Title

RMeDPower for Biology: guiding design, experimental structure and analyses of repeated measures data for biological studies

Authors

Min-Gyoung Shin[1], Julia A. Kaye[2*], Naufa Amirani[2], Stephanie Lam[2], Reuben Thomas[1*], Steven Finkbeiner [2,3,4,5*] *Corresponding authors

Affiliations

1. Gladstone Institute of Data Science and Biotechnology, Gladstone Institutes, San Francisco, CA, USA
2. Center for Systems and Therapeutics, Gladstone Institutes, San Francisco, CA 94158, USA
3. Taube/Koret Center for Neurodegenerative Disease Research, Gladstone Institutes, San Francisco, CA 94158, USA
4. Gladstone Institutes, San Francisco, CA 94158, USA
5. Department of Neurology, University of California, San Francisco, CA 94158, USA; Department of Physiology, University of California, San Francisco, CA 94158, USA; Neuroscience Graduate Program and Biomedical Sciences Program, University of California, San Francisco, CA 94158, USA

Contributions

Conceived the idea SF
Designed the overall work and oversaw all development MS, JAK, RT
Wrote the manuscript with input and edits from all the authors MS, JAK, RT
Provided project leadership JAK, RT, MS
Performed integrative analysis and computational modeling MS, NA, SL
Developed the statistical models MS, RT
Ran and optimized the R code MS, NA, SL, RT
Optimized the statistical models MS, NA, SL, RT, JAK

Acknowledgments and Funding

We would like to acknowledge these sources of funding: W81XWH-20-1-0710, NSF 1761941, NIH R37NS101996, P01 AG54407, R01 LM013617, W81XWH-18-1-0696, ALS FindingACure and Answer ALS and the Robert Packard Center for ALS Research. We also would like to thank all members of the Finkbeiner lab, Alex Pico, Kathryn Claiborn, Kelley Nelson and Gayane Abramova.

Abstract

Reproducibility is science has plagued efforts to understand biology at both basic and biomedical and preclinical research levels. Poor experimental design and execution can result in datasets that are improperly powered to produce rigorous and reproducible results. In order to help biologists better model their data, here we present a statistical package called RMeDPower in R, which is a complete package of statistical tools that allow a scientist to understand the effect size and variance contribution of a set of variables one has within a dataset to a given response. RMeDPower can estimate the effect size of variables within an experiment based on an initial pilot dataset. In this way, RMeDPower can inform the user how to predict the scope, dimension and size of biological data needed for a particular experimental design. RMeDPower employs a generalized linear mixed model (LMM) -based power analysis, specifically targeting cell culture-based biological experimental designs. This package simulates experiments based on user-provided experimental design related variables, such as experiments, plates, and cell lines as random effects variables. This package not only allows us to use pilot data to estimate variance components for power simulation, it also accepts a set of variance components, which is an estimation of variance of the random effects linked to experimental variables and transformed into Intra-class Correlation Coefficients (ICC), as input which is precalculated from different data sets. The latter case is suitable when pilot data has an insufficient number of replications of experimental variables to directly estimate associated variance components. RMeDPower is a powerful package that any scientist or cell biologist can use to determine if a dataset is adequately powered for each experiment and then model accordingly.

Introduction

Improperly powered experiments lead to inconclusive results and can produce misleading findings that are either falsely positive or negative. The resulting lack of rigor and reproducibility has caused a crisis in science, highlighted in the popular literature, which has become a focus for the National Institutes of Health. It has been estimated that the majority of published empirical observations cannot be reproduced[1-6] rendering forward movement in scientific endeavors to not only understand basic biological mechanisms but also to design effective therapeutic approaches to combat disease futile. Further, the resources and time spent to attempt to reproduce findings from low quality or incorrectly acquired data is estimated to cost the global scientific communities and institutions about 200 billion dollars per year[7]. Therefore, tools are desperately needed to help researchers guide their experimental design as well as apply adequate statistical power estimation. Not only can this improve our confidence in scientific outcomes, but can help make biological experiments more time-efficient and cost-effective. For example, if a researcher can estimate how many experiments should be performed, how many cell lines should be chosen, and how many cells per cell line should be collected to achieve sufficient statistical power to test their hypothesis, they can invest the correct amount of time and resources to conclusively test the motivating hypothesis. Further, the dangers of pseudo-replication inherent in cell-culture based experimental designs[8-10] have been pointed out.

Statistical tools to perform power analyses in the context of cell culture experiments have been lacking. The packages in R that provide power estimates either do not cover the repeated measures nature of the biological experiments[11] or do so in the general situation[12, 13] where it becomes difficult for a non-statistician to immediately use. These packages —namely Pwr, Pamm and Simr[11, 14, 15]—are useful for many applications including ecological psychology and language acquisition, but they are not specialized tools for molecular biology experiments and can be challenging to apply to cell culture. The Pwr package allows power estimation and sample size calculation for given input effect sizes based on user-selected statistical tests such as proportion test, t-test, and ANOVA but does not provide estimations involving random effects. The pamm package is specifically built for power estimation of random effects in regression models [2] given input variance components for these effects. A more advanced tool is the simr package, which is built for generalized linear mixed models[15]. Users can calculate power or sample size for a given regression model that includes single, multiple fixed or random effects in a general setting. Simr simulates response variables using user-provided input values or pilot data to estimate the statistical power to capture given effects.

Investigators planning clinical trials or scientists performing animal experiments have readily available tools for assessing the quality of their experimental design and ensuring that the cohorts they assemble will be properly powered to reach strong conclusions. However, these tools are largely lacking for cell biologists. Of course, many simple statistical packages exist that use simple graphical user interface, such as such as GraphPad Prism, but they do not provide functionality for statistical power estimation[16] while others statistical software like SPSS[17], STATA[18] and SAS[19] do provide this functionality though they require licenses.

Therefore, we set out to develop a set of open-source rigorous tools that will allow scientists to understand the statistical power in the context of a cell-culture based repeated measures design they are studying and consequently the degree of confidence they can have in any given set of observations. RMeDPower would thus represent an invaluable addition to the field.

We designed RMeDPower to allow researchers to calculate power using detailed cell culture experiment-based parameter input options. RMeDPower consists of two separate tools: 1) CalcPower and 2) FinalCalc.

CalPower employs a generalized linear mixed model (LMM)-based power analysis, and simulates data based on user-provided experimental design-related variables, such as experimental batches, plates, cell lines, and cell counts, modeled as random effects. Condition variables define and specify cell status such as treatment or control are considered fixed effect variables, and response variables correspond to traits measured from the experiment. This package not only allows users to use pilot data to estimate variance components for simulation, it also accepts a set of variance components, which is an estimation of variance of the random effects linked to experimental variables and transformed into Intra-class Correlation Coefficients (ICC)[20], as input. The latter case is suitable when pilot data has an insufficient number of replications of experimental variables to directly estimate associated variance components, for example when pilot data are derived only from one experimental batch of cells that were assayed on one plate.

The second part of RMeDPower, FinalCalc, is a sophisticated set of tools that allows a user to examine the normality assumptions for the LMM[21], log-transforms the response values if

necessary, and uses the Rosner's test[22-27] to remove outliers. It includes visualization of distribution of raw and log-transformed response values, allowing users to check whether the response values conform to the normality assumption. In addition, the tool box includes a set of scripts to use the LMM to account for all of the variability observed within a dataset and estimate the desired parameter of interest. The scripts also allow users to specify which experimental variables conform to a nested design. Users can also specify a desired effect size that the model can rely on instead of the estimated effect size from the observed data. These additional steps allow a user to model their data in the most rigorous fashion.

For example, induced pluripotent stem cells (iPSCs) have transformed the ability of researchers to model human disease in a dish[28-30], but the system poses numerous challenges. Genetic variability results in variation across cell lines[31-33], and a typical neuronal differentiation can take up to 1.5 months and use dozens of morphogens and reagents, and hence is subject to fluctuations due to time, batch and cell culturist. These variables make it challenging to determine how much data are required in order to reliably interpret a set of results[34, 35].

We validated RMeDPower on a set of experiments using images from iPSCs that had been differentiated towards a neuronal lineage. The pilot dataset consisted of measurements of cells that were captured using Robotic Microscopy and[36-44] over[36-44] and 5 simulated measurements based on the original data with different effect sizes (**Supp Data Table 1**). The raw images are run through our custom-built imaging pipeline we assembled in Galaxy software[45] in order to obtain object crops that contain a single cell from raw image tiles. These crops containing single cells are then contrast enhanced with 1.5% saturation, normalized, denoised, and pixels that deviate from their neighborhood median by threshold are removed using FIJI[46]. A smoothing algorithm is applied to remove extraneous debris/cells around the central cell. The crops were then subjected to our morphological feature-based pipeline that processes single images from disease or control neurons. Here we are examining the perimeter of the cell based on contour ellipse[47].

This simplified dataset consists of a minimal category of experimental variables used to run power simulation analyses with known ICC values. The results demonstrate the capacity of this package to be used on any biological or cell culture-based set of data generated under a repeated measures design.

Methods

Implementation

The CalPower portion of RMeDPower broadly builds upon the power simulation structure of the simr package[15] to enable power calculations based on a generalized LMM with biological variables relevant to cell culture experiments. LMMs[21] are used to estimate associations of interest in situations where the responses are clustered or correlated by design. In cell culture experiments, responses are typically clustered by batch, experiment, plate or wells. In these LMMs the clustering variable is typically modeled as a random effect whose influence on the mean response is assumed to be drawn from a normal probability distribution with zero mean and given variance. The variables that specify conditions that distinguish groups (condition variable), for example disease vs control or drug vs non-treated, are assigned as fixed effects in a LMM, fit using the lmerTest

function from R library lme4[48]. As in simr, CalPower simulates response variables, which correspond to direct measurements, based on given variance components of the random effects estimated from a user-provided pilot dataset. The package also requires distinguishing between repeatable and non-repeatable variables to account for experimental settings that include measurements that occur more than once but that should be considered as part of the nested experimental design (e.g., plates within a given experimental batch or wells within a plate). Experimental variables such as plates are considered non-repeatable variables that are unlikely to exist in multiple experiments with the same ID. The power estimation is based on the assumption that the relationship between the condition variable and the response variable, as captured by the effect size, is a true association. The simulated response variables are refitted into a LMM and the total number of significant association results are counted to estimate power to detect the association at a chosen Type I error level. In cases where the pilot dataset does not have enough data for each experimental variable, users can provide ICC values (Formula 1), which can be used to estimate variance components. These will come from prior knowledge based on empirical data. Here, a high ICC value indicates that the variable has a relatively high variance relative to the overall variance estimate of the response variable. Power simulation can be performed under user-specified rules such as expected power, sample size or number of independent experimental variables (level) of the hypothetical experiment. Users can also choose a Type I error level, the desired number of simulations, the number of maximum levels or samples to be tested, and which output format or file name to use.

