ICSB Workshop: Drug Response Measurement and Analysis

Part 2: Best practices for experimental design, execution, and analysis

Caitlin Mills
Kartik Subramanian
Drug-response experiments are becoming increasingly high-throughput.

96 or 384 well plates
Design example: testing 2 drugs across multiple doses in 2 conditions

96-well plate
C
D
E
F

Condition 1
Untreated controls for condition 1

Doses of treatment 1
Doses of treatment 2

Condition 2
Untreated controls for condition 2

Doses of treatment 1
Doses of treatment 2
What does the pattern in the output response suggest?
Edge effects warrant randomization
Randomizing the position on the plate avoids biases and artefacts.
Spreadsheets are error-prone and disconnected

### Drug concentration

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.31</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0.31</td>
<td>1</td>
<td>0.31</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0.31</td>
<td>3</td>
<td>0.1</td>
</tr>
</tbody>
</table>

### Drug names

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D_1</td>
<td>D_5</td>
<td>D_4</td>
</tr>
<tr>
<td>2</td>
<td>D_4</td>
<td>D_1</td>
<td>D_2</td>
</tr>
<tr>
<td>3</td>
<td>D_5</td>
<td>D_1</td>
<td>D_2</td>
</tr>
<tr>
<td>4</td>
<td>D_4</td>
<td>D_3</td>
<td>D_3</td>
</tr>
</tbody>
</table>

### Cell lines

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HeLa</td>
<td>HeLa</td>
<td>HeLa</td>
</tr>
<tr>
<td>2</td>
<td>MCF7</td>
<td>MCF7</td>
<td>MCF7</td>
</tr>
<tr>
<td>3</td>
<td>DU145</td>
<td>DU145</td>
<td>DU145</td>
</tr>
<tr>
<td>4</td>
<td>A375</td>
<td>A375</td>
<td>A375</td>
</tr>
</tbody>
</table>

### Experimental design long table

<table>
<thead>
<tr>
<th>Well</th>
<th>Cell Line</th>
<th>Drug</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>HeLa</td>
<td>D_1</td>
</tr>
<tr>
<td>B1</td>
<td>HeLa</td>
<td>D_5</td>
</tr>
<tr>
<td>C1</td>
<td>HeLa</td>
<td>D_4</td>
</tr>
<tr>
<td>A2</td>
<td>MCF7</td>
<td>D_4</td>
</tr>
<tr>
<td>B2</td>
<td>MCF8</td>
<td>D_1</td>
</tr>
<tr>
<td>C2</td>
<td>MCF9</td>
<td>D_2</td>
</tr>
</tbody>
</table>
Merging experimental design with measurements

<table>
<thead>
<tr>
<th>Well</th>
<th>Cell Line</th>
<th>Drug</th>
<th>Concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>HeLa</td>
<td>D_1</td>
<td>0.1</td>
</tr>
<tr>
<td>B1</td>
<td>HeLa</td>
<td>D_5</td>
<td>0.31</td>
</tr>
<tr>
<td>C1</td>
<td>HeLa</td>
<td>D_4</td>
<td>3</td>
</tr>
<tr>
<td>A2</td>
<td>MCF7</td>
<td>D_4</td>
<td>0.31</td>
</tr>
<tr>
<td>B2</td>
<td>MCF7</td>
<td>D_1</td>
<td>1</td>
</tr>
<tr>
<td>C2</td>
<td>MCF7</td>
<td>D_2</td>
<td>3</td>
</tr>
</tbody>
</table>

Measurement file

<table>
<thead>
<tr>
<th>Well</th>
<th>Cell count</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>2500</td>
</tr>
<tr>
<td>B1</td>
<td>3168</td>
</tr>
<tr>
<td>C1</td>
<td>2110</td>
</tr>
<tr>
<td>A2</td>
<td>5673</td>
</tr>
<tr>
<td>B2</td>
<td>4389</td>
</tr>
<tr>
<td>C2</td>
<td>1290</td>
</tr>
</tbody>
</table>
Steps to achieve reliable experimental measurements

