Well Plate Maker: A user-friendly randomized block design application to limit batch effects in largescale biomedical studies

Hélène Borges, Anne-Marie Hesse, Alexandra Kraut, Yohann Couté, Virginie Brun, Thomas Burger

HAL Id: hal-03135100

https://hal.science/hal-03135100

Submitted on 8 Feb 2021

Well Plate Maker: A user-friendly randomized block design application to limit batch effects in large-scale biomedical studies

Hélène Borges*, Anne-Marie Hesse, Alexandra Kraut, Yohann Couté, Virginie Brun, Thomas Burger*
helene.borges@cea.fr; thomas.burger@cea.fr

[1]Univ. Grenoble Alpes, CEA, Inserm, BGE U1038, 38000, Grenoble, France and
[2]CNRS, BIG FR3425, Grenoble, 38000, France.

Abstract: Many factors can influence results in clinical research, in particular bias in the distribution of samples prior to biochemical preparation. Well Plate Maker is a user-friendly application to design single- or multiple-well plate assays. It allows multiple group experiments to be randomized and therefore helps to reduce possible batch effects. Although primarily fathered to optimize the design of clinical sample analysis by high throughput mass spectrometry (e.g. proteomics or metabolomics), it includes multiple options to limit edge-of-plate effects, to incorporate control samples, or to limit cross-contamination. It thus fits the constraints of many experimental fields. Well Plate Maker is implemented in R and available at Bioconductor repository (https://bioconductor.org/packages/wpm) under the open source Artistic 2.0 license. In addition to classical scripting, it can be used through a graphical user interface, developed with Shiny technology.

***

Recent progress in the fields of instrumentation and analytical methods have considerably improved the performance and throughput of omics analyses. These types of analyses are increasingly being used to explore extensive patient cohorts, notably to discover new biomarkers, with a view to improving medical care (Geyer et al., 2017). Concomitantly, the needs for streamlined, unbiased and scalable experimental workflows for sample preparation and analysis have emerged. In particular, it is necessary to optimize the distribution of samples over the preparation plates to limit biases and uncontrolled sources of technical variability as a result of batch effects (Goh et al., 2017) or possible confounding factors (e.g. age, gender, diet (Forshed, 2017; Yan et al., 2020)). In this context, we introduce Well Plate Maker (WPM), a software tool that simplifies the randomization of the different sample groups in an experiment (biological conditions to be compared, treatments, blocking factors, etc.) while allowing for many practical constraints (buffer solution and control samples, location effects, number and type of plates, etc.).

Several tools already exist to work on plate maps, as R packages or as MATLAB toolboxes. These include plater (Hughes, 2016), highSCREEN (Shterev et al., 2018), platetools (Warchal, 2019), prada (Hahne F et al., 2019), and the microplateplot function from the Bioinformatics toolbox (MathWorks, 2020). However, WPM differs in two ways. Firstly, even though other tools are helpful for map display (Hughes, 2016; Shterev et al., 2018) or graphical summaries (Warchal, 2019; Hahne F et al., 2019; MathWorks, 2020), they cannot be configured via a Graphical User Interface (GUI). WPM includes a Shiny-based GUI (Chang et al, 2019), making it easily accessible to users with no programming skills. Secondly, other tools are dedicated to *a posteriori* analysis, to help correct for undesirable effects after the experiment has been completed. In contrast, WPM can be used upstream of the experiment to optimize the design of the plate maps. Concretely, users can specify a number of constraints, such as group membership; spatial or neighborhood constraints; plate size and number. Notably, for multiple plates, samples are evenly distributed across different plates based on the total number of samples and the group sizes, thereby preventing "plate effects". Moreover, WPM simplifies block randomization (Burger et al., 2020) to limit other types of batch effect. WPM is an R package that can be installed on Linux, Windows or MacOS operating systems by following the instructions available at the Bioconductor (Gentleman et al., 2004) repository: https://bioconductor.org/packages/release/bioc/html/wpm.html. It is launched with the following commands :

```
> library(wpm); wpm()
```
   A new tab will then open in the default browser displaying the WPM home page. WPM can also be used through command lines for seamless workflow integration. This is why, in addition to CSV/text files (preferred for the GUI), other Bioconductor formats (*ExpressionSet*, *MSnSet* and *SummarizedExperiment*) can be accessed using command-line wrappers.

   WPM identifies four different types of samples with decreasing priority: i) **Forbidden wells** that should not be filled with any kind of sample, either because the user does not want to (e.g. plate corners in case of non-uniform heat distribution), or because of material constraints (e.g. dirty wells, broken pipettes); ii) **Buffers** filled with solution but without biological material (e.g. to avoid/check for cross-contamination); iii) **Fixed samples** for quality control samples or standards, the precise location of these samples must be controlled by the researcher; iv) **Randomized samples**, split into groups according to their biological content, and which will be randomized within and between plates.

   WPM is structured across four tabs: *Home* introduces the application and contact details; *Parameters* is used to tune the different settings for plate randomization; *Results* displays the proposed plate maps; and *Help* provides a detailed tutorial for new users. A vignette is also available in the package and on the website. The two most important tabs are *Parameters* and *Results*. The *Parameters* tab is split into seven blocks, dedicated to as many specific steps:

Step 1: Select a CSV file containing the sample names and groups, if any. A visualization window helps the user to specify the import options.

Step 2: Give a project title in order to name the output files accordingly.

Step 3: Specify the number of plates and their dimensions, either with a customary or with a predefined format (6, 24, 48, 96, 386, or 1534 wells).

Step 4 to 6: Specify which wells should be **forbidden** (step 4), should contain a **buffer** (step 5) or a fixed sample (step 6); all the others being left for **randomized samples**. Moreover, during step 5, it is possible to apply a neighborhood constraint to the randomized samples.