Operation

Users can find RMeDPower on github page
https://github.com/gladstone-institutes/RMeDPower/

The library can be installed directly from github using commands:

```
library(devtools)
install_github('gladstone-institutes/RMeDPower', build_vignettes=TRUE)
```

To reproduce the results of examples in this paper, based on RMeDPower release 1.0:
- R >= 4.0.4
- simr >= 1.0.5
- lme4 >= 1.1-27.1

Use cases

In a typical scenario, a user would first run the CalPower portion of RMeDPower to estimate the power of a pilot dataset. Once the final data acquisition has been completed, then the user will model the data using the FinalCalc portion of RMeDPower (Figure 1). In the data structure we refer to, each experiment is considered a separate batch of relevant observations. These observations can be any set of values that have been measured with the user-defined experimental framework. To improve comprehension about the practical use cases of the package, we included two subsets of real biological experimental data—the physical size of a set of cells from different cell lines—in the package. The first dataset, RMeDPower_data1, is composed of three pilot

experimental batches (identified using the column labeled "experiment"), 10 cell lines (identified using the column labeled "line"), condition variable (identified using the column labeled "classification") and 6 response variables that are cell-based measures that capture information about cell morphology. The first response variable, 'cell_size1', corresponds to the measured perimeter value of cells and displays an effect size of 1.1, and the second to sixth response variables correspond to simulated response variables with effect sizes of 1.56, 2.02, 3.16, 4.08, 5.2 (Table 1, **Supp Data Table 1).** The second dataset contains one pilot experiment that consists of one plate, two cell lines, and one response variable (**Supp Data Table 2).** Since data in **Supp Data Table 2** cannot be used for estimating variance components, users are required to provide ICC values which can be calculated using Formula 1. Based on given ICC values, the package will calculate variable components and simulate the response variable to assess power.

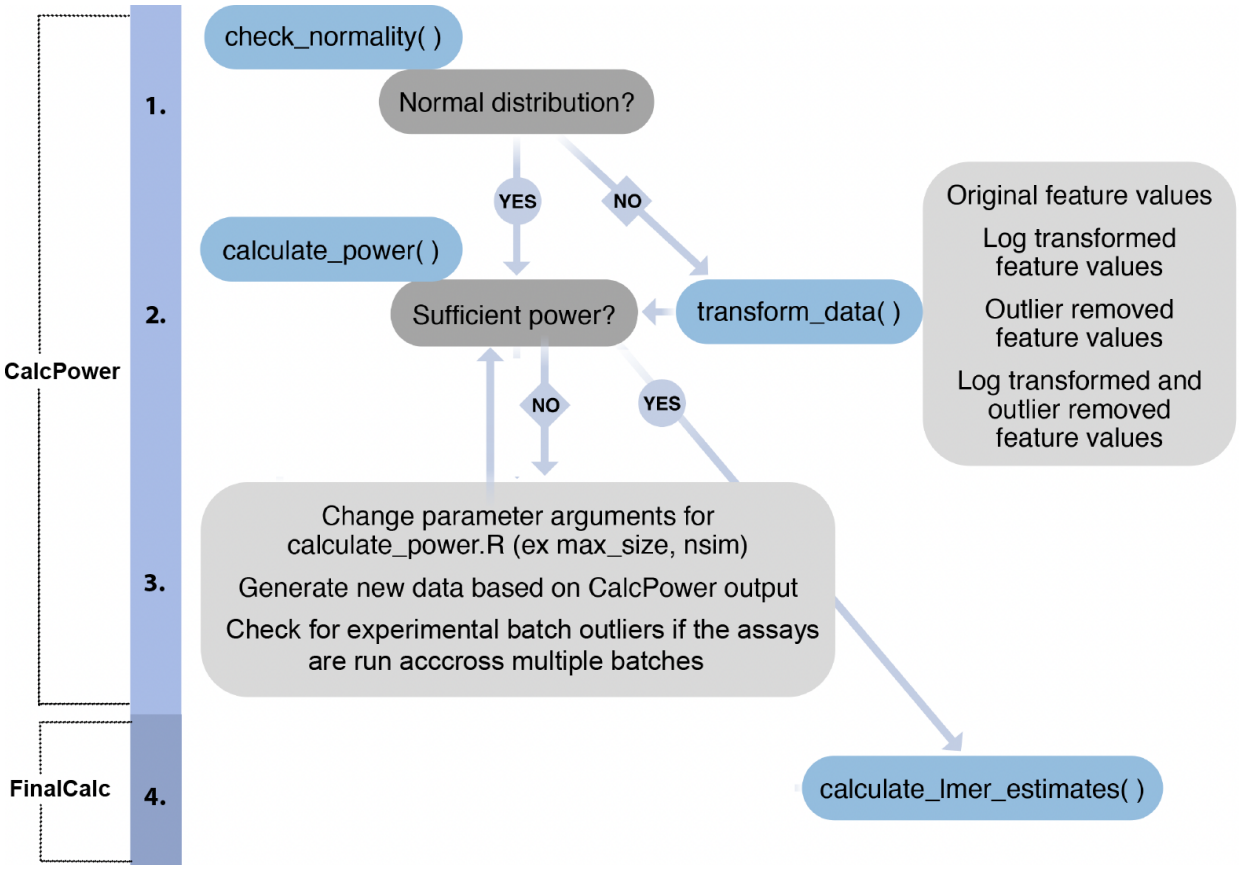

Figure 1. Flowchart of data analysis using RMeDPower. 1. First, the 'check_normality' function is run to explore if the data pass a normality distribution. 2. In the case that the data do not pass the normality check, the 'transform_data' function can be used to log-transform the response values and/or remove outliers. 3. The function 'calculate_power' can be used to estimate the statistical power given the data in its current form. 4. If the final dataset has been acquired and is

now is ready for statistical modeling, the 'calculate_lmer_estimates' function can be used to perform a LMM regression to estimate the parameter of interest.

Experimental variable simulation scheme

RMeDPower is designed to simulate the effect of variability of the responses which could come from differences in processing batch, plates or cell lines on the responses of interest. There are several ways to assess the variability of experimental variables in this package as outlined below.

First, a user can assess how increasing the number of independent experiments affects power. For example, if a user has a pilot dataset that consists of 3 experimental batches that each contain 2 plates, expanding this dataset two-fold would have the effect of simulating 6 experiments with a total of 12 plates (**Figure 2)**. Alternatively, a user can assess how increasing the number of plates per experiment affects power. In the case where there are two experiments that each contain two plates, the user can double the number of plates used per experiment. In this way, the user can simulate how the statistical power changes as a result of increasing the number of plates used per experiment rather than increasing the number of experimental batches (**Figure 3)**. In a third example, a user can examine the effect of expanding the number of cell lines within each experiment affects power (**Figure 4**). This would capture the effect of increasing the number of cells assayed as a consequence of having more cell lines. This type of variable expansion can be accomplished by setting 'level=1' in the calculate_power function.

A user may want to examine the power of increasing the total number of cells measured from each experimental variable per experiment. For example, if there are 12 cells per cell line on plate 1, doubling the number of cells from each plate will result in assessing the effect in twice the number of cells per cell line, even if the number of experiments and cell lines remain the same (**Figure 5).** Alternatively, one might want to assess the effect of increasing the total number of cells by increasing the number of cells per cell line (**Figure 6**). This type of variable expansion can be accomplished by setting 'level=0' in the calculate_power function.

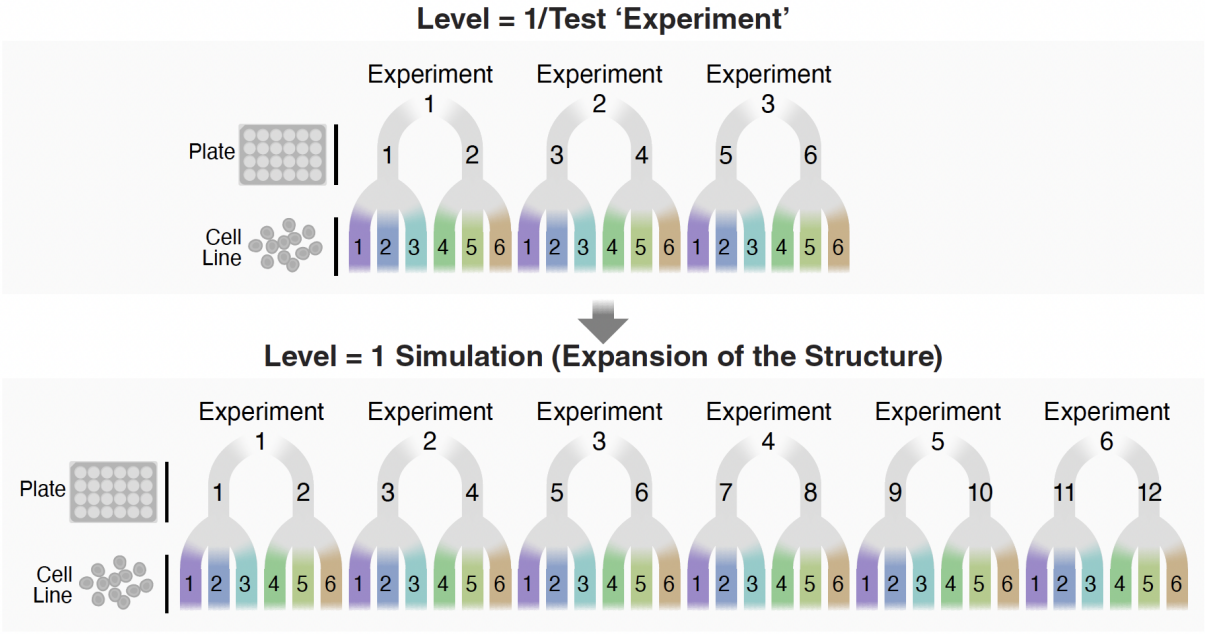

Figure 2. Example of RMeDPower's simulation of experimental variability for 'experiment' for the power analysis. If level=1 is set, RMeDPower simulates the effect of adding in additional experiments by inheriting the experimental design structure from the existing data. This describes the situation where the pilot study involves data from 3 experiments, with 2 plates used per experiment, 3 cell lines within each plate and 12 cells per cell line are assayed.