Manual layout of drugs on a plate

Randomized Assay Plates
Using high-dimensional data containers in the design, storage, and analysis of drug-response experiments.
Additional notes: types of variables

• Model variables:
  – Treatment variables (drug, concentration, …)
  – Condition variables (growth media, seeding density, …)

• Confounder variables:
  – Plate model
  – Assay date

• Readout variables
**Pipeline**

1. Designing experiment
   - Running the experiment

2. Quality control

3. Evaluating sensitivity metrics

Hafner*, Niepel*, Subramanian*, Sorger
Experimental Design

Protocol 1: Designing experiment

1. Treatment list
2. Plate list
3. Well annotations
   → Plate information
   → Treatment file
      (D300/pin transfer)
Details about specific example

- 30 drugs
- 9 doses
- 5 cell lines
- 2 time points
- 3 replicates

Plate layout:
- 30 drugs and 9 doses
- 3 replicates
- 5 cell lines
- 2 time points
Template for specifying the experimental design.

The compounds, number of doses and information about the role of each compound (treatment, negative control etc) is defined in the file "compound_list.tsv". The scripts below take this tsv file as input in order to design the layout on the plate.

The size of the plate has to be provided as number of rows and columns. The number of replicates and the plate barcode are also provided in the block of code below.

Design of the experiment and treatment layout (protocol 1)

```python
import datarail.experimental_design.process_assay as process_assay
import datarail.experimental_design.designer as designer
import datarail.experimental_design.plot_panels as design_plot
import matplotlib.pyplot as plt
%matplotlib inline
input_file = 'INPUT/compound_list.tsv'
plate_dims = [16,24]
fingerprint_prefix = 'DRUG_TRT_
num_replicates = 3
treatment_dicts = process_assay.read_input(input_file, plate_dims, fingerprint_prefix, encode_plate=True, num_replicates=num_replicates)
```

There are 20 untreated wells on the inner plate. Consider allotting more wells to negative controls.
Use Jupyter notebooks to keep track of design steps and export drug layout

```python
Designs = designer.make_layout(treatment_dicts, fingerprint_prefix,
    encode_fingerprint=True,
    plate_dims=plate_dims, num_replicates=num_replicates,
    randomize=True, biased_randomization=True)

design_plot.plot_layout(Designs.sel(plates='DRUG_TRT_A'))
design_plot.plot_drug(Designs.sel(plates='DRUG_TRT_A'), 'D_1')
```
Basic experimental workflow

• Grow (happy) cells
• Seed cells at appropriate densities in multi-well plates
• Deliver drugs to multi-well plates
• Stain and fix cells
• Image cells
• Extract quantitative data from images
To consider *before* you start

• How many cell lines do I want to test?
  – Are they amenable to imaging?
    • Are they adherent? Do they grow in a monolayer?
    – How densely should they be seeded?

• How many drugs do I want to collect dose response data for?
  – Are they DMSO soluble?
  – What’s an appropriate dose range?
Cell seeding

- Seed plates at an appropriate density
- Use automation if possible
- Barcode plates to keep track of them
Cell seeding density influences growth rate...
...which influences the dose response.

![Graphs showing concentration response curves for different division times.](image)
Division rate differs across densities

Seeding density affects the number of divisions.

→ $IC_{50}$ and $E_{\text{max}}$ are correlated with density.

[Graph showing correlation between seeding number and division rate, $IC_{50}$, and $E_{\text{max}}$ across different cell lines.]
Drug delivery via pin transfer

• For simultaneous delivery of many drugs
• For large scale experiments (many cell lines, conditions)
• Facilitates reproducibility
Drug delivery via digital drug dispenser

- For accurate delivery of a few drugs
- Pilot experiments - to identify appropriate doses
- Follow-up experiments, ‘hit’ validation
- Drugs that cannot be prepared in DMSO
Drug delivery via manual pipetting

