The Journal of Open Source Software

experDesign: helping performing experiments on batches

Lluís Revilla Sancho[1, 2], **Juan-José Lozano**[1], **and Azucena Salas**[2]

1 Centro de Investigación Biomédica en Red, Enfermedades Hepáticas y Digestivas **2** Institut d'Investigacions Biomèdiques August Pi i Sunyer, IDIBAPS

DOI: 10.21105/joss.03358

Software

- Review
- Repository
- Archive

Editor: Lorena Pantano

Reviewers:

- @abartlett004
- @stemangiola

Submitted: 23 April 2021
Published: 11 June 2021

Summary

The design of an experiment is critical to its success. Nonetheless, even with a correct design, the process up to the moment of measurement is critical. At any one of the several steps required from collection to measurement various errors and problems could affect the experimental results. Failure to take such variability into account can render an experiment inconclusive. *experDesign* provides tools to minimize the risk of inconclusive results by specifically assigning samples to particular batches.

Introduction

To properly design an experiment, the source of the variation between samples must be identified. Typically, one can control the environment in which the study or experiment is being conducted. Sometimes, however, this is not possible, and then one needs to apply certain techniques to control for such variations. There are three techniques used to reduce unwanted variation: blocking, randomization and replication (Klaus 2015).

Blocking group samples that are equal, according to one or more variables, allows one to estimate the variation in the measurements of the factors from these variables. **Randomization** is a method to minimize the variation in the measurements by mixing the potential confounding variables. **Replication** increases the number of samples used in an experiment to better estimate the variation of the experiment. In some settings (e.g., clinical, agriculture, etc.), several of these techniques can be applied together to ensure the robustness of the study.

Between the designing of an experiment and the measurement of the samples, many things can happen. If some samples are lost, are contaminated, or do not pass quality control for measurement purposes, the unwanted variation of the experiment will increase. Even if this doesn't happen, experiments will occasionally need to be carried out in batches due to technical reasons; for example, the machine cannot measure more than one samples at a time, or because of other practical considerations; for instance, it may not be possible to obtain additional measurements in the field during the allotted time.

There are several techniques to identify and assess batch effects when analyzing an already measured experiment (Leek et al. 2010). Experiments that run over long periods of time or that are run across different laboratories are highly susceptible to batch effects. This is also true of even smaller single-laboratory studies, if they span several days or include personnel changes. However, if the source of variations are not taken into account before measuring is carried out, a batch effect can be introduced that later is almost impossible to remove. Thus, it would be better to avoid such batch effects before carrying out an experiment. Taking into account the process from the design phase to the final measurement can help prevent batch effects.

Experiments should be designed to distribute batches and other potential sources of experimental variation across groups. To prevent the occurrence of batch effects after the initial design of the experiment, there are two options: randomization and replication. These techniques can not only prevent batch effects, but can also facilitate comparisons between or within the desired variable(s). Both can be utilized, but in any case one must take blocking into account from the beginning or the problem might be further exacerbated.

Randomization, if done correctly, can helps reducing variations across groups, while replication can help one be more confident of the experimental results. For instance, if one designs an experiment with cases and controls and the latter are measured in one batch and the controls in a different batch, then any difference between them can be entirely confounded by a malfunction of the measuring machine in either one of the batches (Chen et al. 2011). However, by examining how the variables are distributed across each batch sample, proper randomized can be ensured, thus minimizing batch effects. This is known as randomized block experimental design or stratified random sampling experimental design.

Replications consist of increasing the number of measurements with similar attributes. Usually at least three samples are included for each condition of interest in order to estimate the variations of these attributes. When there is one extraction and then a sample is measured multiple times, this is referred to as a technical replicate. Technical replicates help estimate the variation of the measurement method, and thus the possible batch effect.

Replications consist on increasing the number of measurements with similar attributes. Usually at least three samples are included for each condition of interest to be able to estimate the variation of the attributes. If there is one extraction and then a sample is measured multiple times it is called a technical replicate. Technical replicates help estimate the variation of the measurement method and thus of the possible batch effect.