(A)

	Cell line1	Cell line2
Exp1	3	6
Exp2	3	0

(B)

	Cell line1	Cell line2
Exp1	3	6

Exp2	3	0
Exp3	3	6
Exp4	3	0

Table 1. Example of changes in cell counts in a simulation of experimental batch variability. (A) Original data structure and (B) changed data structure after simulation. Simulation based on level=1 keeps the same structure and increases the number of experiments. This describes the situation where the pilot study involved 2 experiments, in the first experiment two cell lines are used with 3 and 6 cells assayed, respectively, and in the second experiment only 3 cells from the first experiment are assayed.

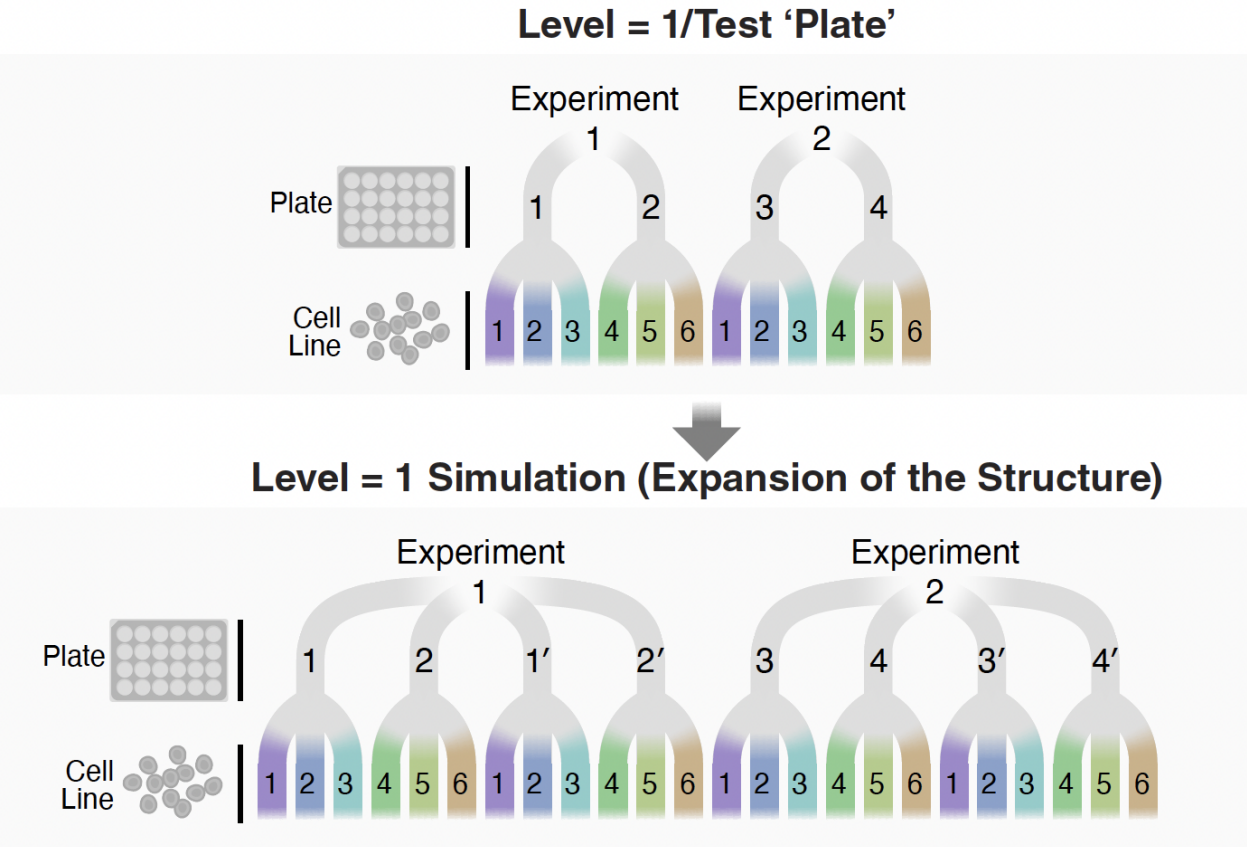

Figure 3 Example of RMeDPower's simulation of plate variability for 'plate' power analysis. If level=1 is set, RMeDPower simulates new plates by inheriting the experimental design structure from the existing data.

	Cell line1	Cell line2
Plate1	3	6
Plate2	3	0

(B)

	Cell line1	Cell line2
Plate1	3	6
Plate2	3	0
Plate1'	3	6
Plate2'	3	0

Table 2 Example of changes in cell counts in a simulation of plate variability. (A) Original data structure and (B) changed data structure after simulation. Simulation based on level=1 keeps the same structure and increases the number of plates.

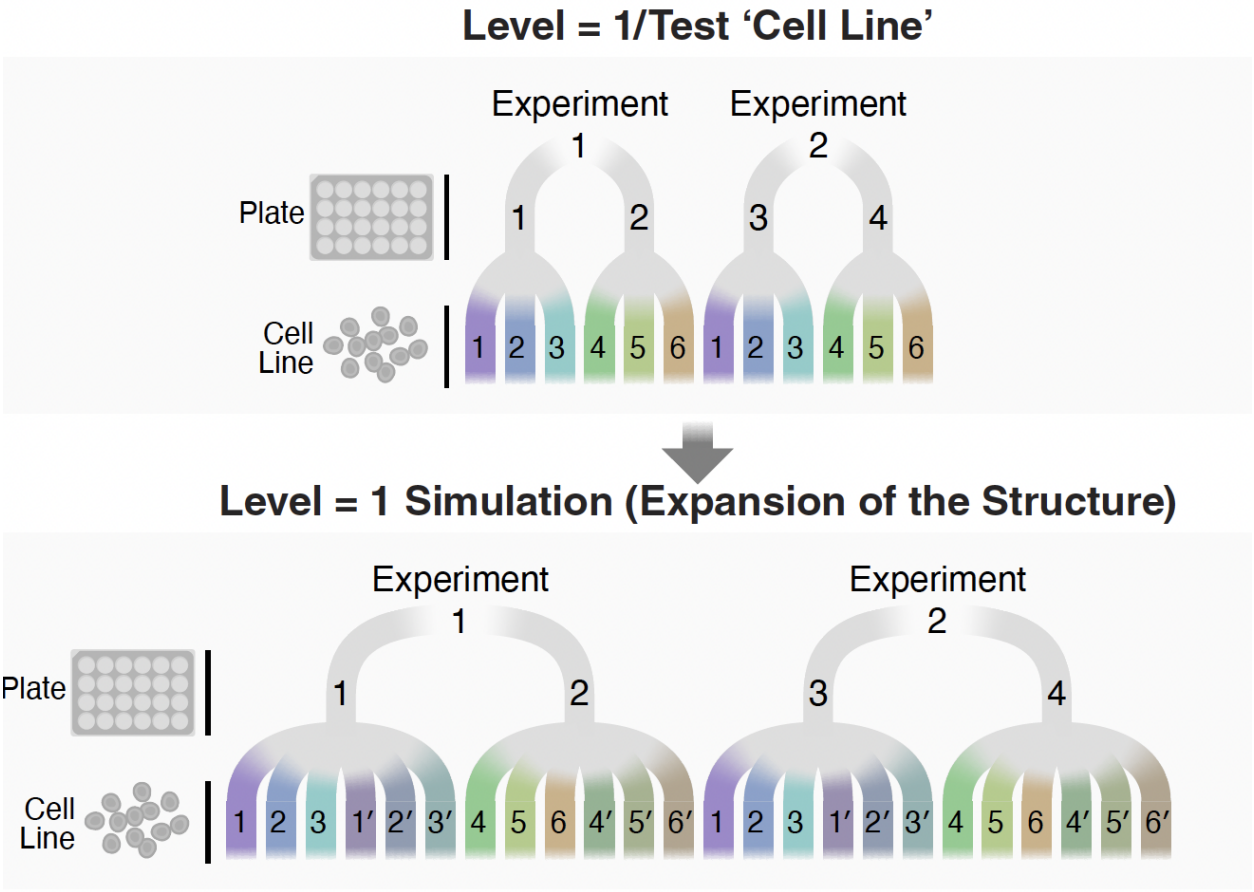

Figure 4. Example of RMeDPower's simulation of cell line variability for 'cell line' power analysis. If level=1 is set, RMeDPower simulates new cell lines by inheriting the experimental design structure from the existing data.

	Cell line1	Cell line2
Exp1	3	6
Exp2	3	0

	Cell line1a	Cell line2a	Cell line1b	Cell line2b	Cell line 1c	Cell line 2c

Exp1	3	6	3	6	3	6
Exp2	3	0	3	0	3	0

Table 3. Example of changes in cell counts in a simulation of cell line variability. (A) Original data structure and (B) changed data structure after simulation. If level=1 is set, RMeDPower simulates new cell lines by inheriting the experimental design structure from the existing data. An example of a simulation in tabular format shows the change in the number of cells per cell line and experiment after simulation. Simulation based on level=1 keeps the same structure and increases the number of cell lines.