1200 μL
1000 nM drug in media

Transfer 700 μL
400 μL media
400 μL media
400 μL media
400 μL media
400 μL media
Drug delivery via manual pipetting

1000 nM drug in media
636 nM drug
405 nM drug
258 nM drug
164 nM drug
104 nM drug

\[ \left( \frac{700}{700 + 400} \right)^5 = 0.104 \]
Other considerations

• Artefacts
  – Edge effects
    • Exclude outer wells
    • Use humidified secondary containers
    • Some cell lines are more sensitive than others
    • Depends on the duration of the experiment
  – Systematic bias from automation

• Randomization helps!
Dye-drop assay reagents

- Minimally-disruptive, reagent-sparing cell staining and fixation protocol
Dye-drop assay protocol

- Stain: Hoechst + LDR in 10% optiprep in PBS
- Fix: 4% formaldehyde in 20% optiprep in PBS
Plate washer

- Uniform and controlled aspiration and liquid dispensing

- Is repeat washing really that bad?
Repeat washing can result in cell loss...

No wash

PBS wash x 1

PBS wash x 2
...that can bias your results
Image acquisition

- Operetta microscope with plate hotel, barcode reader & robot
  - Automated data collection for 40+ plates
Image acquisition

Imaging 6 fields of view @ 10x captures *almost* the entire well
Image acquisition
1. Segment nuclei
2. Measure LDR signal
3. Classify live/dead cells
Can I just count cells?
Strengths and limitations

• Imaging based
  – Best suited for adherent cells that grow in monolayer culture
• Image analysis can be time consuming
• Can go back and visually inspect imaging data
• Potential for multiplexing, immunofluorescence
• Fate of live cells unknown
• Reagent sparing
• Distinction between cytotoxic and cytostatic effects
Other assays

• CellTiter-Glo etc.
  – Simple, no wash protocol
  – Luminescence read-out, simple analysis, rapid results
  – Treatment-induced changes in metabolic activity of cells can skew results

• Measurement of confluency
  – Inaccurate
  – Treatment-induced changes in morphology can skew results
Take away messages

• Include a t=0 plate
• Optimize conditions
  – Seeding density per cell line
  – Dose range per drug
  – Duration of assay
• Automate as much as possible
Processing

Well annotations

Plate information

Microscope or scanner

Unprocessed result files

Protocol 2: Processing data files

1. Processed file

2. Annotated file

3. Quality control
Use Jupyter notebooks to import and annotate results from experiments

Data import and annotation (protocol 2, steps 1 and 2)

```python
# load the synthetic data (Columbus output)
df=CoiMp.Columbus_processing('..tests/drug_response_data/OUTPUT/Example1_Columbus_output.tsv',
    (('Hoechst_pos', 'cell_count_total'),
    ('LDR_pos_Hoechst_neg', 'corpse_count'),
    ('Hoechst_LDR_pos', 'cell_count_dead'),
    'cell_count_total - cell_count_dead'))

# annotate the data
df_annnotated = TrtAnnot.add_treatments(TrtAnnot.add_plate_info(df,
    pd.read_csv('..tests/drug_response_data/OUTPUT/Example1_plate_info.tsv', sep='\t')),
    '..tests/drug_response_data/OUTPUT/'
)
df_annnotated.to_csv('OUTPUT/AnnotatedData_Example1.tsv', sep='\t', index=False)
```

Default number of fields: 6 ; 505 wells with missing field(s)
Concentrations rounded in the log domain
Check for unwanted biases using embedded functions

Quality control (protocol 2, step 3)

```python
import datarail.data_processing.drug_response.qc_plate as qcfct

# use the raw data to perform the plate QC
qcfct.Plate_bias(pltfct.dfplate2x(df), filename='OUTPUT/QC_report_Example1.pdf')

# use the annotated data to perform QC based on the negative controls
qcfct.Negative_control_bias(df_annotated)
```

QC report file

Plate bias QC

Negative control QC
Analysis: data normalization and dose-response curve parametrization