State of the art

There are some tools to prevent batch effects on the R language in multiple fields and areas, and particularly for biological research (R Core Team 2014). They carry some caveats limiting their application in some cases, and to our knowledge no comparisons of these various methods has been conducted. Here we briefly describe the currently available packages:

- *OSAT*, at Bioconductor, first allocates the samples from each batch according to a variable; it then shuffles the samples from each batch in order to randomize the other variables (Yan et al. 2012). This algorithm relies on categorical variables and cannot use numerical variables (e.g., age- or time-related) unless they are treated as categorical.
- *minDiff*, at github, and its successor *anticlust*, at CRAN, divide the samples into similar groups, ensuring similarity by enforcing heterogeneity within groups (Papenberg and Klau 2020). Conceptually it is similar to the clustering methods k-means.
- Recently, *Omixer*, a new package, has recently been made available at Bioconductor (Sinke, Cats, and Heijmans 2021). It tests whether the random assignments are homogeneous by transforming all variables to numeric values and using the Kendall's correlation when there are more than 5 samples; otherwise , it utilizes the Pearson's chi-squared test.
- Finally there is the package *experDesign*, at CRAN, which provides groups with characteristics similar to the entire sample by comparing the random samples with the whole dataset across several statistics.

The Journal of Open Source Software

Description

The package ***experDesign*** provides the functional `design` to distribute the samples into multiple batches such that each variable is homogeneous within each batch. It is similar to the *anticluster* method used for maximum variance. If the experiment is carried out with a specific spatial distribution, the `spatial` function also distributes the samples homogeneously by position in a manner similar to *Omixer*.

In addition to distributing the samples into batches, ***experDesign*** provides tools to add technical replicates. When the design is complete, the replicates are referred to as technical replicates (Blainey, Krzywinski, and Altman 2014).Technical replicates are samples that are measured multiple times, which reduces the uncertainty of the measurement. If the technical replicates are distributed across several batches, batch effects can be measured and thus minimized. To select the technical replicates needed and in order to choose from which samples they are calculated, the function `extreme_cases` is provided. For easier usage, the `replicates` function designs an experiment with the number of replicates per batch desired.

experDesign also provides several small utilities to make it easier to design the experiment in batches. For instance, a function called `sizes_batches` helps calculate the number of samples in order to distribute them across the number of batches required.

Acknowledgments

We are grateful to Joe Moore for English-language assistance.

References

Blainey, Paul, Martin Krzywinski, and Naomi Altman. 2014. "Replication." *Nature Methods* 11 (9): 879–80. https://doi.org/10.1038/nmeth.3091.

Chen, Chao, Kay Grennan, Judith Badner, Dandan Zhang, Elliot Gershon, Li Jin, and Chunyu Liu. 2011. "Removing Batch Effects in Analysis of Expression Microarray Data: An Evaluation of Six Batch Adjustment Methods." *PLOS ONE* 6 (2): e17238. https://doi.org/10.1371/journal.pone.0017238.

Klaus, Bernd. 2015. "Statistical Relevancerelevant Statistics, Part i." *The EMBO Journal* 34 (22): 2727–30. https://doi.org/10.15252/embj.201592958.

Leek, Jeffrey T., Robert B. Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W. Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A. Irizarry. 2010. "Tackling the Widespread and Critical Impact of Batch Effects in High-Throughput Data." *Nature Reviews. Genetics* 11 (10). https://doi.org/10.1038/nrg2825.

Papenberg, Martin, and Gunnar W. Klau. 2020. "Using Anticlustering to Partition Data Sets into Equivalent Parts." *Psychological Methods*. https://doi.org/10.1037/met0000301.

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://R-project.org/.

Sinke, Lucy, Davy Cats, and Bastiaan T Heijmans. 2021. "Omixer: Multivariate and Reproducible Sample Randomization to Proactively Counter Batch Effects in Omics Studies." *Bioinformatics*, no. btab159 (March). https://doi.org/10.1093/bioinformatics/btab159.

Yan, Li, Changxing Ma, Dan Wang, Qiang Hu, Maochun Qin, Jeffrey M. Conroy, Lara E. Sucheston, et al. 2012. "OSAT: A Tool for Sample-to-Batch Allocations in Genomics Experiments." *BMC Genomics* 13 (1): 689. https://doi.org/10.1186/1471-2164-13-689.