Step 7: Before initiating plate randomization, a maximum number of attempts is selected over which the randomized placement algorithm will seek a solution meeting all the constraints defined. Our placement algorithm was inspired by the backtracking algorithm (Knuth, 2018). The principle is as follows: Test combinations, by abandoning tests that do not produce a solution (backtrack), until a result is found (among a set of possible solutions) which satisfies the constraints specified (here, spatial constraints). Therefore, if no groups are defined, any randomization is acceptable and a single iteration is sufficient. Conversely, if too many constraints are set, there may be no solution. In practice, for a very simple case, WPM finds a solution from the first attempt with an execution time of the order of one second. For more complex experimental designs, WPM performs between 10 and 80 iterations in just under a minute.

   WPM generates two types of results: a dataset and plate maps. Both are displayed and downloadable from the *Results* panel. The dataset contains the information provided by the user alongside information generated by the application. In the downloadable CSV file, sample IDs are concatenated with the project name to avoid confusing samples from different projects. On the map (downloadable as PNG), however, shortened IDs are displayed to facilitate readability (Fig. 1). The coordinates (column and row references) on the plates are also provided to facilitate placement of the samples on the plates.

Funding

References

   Burger, B. *et al.* (2020). Importance of Block Randomization When Designing Proteomics Experiments. *Journal of Proteome Research*, DOI: 10.1021/acs.jproteome.0c00536

Chang, W. *et al*. (2019). shiny: Web application framework for R. R package version 1.4.0.

Forshed, J. (2017). Experimental design in clinical 'omics biomarker discovery. *Journal of proteome research*, 16(11), 3954–3960.

Gentleman, R. C. *et al*. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10), R80.

Geyer, P. E. *et al*. (2017). Revisiting biomarker discovery by plasma proteomics. *Molecular systems biology*, 13(9).

Goh, W. W. B. *et al*. (2017). Why batch effects matter in omics data, and how to avoid them. *Trends in biotechnology*, 35(6), 498–507.

Hahne F. *et al.* (2019). prada: Data analysis for cell-based functional assays. R package version 1.63.0.

Hughes, S. M. (2016, nov). plater: Read, tidy, and display data from microtiter plates. *The Journal of Open Source Software*, 1(7).

Knuth, D. E. (2018). Art of computer programming: The mathematical preliminaries redux; backtracking; dancing links. *Addison Wesley Professional,* 4b(5).

MathWorks. (2020). Bioinformatics toolbox: User's guide (r2020a) [Computer software manual]. Natick, Massachusetts, United State.

Shterev, I. D. *et al*. (2018). highscreen: High-throughput screening for plate based assays [Computer software manual].

Warchal, S. (2019). platetools: Tools and plots for multi-well plates . R package version 0.1.2.

Yan, H. *et al*. (2020). Comparing effects of treatment: Controlling for confounding. *Neurosurgery*, 86(3), 325–33.

A)

Sample	Group	ID	Well	Row	Column	Plate
8	1	5_Projecttitle	F2	6	2	2
10	1	6_Projecttitle	C9	3	9	2
59	1	30_Projecttitle	B4	2	4	2
128	1	79_Projecttitle	E9	5	9	2
223	1	158_Projecttitle	E6	5	6	2
3	3	2_Projecttitle	C4	3	4	2
12	3	7_Projecttitle	B9	2	9	2
31	3	17_Projecttitle	F5	6	5	2
48	3	25_Projecttitle	G3	7	3	2
54	3	28_Projecttitle	E7	5	7	2

B)
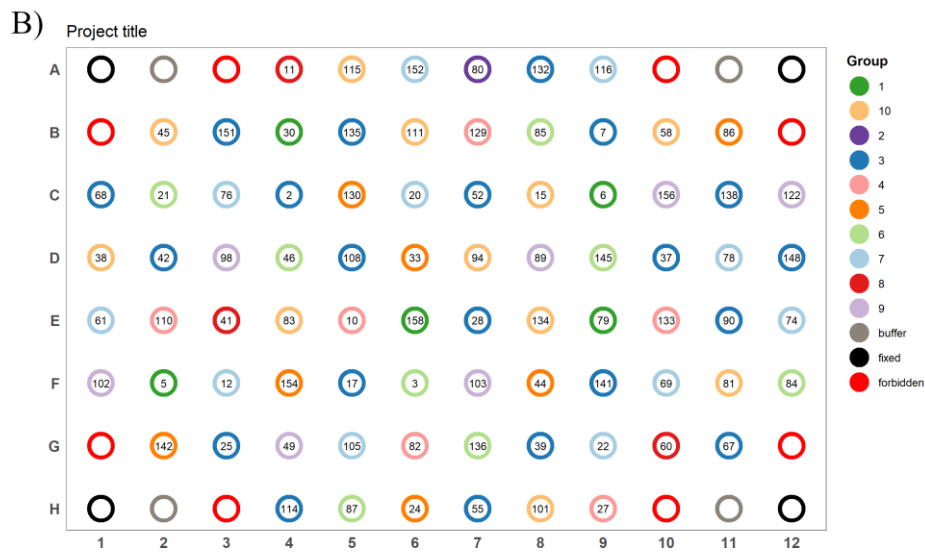
Figure 1 A) Sample output dataset listing experimental samples and their corresponding wells. B) Sample 96-well plate containing 80 samples (divided into 10 groups of different sizes), 8 forbidden wells (in red, unnumbered), 4 fixed wells (in black, unnumbered), 4 blank wells (in grey, unnumbered). The neighborhood constraints chosen were North/East/West/South. Wells containing samples are colored according to their group and labelled with their ID.