Protocol 3:
Evaluating sensitivity metrics

1. Annotated file
2. Normalized file
   - Dose-response plots
   - Sensitivity metrics
Normalize the data to obtain the GR values

**Calculate the GR values (protocol 3, step 1)**

```python
# first calculate the GR values for each replicate then merge them
df_mean = TrtAnnot.average_replicates(
    gr50.compute_gr(
        gr50.assign_ctrls(df_annotated, ['cell_line']))
)
df_gr = df_mean.drop(['cell_count_dead', 'corpse_count', 'role', 'cell_count_total'], axis=1)
df_gr.to_csv('OUTPUT/GRvalues_Example1.tsv', sep='	', index=False)
df_gr.head()
```

Columns to average: "corpse_count" "cell_count_total" "cell_count" "cell_count_dead" "cell_count_ctrl" "GRvalue" "cell_count_time0"

Columns added as annotations: "date"

---> Following columns are discarded:
"treatment_file" "well" "barcode"
(set as key if necessary)

<table>
<thead>
<tr>
<th>cell_line</th>
<th>treatment_duration</th>
<th>concentration</th>
<th>agent</th>
<th>date</th>
<th>cell_count</th>
<th>cell_count_ctrl</th>
<th>GRvalue</th>
<th>cell_count_t</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL_1</td>
<td>72.0</td>
<td>0.001</td>
<td>D_1</td>
<td>2016-06-06</td>
<td>3583.4444</td>
<td>3627.85</td>
<td>0.991393</td>
<td>491.525</td>
</tr>
<tr>
<td>CL_1</td>
<td>72.0</td>
<td>0.001</td>
<td>D_2</td>
<td>2016-06-06</td>
<td>3612.0000</td>
<td>3627.85</td>
<td>0.996945</td>
<td>491.525</td>
</tr>
</tbody>
</table>
Fit a dose-response curve to obtain sensitivity metrics

Evaluate the GR metrics and plot the dose-response curves (protocol 3, step 2)

```python
df_grmetrics = gr50.gr_metrics(df_gr)
df_grmetrics.to_csv('OUTPUT/GRmetrics_Example1.tsv', sep='\t', index=False)
print(df_grmetrics.head())
fig = gr50.plot.plot_curves(df_grmetrics, df_gr, colorvar='cell_line', colvar='agent')
fig.savefig('OUTPUT/GRcurves_Example1.pdf')
```

<table>
<thead>
<tr>
<th>date</th>
<th>cell_line</th>
<th>agent</th>
<th>treatment_duration</th>
<th>GR50</th>
<th>GRmax</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-06-06 12:34:56</td>
<td>CL_1</td>
<td>D_1</td>
<td>72.0</td>
<td>0.126677 -0.173488</td>
<td></td>
</tr>
<tr>
<td>2016-06-06 12:34:56</td>
<td>CL_1</td>
<td>D_2</td>
<td>72.0</td>
<td>0.311289 -0.004219</td>
<td></td>
</tr>
<tr>
<td>2016-06-06 12:34:56</td>
<td>CL_1</td>
<td>D_3</td>
<td>72.0</td>
<td>0.690251 0.401416</td>
<td></td>
</tr>
<tr>
<td>2016-06-06 12:34:56</td>
<td>CL_1</td>
<td>D_4</td>
<td>72.0</td>
<td>3.703113 -0.096859</td>
<td></td>
</tr>
<tr>
<td>2016-06-06 12:34:56</td>
<td>CL_1</td>
<td>D_5</td>
<td>72.0</td>
<td>0.668006 -0.094478</td>
<td></td>
</tr>
</tbody>
</table>

Dose-response plots
GRcalculator.org can replace the last part of the protocol

GRcalculator.org

Clark*, Hafner* et al., BMC Cancer, in review

Hafner*, Heiser* et al., Sci Data, in review
Advantages of an automated pipeline

1. Complex plate layouts can be designed
2. A single data container for data and metadata
3. Extensions and modifications can be recorded
4. Integration with analysis tools
   - Jupyter notebooks enable ease of documentation and executions