GRbox(output1, GRmetric ='GRinf', groupVariable = 'cell_line',
      pointColor = 'agent')
GRbox(output1, GRmetric ='GRinf', groupVariable = 'cell_line',
      pointColor = 'agent',
      factors = c('BT20', 'MCF10A'))
GRbox(output1, GRmetric ='GRinf', groupVariable = 'cell_line',
      pointColor = 'agent',
      factors = c('BT20', 'MCF10A'), plotly = FALSE)
```



GR metric details

We have developed scripts to calculate normalized growth rate inhibition (GR) values and corresponding metrics (GR50, GRmax, ...) based on cell counts measured in dose-response experiments. Users provide a tab-separated data file in which each row represents a separate treatment condition and the columns specify the keys that define the treatment condition (e.g. cell line, drug or other perturbagen, perturbagen concentration, treatment time, replicate) and the measured cell counts (or surrogate). The experimentally measured cell counts that are required for GR metric calculation are as follows: - measured cell counts after perturbagen treatment (*cell_count*, *x(c)*) - measured cell counts of control (e.g. untreated or DMSO-treated) wells on the same plate (*cell_count___ctrl*, *x_ctrl*) - measured cell counts from an untreated sample grown in parallel until the time of treatment (*cell_count___time0*, *x_0*)

The provided GR scripts compute over the user's data to calculate GR values individually for each treatment condition (cell line, time, drug, concentration, ...) using the formula:

```
GR(c) = 2 ^ ( log2(x(c)/x_0) / log2(x_ctrl/x_0) ) - 1
```

Based on a set of GR values across a range of concentrations, the data are fitted with a sigmoidal curve:

```
GR(c) = GRinf + (1-GRinf)/(1 + (c/(GEC50))^Hill )
```

The following GR metrics are calculated: - **GRinf** = GR(c->inf), which reflects asymptotic drug efficacy. - Hill coefficient of the sigmoidal curve (**Hill**), which reflects how steep the dose-response curve is. - **GEC50**, the drug concentration at half-maximal effect, which reflects the potency of the drug. - **GR50**, the concentration at which the effect reaches a GR value of 0.5 based on interpolation of the fitted curve. - **GR_AOC**, the area over the dose-response curve, which is the integral of *1-GR(c)* over the range of concentrations tested,

normalized by the range of concentration. - **GRmax**, the effect at the highest tested concentration. Note that *GRmax* can differ from *GRinf* if the dose-response does not reach its plateau value.

In addition, the scripts report the r-squared of the fit and evaluate the significance of the sigmoidal fit based on an F-test. If the fit is not significant ($p < 0.05$, or any arbitrary value), the sigmoidal fit is replaced by a constant value (flat fit). The cutoff value for the p-value can be set above 1 for bypassing the F-test. Additional information and considerations are described in the supplemental material of the manuscript referred above.