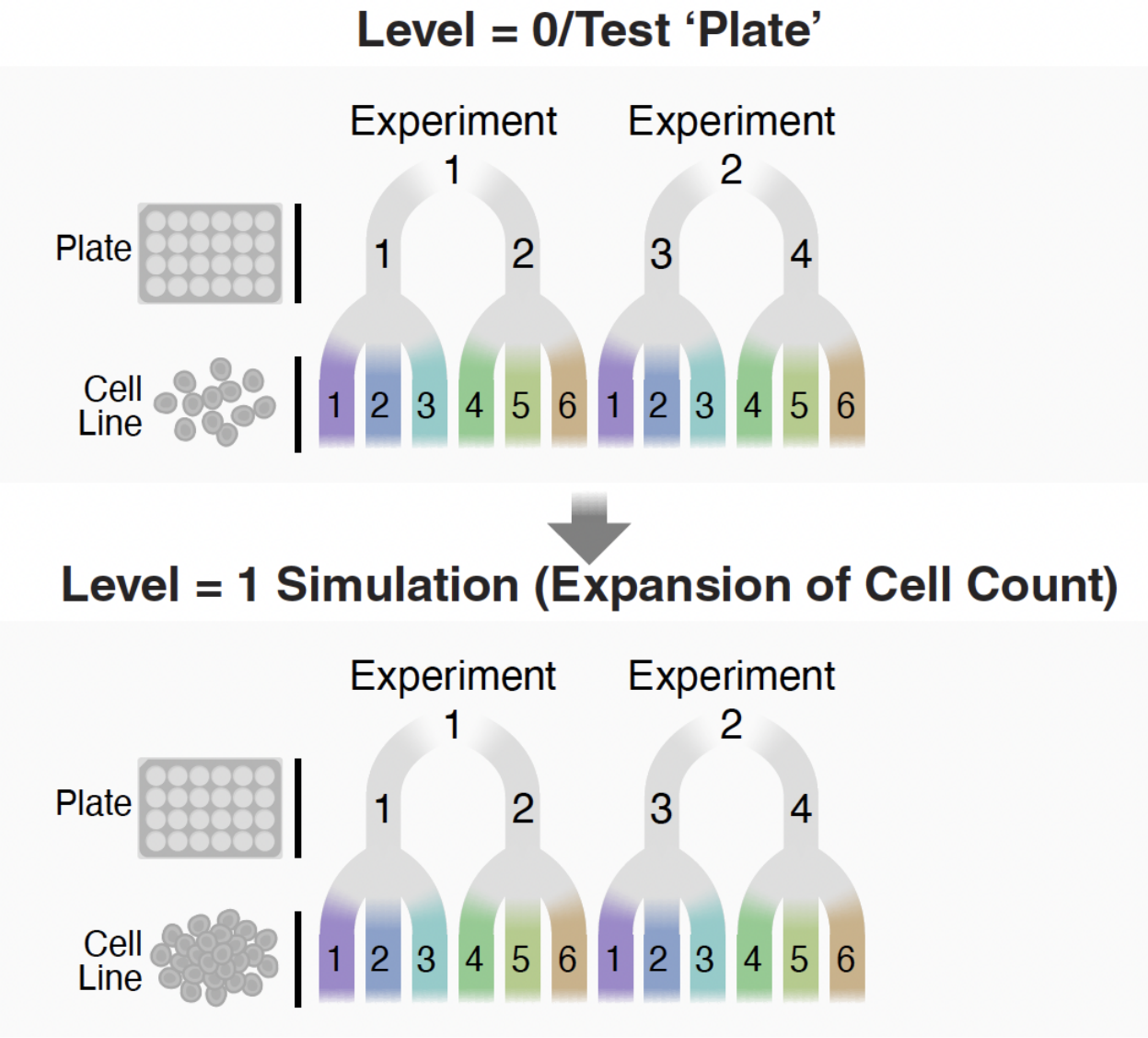

Figure 5. Example of RMeDPower's simulation of a sample size increase for 'plate' power analysis. If level=0 is set, RMeDPower multiplies the number of cells per plate by M/N, where N is the maximum number of cells per plate and M is a value assigned to the parameter 'max_size'.

	Cell line1	Cell line2
Plate1	3	6
Plate2	3	0

(B)

	Cell line1	Cell line2
Plate1	6	12
Plate2	6	0

Table 4. Example of plate-based cell count expansion simulation. (A) Original data structure and (B) changed data structure after simulation. The simulation is based on level=0 which results in increased number of cells per plate

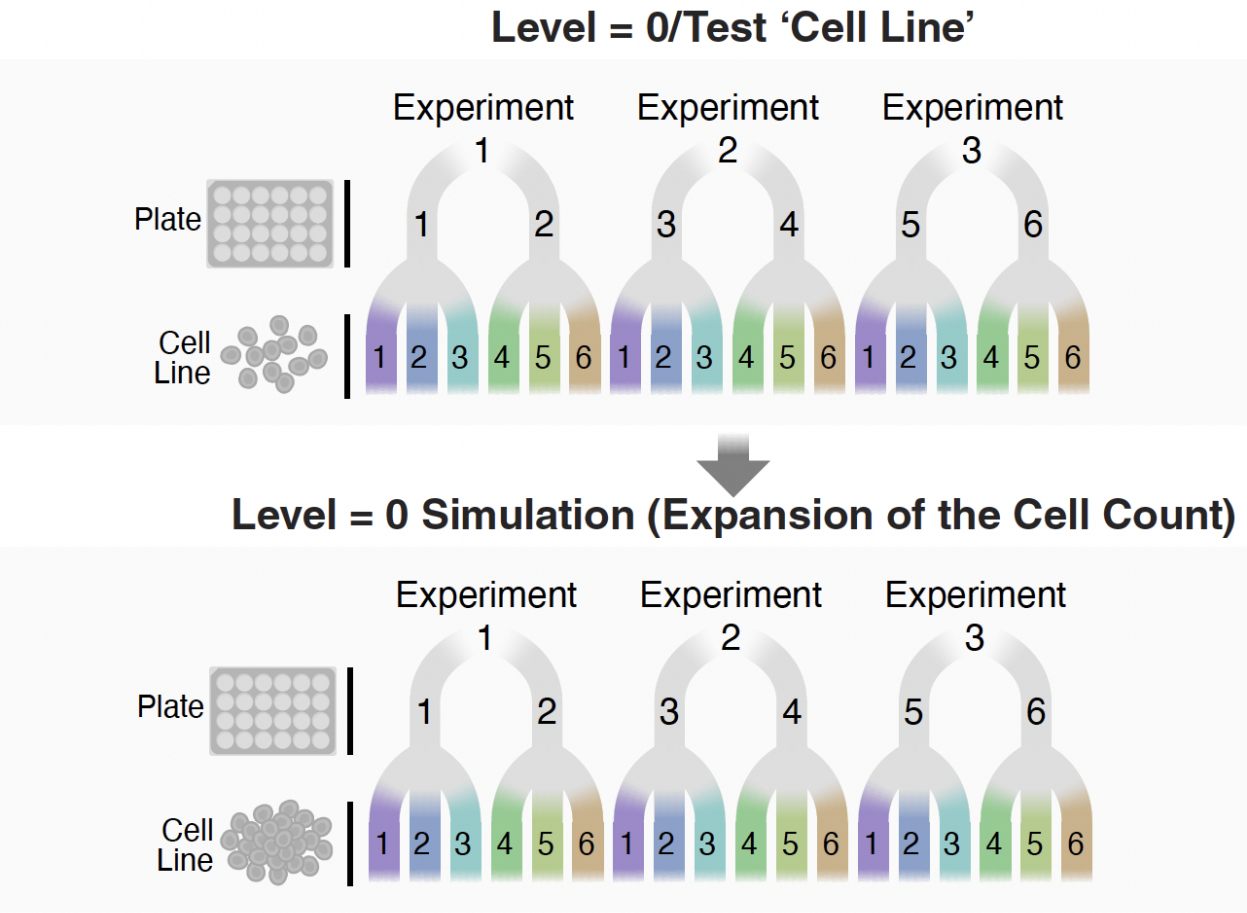

Figure 6. Example of RMeDPower's simulation of a sample size increase by increasing the number of cell line. If level=0 is set, RMeDPower multiplies the number of cells per cell line by M/N, where N is the maximum number of cells per cell line and M is a value assigned to the parameter 'max_size'.

(A)

	Cell line1	Cell line2
Exp1	3	6
Exp2	3	0

(B)

	Cell line1	Cell line2
Exp1	9	18
Exp2	9	0

Table 5. Example of cell line-based cell count expansion simulation. (A) Original data structure and (B) changed data structure after simulation. An example of a simulation in tabular format shows the change in the number of cells per cell line and experiment after simulation. The simulation is based on level=0 which results in an increased number of cells per cell line.

Increasing effect sizes requires fewer experiments

A user would intuitively expect that datasets with larger number of experimental batches are needed to estimate a smaller effect size. To demonstrate this, we performed power analysis with 6 different effect sizes (1.1, 1.56, 2.02, 3.16, 4.08, 5.2) to test the relationship between effect size and number of experimental batches needed to detect an association at a 0.05 Type I error level. The effect size of 1.1 is the original effect size of the response variable in the assayed data, while the 5 other effect sizes are simulated. The assayed data included 8 experimental batches **(Supp Data Table 3)**. For each effect size, experimental batch datasets were subsetted so each subset with 3 batches is more representative of the size of pilot datasets. The subsetting resulted in 56 sub datasets or pilot studies. Then the number of experimental batches required to achieve 80% power

was obtained from each pilot dataset. For cases where 80% power was never achieved in the range of (1, 15) experimental batches, the censored value 15 was assigned as the required number of experiments. The plot of the average number of batches required to achieve 80% power (Figure 7) along with its associated standard deviation (in the gray shaded area) shows that for this particular response in the given dataset, the required number of experiments was close to 15 when the effect size was around 1, and the number decreases as the effect size increase, eventually reaching a plateau of value 3 with high effect sizes. As a reference for this simulation result, the pwr.t.test function from the R library pwr[11] was used to test the number of samples per group required to reach 80% power at a 0.05 Type I error level for a two group comparison across the range of the observed effect sizes. The results of both analyses showed a similar relationship between the number of required experiments and the effect size.

