Tools and scripts to evaluate GR values and metrics

Companion of the manuscript:
Metrics of drug sensitivity and resistance based on growth rate inhibition correct for the confounding effects of variable division rates

References:

Hafner, M., Niepel, M. Chung, M. and Sorger, P.K., *Metrics of drug sensitivity and resistance based on growth rate inhibition correct for the confounding effects of variable division rates*, in revision, doi:####

Scripts available on repo https://github.com/sorgerlab/gr50_tools

Browser interface and online tools: (in preparation)

General approach

We have developed scripts to calculate normalized growth rate inhibition (GR) values and corresponding metrics (GR_{50}, GR_{max}, …) 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 (control cell count, x_{ctrl})
- measured cell counts from an untreated sample grown in parallel until the time of treatment (time 0 cell count, 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:

$$GR(c) = 2^{\frac{\log_2(x(c)/x_0)}{\log_2(x_{ctrl}/x_0)}} - 1$$

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

$$GR(c) = GR_{inf} + \frac{1 - GR_{inf}}{1 + \left(\frac{c}{EC_{50}}\right)^{h_{GR}}}$$

The following GR metrics are calculated:

- GR_{inf} = GR(c $\rightarrow$ inf), which reflects asymptotic drug efficacy.
- Hill coefficient of the sigmoidal curve (h_{GR}), which reflects how steep the dose-response curve is.
- EC_{50}, the drug concentration at half-maximal effect, which reflects the potency of the drug.
- GR_{50}, the concentration at which the effect reaches a GR value = 0.5 based on interpolation of the fitted curve.
- AUC, the area under the dose-response curve, which is the integral of $GR(c)$ over the range of concentrations tested.
- GR_{max}, the effect at the highest tested concentration. Note that GR_{max} can differ from GR_{inf} 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 ($p < 0.05$), the sigmoidal fit is replaced by a constant value (flat fit).

Additional information and considerations are described in the supplemental material of the manuscript referred above.
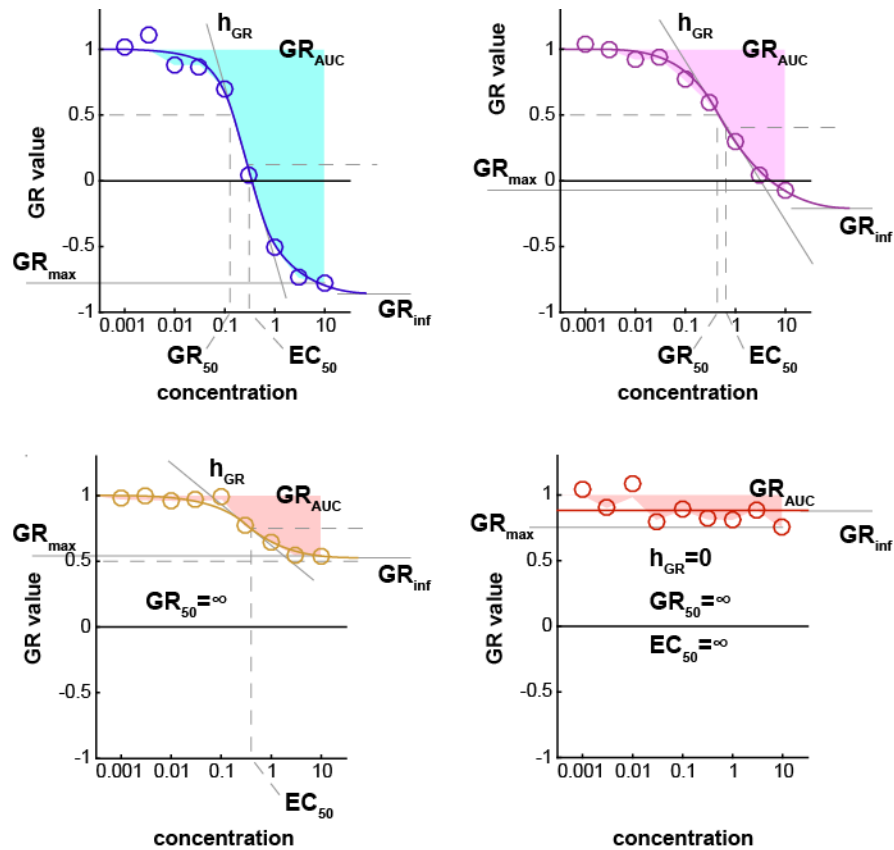
Examples of dose-response curves and fits. The upper panels depict strong responses to drugs for which all sensitivity parameters can be defined. In contrast, in the case shown in the lower left panel GR_{inf} is above 0.5, so GR_{50} cannot be defined (and thus is set to ∞). In the case shown in the lower right panel, the response is weak and noisy, so the sigmoidal fit is not significant and a straight flat line is fitted. As a result, only GR_{AUC} and GR_{inf} can be defined.