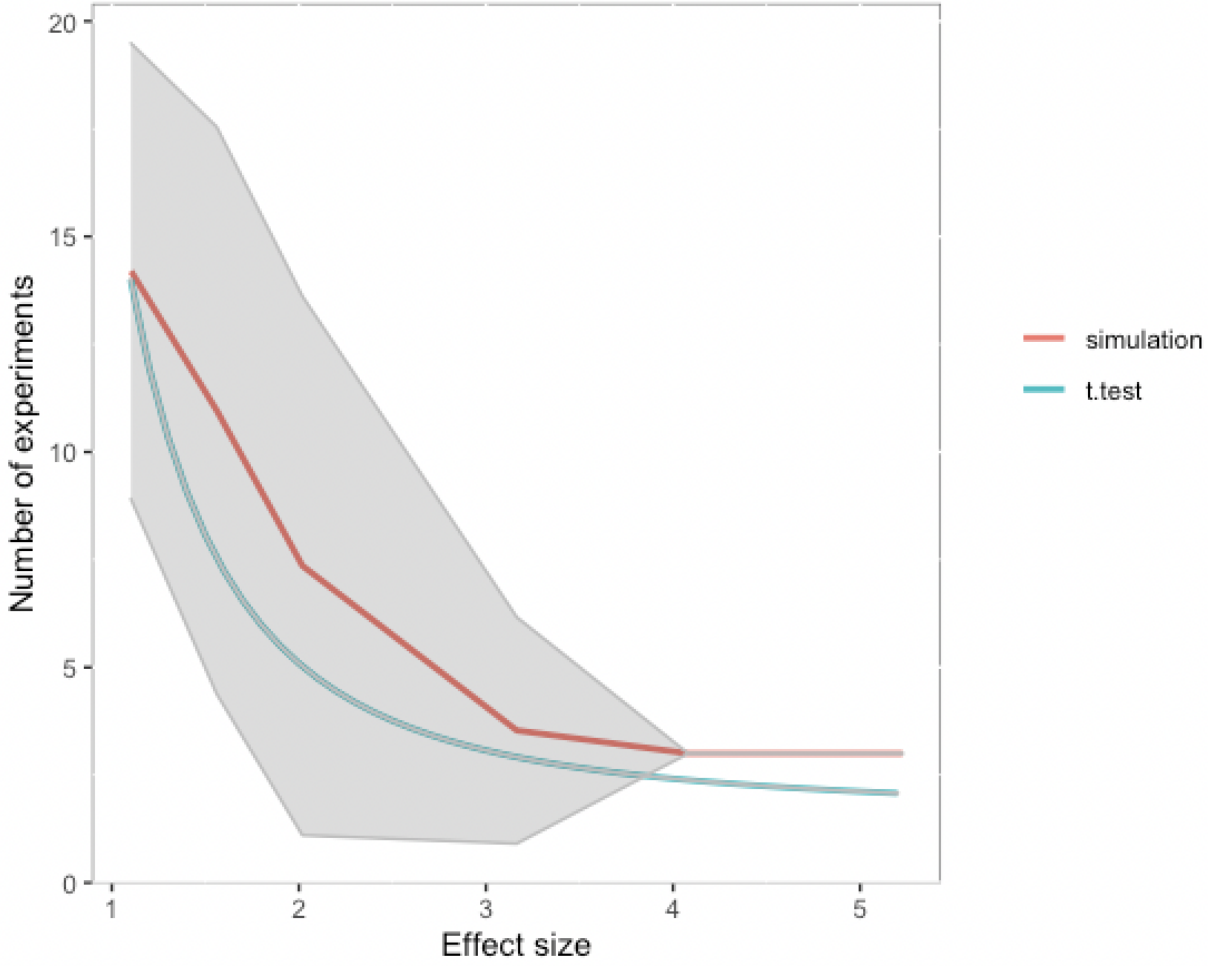

Figure 7. The number of experiments required to achieve 80% power at various effect sizes. Results are calculated from the mean number of experiments estimated by simulation (red) and t.test (blue), where gray intervals represent standard deviations calculated from simulations.

Preprocessing input dataset

In order to determine if biological/disease groups are different from one another, RMeDPower employs a LMM[21]. Cell-based measurements such as the response values we are using here often typically follow a normal distribution, but in some cases they do not. One of the assumptions of LMMs is that the conditional residuals from the model fit follow a normal distribution[21]. Users should examine this assumption using the 'check_normality' function to test the normality of each response variable. This function produces a quantile-quantile (QQ) plot that can be used to evaluate the level of normality of the given response variable. The user must provide the names of the cell condition, experimental variables, and response columns along with the dataset to be tested. Cell conditions can be continuous or discrete values. In the case where the data follow a normal distribution, we expect the QQ plot to be aligned on the line x=y line[49].

Code example:

```
check_normality(data=data, condition_column="classification", experimental_columns=c("experiment","line"), response_column="cell_size2", condition_is_categorical="TRUE")
```

Additional preprocessing is possible with the 'transform_data' function. This function not only generates a QQ plot from given response values, it also performs outlier analysis and log transformation of the response values. Outlier analysis is based on Rosner's test, which tests the number of outliers by assuming normality of the distribution[22-27]. The initial number of outliers is calculated based on a cutoff, median(value) ±3 MAD(value), where MAD is the median absolute deviation. 'transform_data' performs outlier analysis on raw response values and log-transformed response values, and produces QQ plots on four types of dataset: raw values, outlier-removed raw values, log-transformed values, and outlier-removed log-transformed values. Users can check QQ plots and use one of the values that can be retrieved in matrix form from the function output to proceed down the pipeline.

Code example:

```
Transformed_data = transform_data(data=data, condition_column="classification", experimental_columns=c("experiment","line"), response_column="cell_size2", condition_is_categorical="TRUE")
```

CalcPower

Ex1. Testing different levels of an experimental variable

For any given response variable, a user can investigate the power to detect association between response variables and condition variables by increasing the number of experimental batches as shown in **Figure 2**. As an example, we will examine the power by plotting a **p**ower **c**urve (PC)[15]. A PC plots the power to detect associations between the condition variable and the response variable against the number of replicates for any given data structure (**Figure 8)**. For example, for a pilot dataset that consists of 3 experiments and 10 cell lines, when we apply the PC we find that with the given variability and effect size for this particular response variable, it is predicted that after 2 independent experiments the data type will achieve 80% power to detect associations between the response variable/'cell_size2' and cell condition (**Figure 8**). Since we will be testing several experimental levels, we assign '1' to 'power_curve' to indicate that we want to create a power trend curve. 'condition_is_categorical' is TRUE, as it is binary, and 1000 simulations are chosen to be run. To specify experimental variables that may appear multiple times in different experimental settings, users must name these variables using the 'repeatable_column' parameter. For example, we will set 'repeatable_columns="line"' in this simulation as the same cell line may exist on different plates and experiments.

Code example:

```
calculate_power(data=data_renamed, condition_column="classification",
experimental_columns=c("experiment","line"), response_column="cell_size2",  repeatable_columns='line',
target_columns="experiment", power_curve=1, condition_is_categorical=TRUE, nsimn=1000, levels=1)
```

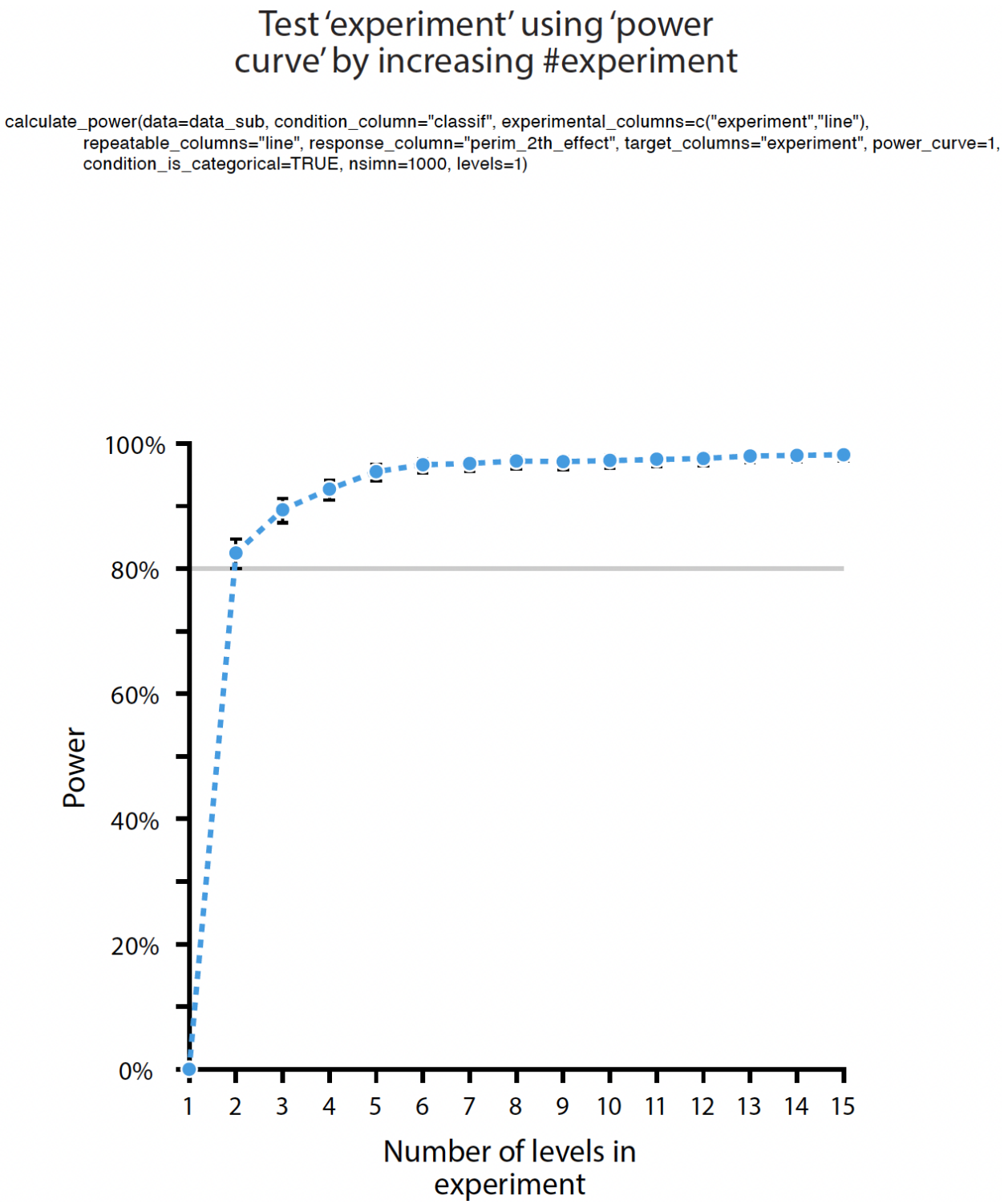

Figure 8. Example of a PC that is simulating the function of levels of experiment. Maximum experiment level was set to five times of the original level, and cell_size2 was used as a response variable. In this case, 2 experiments with this data type will achieve 80% power to detect significant associations across a set of lines for this given response variable.

Ex2. Testing different sample sizes of an experimental variable

A user can also estimate the power to detect the effect of increasing the cell number for each line as in **Figure 6.** For example, we will use an association between 'cell_size2' and the condition variable by increasing the number of cells in each cell line.

We will test up to 700 cells per cell line by setting max_size=700. The other parameter settings are the same except for 'target_columns,' where we input 'line'. For 'levels' we assign '0' because we are simulating multiple cells in the same cell line. The PC result reveals that the power to detect associations between the condition variable and the response variable at about 150 samples is about 80% (**Figure 9)**. In other words, we will need a minimum of 150 cells to achieve 80% power to detect an association between the condition variable and response variable.

Code example:

```
calculate_power(data=data_renamed, condition_column="classification",
experimental_columns=c("experiment","line"), response_column="cell_size2",  repeatable_columns='line',
target_columns="line", power_curve=1, condition_is_categorical=TRUE, nsimn=1000, levels=0, max_size=700)
```

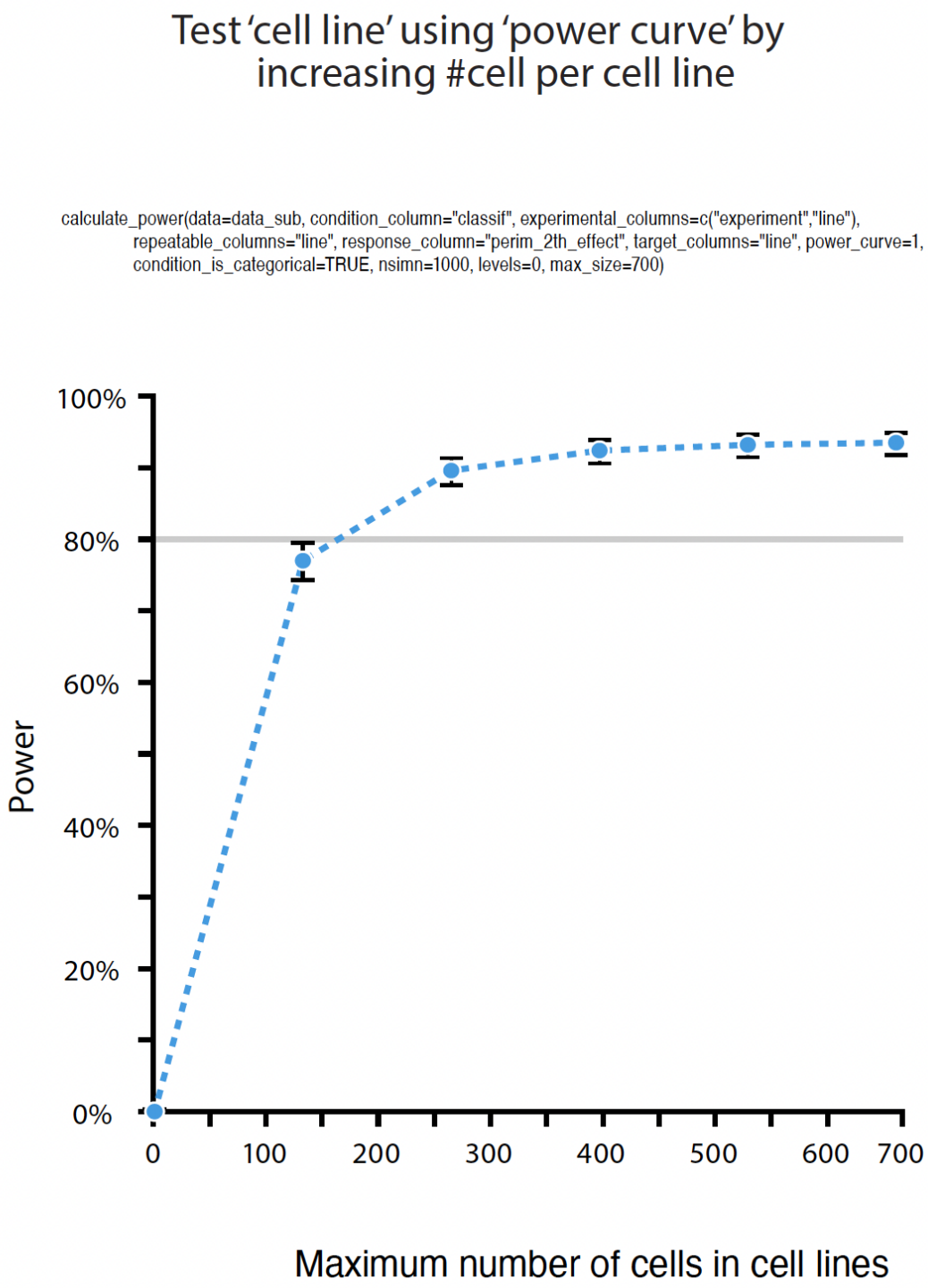

Figure 9. Example of a PC that estimates the power as a function of the number of cells per line. The maximum number of cells was set to 5 times the original number of cells, and cell_size2 was used as the response variable.

Ex3. User-determined level count and output file name

It may be the case that a user obtains a dataset that contains a specific number of experiments and has no opportunity to increase this number. In this case, one can avoid running unnecessary simulations. To do this, the user can assign a '0' to 'power_curve', and test the power of the experiments that contain the completed dataset. For example, if a user ran 15 experiments, then they would assign '15' to 'max_size'. The function will output the combined estimated power for these 15 experiments. We assign 'test.txt' to 'output' to output the result. The resulting power analysis will run 1000 simulations to detect associations between the condition variable and the response variable. For the example, dataset we provide that contains 3 experiments and this set of data achieves an estimated power of 97.60% (95% confidence interval (96.45, 98.46)) to detect associations between 'cell_size2' and cell condition at type I error rate of 0.05 (alpha). The cpu time for this analysis was 7 minutes and 30 seconds.

Code example:

```
calculate_power(data=data_renamed, condition_column="classification",
experimental_columns=c("experiment","line"), response_column="cell_size2",  repeatable_columns='line',
target_columns="experiment", power_curve=0, condition_is_categorical=TRUE, nsimn=1000, levels=1,
max_size=15, output="test.txt")
```

The text output of this example power analysis:

```
Power for predictor 'condition_column', (95% confidence interval):
     97.60% (96.45, 98.46)

Test: Likelihood ratio

Based on 1000 simulations, (0 warnings, 0 errors)
alpha = 0.05, nrow = 12940

Time elapsed: 0 h 7 m 30 s
```

Ex4. User-determined effect size

Consider the situation where a user has a specific effect size in mind based on prior information, but the pilot data itself does not reflect the *a priori* assumed effect size. For example, previously an experimenter found a significant difference comparing a given response between control cell lines "A", "B", "C", and cell lines from patients with a disease, "D", "E" and "F", and there was a sufficient sample size to estimate the effect size reliably. The experimenter then performs another

set of experiments comparing the same disease lines (“D”, “E” and “F”) but this time uses different control cell lines (“G” and “H”), and the data did not have a sufficient sample size in terms of the number of experimental batches. It seems reasonable to expect that the magnitude of the associations of the response variable between the new control lines and the original disease lines would be similar to the first set of experiments. In this case, the user can then run power simulations using the known effect size instead of estimating it from the pilot data. We will assume an effect size of 10 and assign the value to the 'effect_size' parameter. We will use the default max_size setting and the output file setting. The power analysis result shows that the power to detect associations between the condition variable and the response variable using 5 times the original number of experiments is 73.7% in 95% confidence interval (70.85, 76.41) at type I error rate of 0.05 (alpha). The cpu time for 1000 simulations was 7 minutes and 30 seconds.

Code example:

```
calculate_power(data=data_renamed, condition_column="classif", experimental_columns=c("experiment","line"),
repeatable_columns="line", response_column="cell_size2", target_columns="experiment", power_curve=0,
condition_is_categorical=TRUE, nsimn=1000, levels=1, effect_size = c(10))
```

Result:

Power for predictor 'condition_column', (95% confidence interval):

73.70% (70.85, 76.41)

Test: Likelihood ratio

Based on 1000 simulations, (0 warnings, 0 errors)

alpha = 0.05, nrow = 12940

Time elapsed: 0 h 7 m 30 s

Ex5. Test two experimental variables

A user may want to test the power of more than one experimental variable in a single run. For example, a user might be interested in knowing how increasing the total number of cells per experiment versus performing more experiments increases the power to detect associations between the response variable and the condition variable. Consider the example of testing power for two target parameters: experiment and cell line. We will test up to 9 experimental batches, and up to 142 cells per cell line. The function will return two power curves for each target parameter. In this way, the experimenter can pairwise compare different experimental paradigms to change the power to detect associations between the condition variable and the response variable. In the

example shown, we need either 2 experimental batches or about 200 cells per cell line to have at least 80% power (Figure 10).