Input files

The scripts support three different types of inputs. The input file(s) must be tab-separated files (.tsv) and must have the following column headers (first row of the file):

- 'concentration'   concentration of the perturbagen on which dose-response curves will be evaluated
- 'cell_count'   measure of cell number or a surrogate of the number of cells.
- 'time'   mandatory only for input type 'C' (see below)
- control values   mandatory only for input type 'A' (see below)

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

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'.

This case corresponds to the toy example 1 in the GitHub folder.

Case B: three files with control values labelled with a key

The control values (both control and time 0 cell counts) are in two separate files, and the response data are in a third file. The treated cell counts are matched to the control cell counts based on key columns found in both the response data file and the control value files. Across the different files, the column with header 'ctrl_tag' matches the control cell counts to the appropriate treated cell counts, and the column with header 'time0_tag' matches the time 0 cell counts. If the control or time 0 cell count files contain multiple rows with the same key, the values will be averaged (using a 50%-trimmed mean).

This case corresponds to the toy examples 2 and 3 in the GitHub folder. Example 3 is more general, as it contains multiple values per key.

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 or B.

Case C corresponds to the toy example 4 in the GitHub folder.

Scripts for calculation of GR values and metrics

MATLAB implementation

The general MATLAB function is:

```
[t_GRvalues, t_GRmetrics] =
            GRmetrics(output, input_data, input_ctrl, input_time0)
```

Input variables:

- output: folder or tag for the output files (empty means that no output file will be written)
- input_data: file name for the data on which to compute the GR values (see details above)
- input_ctrl: file name for the control data; only for case B (see above)
- input_time0: file name for the time 0 data; only for case B (see above)

Output variables are tables:

- t_GRvalues contains the GR values for all treated cell count measurements.
- t_GRmetrics contains the results of the sigmoidal fit across concentration for each unique set of keys found in the data. Columns of the t_GRmetrics table list the keys and the fitted parameters and values ('GR50', 'GRmax', 'GR_AUC', 'EC50', 'GRinf', 'Hill', 'r2', and 'pval'; see details above).

The MATLAB sub-functions are processing MATLAB tables as follow:

- `add_controls.m`: merge multiple tables to handle case B).
- `assign_controls.m`: identify the controls and assign them to treatment as for case C).
- `evaluate_GRvalues.m`: calculate GR value on formatted MATLAB table.
- `evaluate_GRmetrics.m`: calculate GR metrics based on GR value.

Python

The python function to calculate the GR value is:

```
add_gr_column.py input.tsv > output.tsv
```

This covers case A, and the input.tsv file must:

- have column headers in the first row
- have keys in columns that precede the data columns
- contain numeric columns with the headers 'cell_count', 'cell_count__ctrl', and 'cell_count__time0'

Example in GitHub

The GitHub repo contains an example generated by the generate_data.py. It is based on artificial drug-response data across a combination of 3 cell lines, 4 drugs, 2 perturbations (that affect cell growth but not sensitivity), 2 time points and 2 or 3 replicates (one cell line has more replicate). The data are in written in 4 different tables that cover each of the cases described above (2 examples for case B).