Code example:

calculate_power(data=data_renamed, condition_column="classif", experimental_columns=c("experiment","line"), repeatable_columns="line", response_column="cell_size2", target_columns=c("experiment","line"), power_curve=1, condition_is_categorical=TRUE, nsimn=10, levels=c(1,0), max_size=c(9,142))

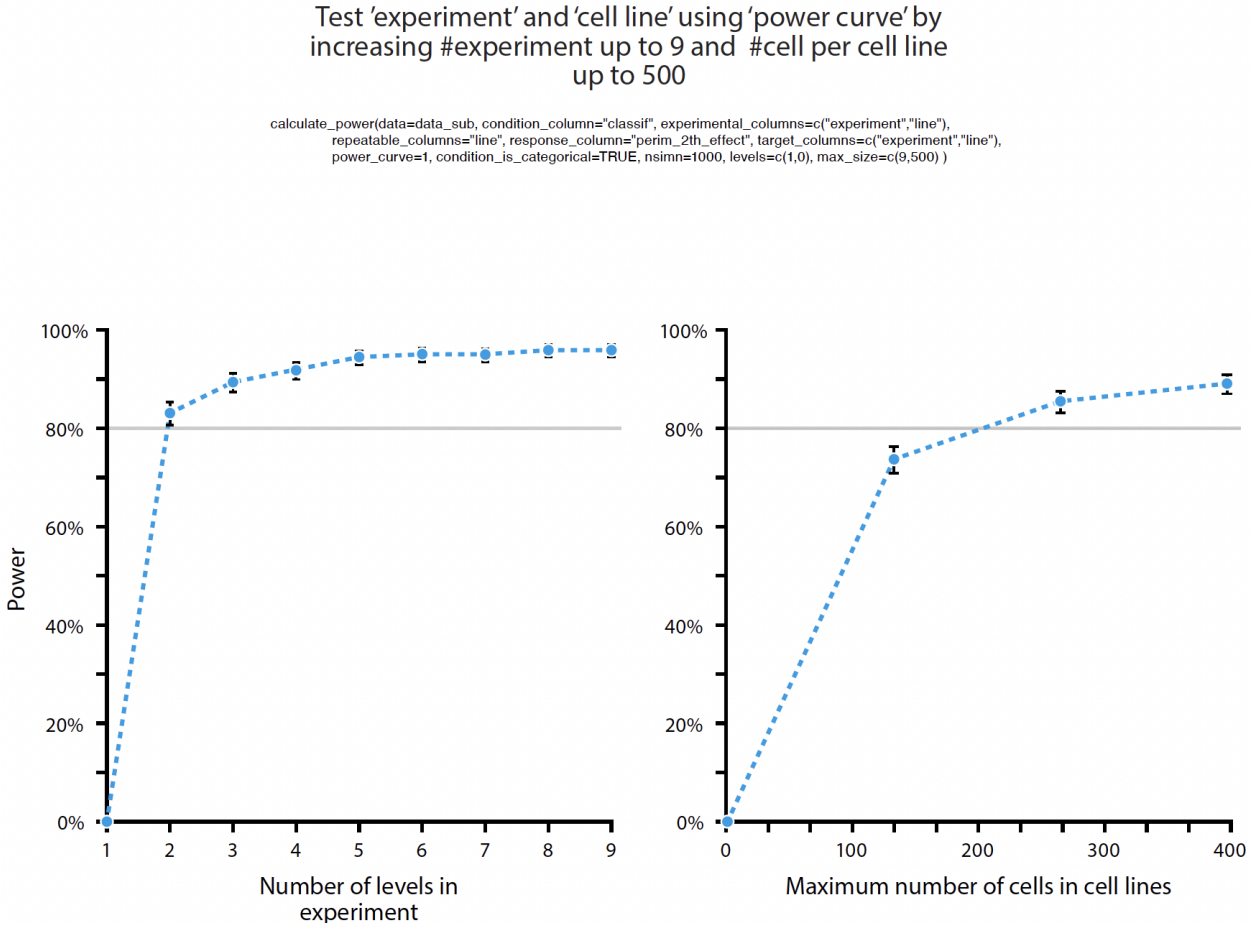

Figure 10. Example of power curves as a function of levels of experiment (A) and number of cell lines. Maximum experiment level was set to 9, and maximum cell line count was set to 142. cell_size2 2 was used as a response variable.

Ex6. Data with a single experimental category

Our power simulations depend on the variance components estimated from the input dataset. However, in some cases there may not be enough data to estimate the variance component. For example, the pilot data might have only a single category for 'experimental batch', ‘plate’ or 'line'. When this occurs, the user needs to provide ICC values. These values can be estimated from another dataset for which the variance components are assumed to be similar to those inherent for the new response being considered in the input dataset. ICC can be estimated by taking the ratio between the variance estimates using the following formula:

$$ICC_i = \frac{V_i}{\sum_{j \in S} V_i + e}$$

Formula 1.

where V_i represents the variance of the random effect linked to experimental variable (*e.g.*, experimental batch, plate or cell-line) i, and epsilon represents the variance of the residual error. S represents all the modeled sources of variability of the response under consideration. We will test this scenario using the example dataset with only single experiment and cell line (Figure 11).

Code example:

```
calculate_power(data=RMeDPower_data2, condition_column="classification",
experimental_columns=c("experiment","plate","line"), repeatable_columns=”line”,
response_column="response_variable", target_columns=c("experiment"), power_curve=1,
condition_is_categorical=TRUE, nsimn=1000, levels=c(1),  ICC=c(0.2,0.15,0.3))
```

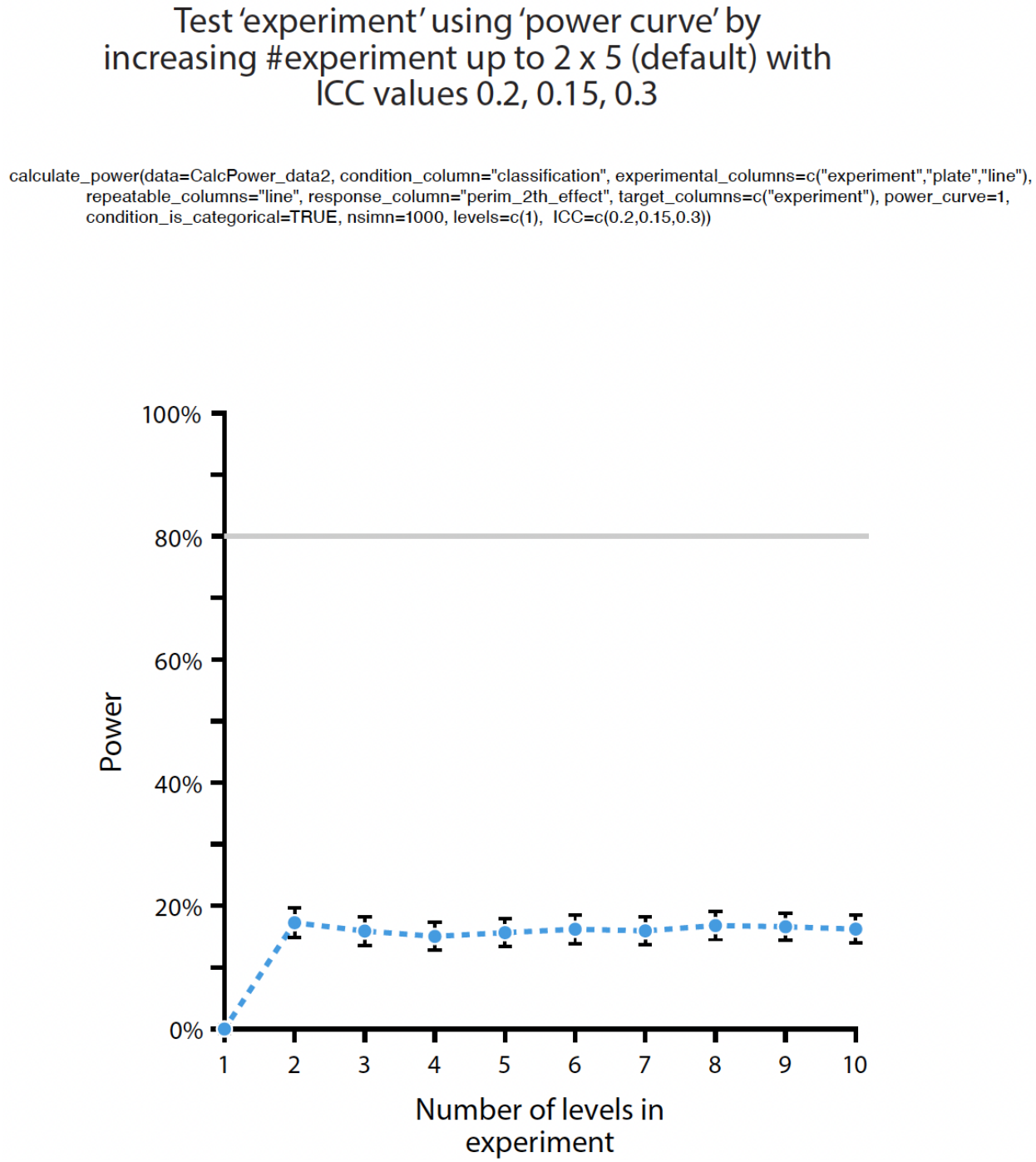

Figure 11. Power curve as a function of levels of experiment. Maximum experiment level was set to five times of the original level, and cell_size2 was used as a response variable. ICC values of each experimental variable were 0.2,0.15, and 0.3, respectively.

FinalCalc

RMeDPower also provides a function to perform LMM regression analysis if the user has data of sufficient size to estimate variance and the power to detect true associations[48, 50]. The parameters for 'calculate_lmer_estimates' are condition_column, Experiment_columns, response_column, condition_is_categorical and the input options are similar to what the 'calculate_power' function requires. The following example shows the result of a regression analysis performed on the original cell size data in 8 experiments.

Code example:

```
calculate_lmer_estimates(data=data, condition_column="classification",
experimental_columns=c("experiment","line"),
                    response_column="perim_2th_effect", target_columns=c("line"),
                   condition_is_categorical=TRUE )
```

Result:

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: response_column ~ condition_column + (1 | experimental_column1) +      (1 |
experimental_column2)
   Data: Data

REML criterion at convergence: 68551.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.5211 -0.5351 -0.1448  0.3264 31.5600

Random effects:
 Groups              Name        Variance Std.Dev.
 experimental_column2 (Intercept)   14.69   3.833
 experimental_column1 (Intercept)  230.95  15.197
 Residual                         3070.12  55.409
Number of obs: 6305, groups:  experimental_column2, 13; experimental_column1, 8

Fixed effects:
                  Estimate Std. Error      df t value Pr(>|t|)
(Intercept)        321.581      5.681   8.282   56.60 5.14e-12 ***
condition_column1   13.621      2.682  10.598    5.08 0.000399 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr)
cndtn_clmn1 -0.220
```

Summary

The RMeDPower R package is a power analysis tool designed to help biological scientists perform time-efficient and cost-effective studies by designing biological experiments with desirable power. The unique benefit of the package compared to other power analysis packages is that it is built to reflect real biological cell culture experimental designs. The package is tailored to manage multiple biological experimental variables to simulate experiments, and users can customize parameter settings to best fit the simulation settings to the experiment to be performed.

References Cited

1. Chan AW, Song F, Vickers A, Jefferson T, Dickersin K, Gøtzsche PC, Krumholz HM, Ghersi D, van der Worp HB. (2014) Increasing value and reducing waste: Addressing inaccessible research. *Lancet* 383:257–266. PMCID:PMC4533904
2. Young SS, Bang H, Oktay K. (2009) Cereal-induced gender selection? Most likely a multiple testing false positive. *Proc. Biol. Sci.* 276:1211–1222. PMCID:PMC2660953
3. Begley CG, Ioannidis JP. (2015) Reproducibility in science: Improving the standard for basic and preclinical research. *Circ. Res.* 116:116–126.
4. Dirnagl U, Duda GN, Grainger DW, Reinke P, Roubenoff R. (2022) Reproducibility, relevance and reliability as barriers to efficient and credible biomedical technology translation. *Adv. Drug Deliv. Rev.* 182:114118. PMCID:PMC Journal - In Process
5. Peers IS, Ceuppens PR, Harbron C. (2012) In search of preclinical robustness. *Nat. Rev. Drug. Discov.* 11:733–734.
6. Macleod MR, Michie S, Roberts I, Dirnagl U, Chalmers I, Ioannidis JP, Al-Shahi Salman R, Chan AW, Glasziou P. (2014) Biomedical research: Increasing value, reducing waste. *Lancet* 383:101-4.
7. Belluz J. (2015) Most research spending is wasted on bad studies. These billionaires want to change that. *Vox* Available from: https://www.vox.com/2015/10/4/9440931/arnold-foundation-meta-research.
8. Sullivan LM, Weinberg J, Keaney JF, Jr. (2016) Common statistical pitfalls in basic science research. *J. Am. Heart Assoc.* 5:e004142. PMCID:PMC5121512
9. Pollard DA, Pollard TD, Pollard KS. (2019) Empowering statistical methods for cellular and molecular biologists. *Mol. Biol. Cell* 30:1359–1368. PMCID:PMC6724699
10. Lazic SE, Clarke-Williams CJ, Munafò MR. (2018) What exactly is 'N' in cell culture and animal experiments? *PLoS Biol.* 16:e2005282. PMCID:PMC5902037
11. Champely S, Ekstrom C, Dalgaard P, Gill J, Weibelzahl S, Anandkumar A, Ford C, Volcic R, De Rosario H. (2020) pwr: Basic functions for power analysis. Available from: https://cran.r-project.org/web/packages/pwr/.
12. Martin J. (2020) pamm: Power analysis for random effects in mixed models. Available from: https://cran.r-project.org/web/packages/pamm/index.html.
13. Green P. (2022) simr: Power analysis for generalised linear mixed models by simulation. Available from: https://cran.r-project.org/web/packages/simr/index.html.
14. Martin J, Nussey DH, Wilson AJ, Réale D. (2011) Measuring individual differences in reaction norms in field and experimental studies: A power analysis of random regression models. *Methods Ecol. Evol.* 2:362–374.
15. Green P, MacLeod CJ. (2016) SIMR: An R package for power analysis of generalized linear mixed models by simulation. *Methods Ecol. Evol.* 7:493–498.
16. Dotmatics Gb. (2009) Can Prism perform sample size and power calculations? Available from: https://www.graphpad.com/support/faq/can-prism-perform-sample-size-and-power-calculations/
17. IBM. (2009) IBM SPSS software. Available from: https://www.ibm.com/analytics/spss-statistics-software
18. STATA. (2021) Stata 17 [software]. Available from: https://www.stata.com/
19. SAS. (2020) SAS/STAT version 15.2 [software]. Availabe from: https://www.sas.com/en_us/software/stat.html
20. Stanish WM, Taylor N. (1983) Estimation of the intraclass correlation coefficient for the analysis of covariance model. *Am. Stat.* 37:221–224.
21. Laird NM, Ware JH. (1982) Random-effects models for longitudinal data. *Biometrics* 38:963–974.
22. Barnett V, Lewis T. (1995) Outliers in Statistical Data. Third Edition. In: *Outliers in Statistical Data. Third Edition*, John Wiley & Sons, Chichester. 235–236.
23. Gilbert RO. (1987) Statistical Methods for Environmental Pollution Monitoring. In: *Statistical Methods for Environmental Pollution Monitoring*, Van Nostrand Reinhold, NY. 188–191.

24. McBean EA, Rovers FA. (1992) Estimation of the Probability of Exceedance of Contaminant Concentrations. In: *Estimation of the Probability of Exceedance of Contaminant Concentrations*, Ground Water Monitoring Review Winter. 115–119.
25. McNutt M. (2014) Raising the bar. *Science* 345:9.
26. Rosner B. (1975) On the detection of many outliers. *Technometrics* 17:221–227.
27. Rosner B. (1983) Percentage points for a generalized ESD many-outlier procedure. *Technometrics* 25:165–172.
28. Sayed N, Liu C, Wu JC. (2016) Translation of human-induced pluripotent stem cells: From clinical trial in a dish to precision medicine. *J. Am. Coll. Cardiol.* 67:2161–2176. PMCID:PMC5086255
29. Tiscornia G, Vivas EL, Izpisúa Belmonte JC. (2011) Diseases in a dish: Modeling human genetic disorders using induced pluripotent cells. *Nat. Med.* 17:1570–7156.
30. Xie YZ, Zhang RX. (2015) Neurodegenerative diseases in a dish: The promise of iPSC technology in disease modeling and therapeutic discovery. *Neurol. Sci.* 36:21–27. PMCID:PMC4282683
31. Burrows CK, Banovich NE, Pavlovic BJ, Patterson K, Gallego Romero I, Pritchard JK, Gilad Y. (2016) Genetic variation, not cell type of origin, underlies the majority of identifiable regulatory differences in iPSCs. *PLoS Genet.* 12:e1005793. PMCID:PMC4727884
32. DeBoever C, Li H, Jakubosky D, Benaglio P, et al. (2017) Large-scale profiling reveals the influence of genetic variation on gene expression in human induced pluripotent stem cells. *Cell Stem Cell* 20:533–546.e7. PMCID:PMC5444918
33. Kilpinen H, Goncalves A, Leha A, Afzal V, et al. (2017) Common genetic variation drives molecular heterogeneity in human iPSCs. *Nature* 546:370–375. PMCID:PMC5524171
34. Volpato V, Webber C. (2020) Addressing variability in iPSC-derived models of human disease: Guidelines to promote reproducibility. *Dis. Model Mech.* 13:PMCID:PMC6994963
35. Carcamo-Orive I, Hoffman GE, Cundiff P, Beckmann ND, et al. (2017) Analysis of transcriptional variability in a large human iPSC library reveals genetic and non-genetic determinants of heterogeneity. *Cell Stem Cell* 20:518–532.e9. PMCID:PMC5384872
36. Arrasate M, Finkbeiner S. (2005) Automated microscope system for determining factors that predict neuronal fate. *Proc. Natl. Acad. Sci. U. S. A.* 102:3840–3845. PMCID:PMC553329
37. Arrasate M, Finkbeiner S. (2012) Protein aggregates in Huntington's disease. *Exp. Neurol.* 238:1–11. PMCID:PMC3909772
38. Arrasate M, Mitra S, Schweitzer ES, Segal MR, Finkbeiner S. (2004) Inclusion body formation reduces levels of mutant huntingtin and the risk of neuronal death. *Nature* 431:805–810. PMCID:PMC15483602
39. Miller J, Arrasate M, Brooks E, Libeu CP, et al. (2011) Identifying polyglutamine protein species *in situ* that best predict neurodegeneration. *Nat. Chem. Biol.* 7:925–934. PMCID:PMC3271120
40. Mitra S, Tsvetkov AS, Finkbeiner S. (2009) Single neuron ubiquitin-proteasome dynamics accompanying inclusion body formation in Huntington disease. *J. Biol. Chem.* 284:4398–4403. PMCID:PMC2640959
41. Tsvetkov AS, Arrasate M, Barmada S, Ando DM, Sharma P, Shaby BA, Finkbeiner S. (2013) Proteostasis of polyglutamine varies among neurons and predicts neurodegeneration. *Nat. Chem. Biol.* 9:586–592. PMCID:PMC3900497
42. HD iPSC Consortium. (2012) Induced pluripotent stem cells from patients with Huntington's disease show CAG-repeat-expansion-associated phenotypes. *Cell Stem Cell* 11:264–278. PMCID:PMC3804072
43. Mitra S, Tsvetkov AS, Finkbeiner S. (2009a) Protein turnover and inclusion body formation. *Autophagy* 5:1037–1038. PMCID:PMC2892253
44. Shaby BA, Skibinsk iG, Ando M, LaDow ES, Finkbeiner S. (2016) A three-groups model for high-throughput survival screens. *Biometrics* 72:936–944. PMCID:PMC4965338
45. Jalili V, Afgan E, Gu Q, Clements D, Blankenberg D, Goecks J, Taylor J, Nekrutenko A. (2020) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. *Nucleic Acids Res.* 48:W395–W402. PMCID:PMC7319590

46. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, et al. (2012) Fiji: An open-source platform for biological-image analysis. *Nat. Methods* 9:676–682. PMCID:PMC3855844
47. Zhou C, Wang G, Huang H, Song L, Xue K. (2019) Edge detection based on joint iteration ghost imaging. *Opt. Express* 27:27295–27307.
48. Bates D, Mächler M, Bolker B, Walker S. (2015) Fitting linear mixed-effects models using lme4. *J. Stat. Softw.* 67:1–48.
49. Dodge Y. (2008) Q-Q Plot (quantile to quantile plot). In: *The Concise Encyclopedia of Statistics*. Eds., Springer New York, New York, NY, Vol. pp. 437–439.
50. Krüger L, Huerta MF, Santa Cruz F, Cárdenas CA. (2021) Antarctic krill fishery effects over penguin populations under adverse climate conditions: Implications for the management of fishing practices. *Ambio* 50:560–571. PMCID:PMC7882667