Bioassays: a new package in R for analyzing cellular assays

Anwar Azad Palakkan[1]

1 Deanery of Biomedical Sciences, University of Edinburgh, UK

Summary

Experimental techniques that acquire data from multi-well plates are very common in life science research; today they account for half of all assays used in drug development and toxicity testing. These experiments include colorimetric, fluorometric and luminometric assays for cytotoxicity / cell viability (Markossian et al., 2004), bacterial assays (Burton, Yakandawala, LoVetri, & Madhyastha, 2007; Pant et al., 2016) and immunoassays (Lin, Salvador, & Carter, 2015; Song et al., 2015). Such assays feature in small-scale experiments and also in high-throughput screens. High-throughput imaging and spectrofluorometric quantification-based methods are increasingly used in the pharmaceutical industry and in academic research aiming to screen large libraries (e.g. potential drug compounds, RNAi, CRISPR/Cas9) or to quantify rare biological events.

Plate-reading equipment is usually supplied with accompanying analytic software, such as Gen5 (BioTek), Multi-Mode Analysis Software (Molecular Devices), and Multiwell-Analyzer (Multi Channel Systems). There are, however, drawbacks to its use. Usually, the software is bound to the data coming from a specific piece of equipment, it is seldom possible to use the software on multiple computers without making additional license purchases. This is a problem in academic settings in which a single machine is shared and used intensively by several research groups. Also, proprietary software has closed code that cannot be checked or understood in detail by the user and cannot be expanded or adapted for analyses not envisaged by the equipment manufacturer.

In response to these limitations, some open-source software packages have been developed to serve particular unusual applications such as single-cell migration (Masuzzo et al., 2017) or differential scanning fluorimetry (Wang, Weeratunga, Pacheco, & Hofmann, 2012). Open software with more general capabilities like DRfit (Hofmann et al., 2019) that run on JAVA platforms has been developed, but they are not ideal for high throughput screening. Many biologists are familiar with the statistical language, R, and R libraries have been created to assist with analysis of data from plate readers. Examples include 'platetools' (https://cran.r-project.org/package=platetools), 'plater' (https://cran.r-project.org/package=plater) and 'phenoScreen' (https://rpubs.com/Swarchal/phenoScreen), each of which addresses particular functions. 'Plater', for example, is focused on data formatting, 'phenoScreen' on data visualization, and 'platetools' on both data visualizing and formatting. The packages are useful, but can be difficult for a beginner to grasp, partly because their documentation seems not to have been written with beginners in mind.

Here, we introduce a new open-source package, 'bioassays', designed to provide a wide range of functions relevant to multi-well plate assays. 'Bioassays' is an R package freely available on CRAN (R Core Team, 2014), and has supports for formatting, visualizing and analyzing multi-well plate assays. It has functions for handling outliers, for handling multiple data sets with separate blanks, for estimating values from standard curves, for summarizing data and for doing statistical analysis. Moreover, we have provided examples (bioassays-examples) in

---

the vignettes, which will be very helpful for beginners to grasp how this package can be used. Because this is a R package, more experienced users will be able add to its functionality using other R packages.

Bioassays

Bioassays can handle data from any of the standard multi-well plate format: 6, 12, 24, 96 or 384 well plates. A prerequisite for the package is the need for both input data (Figure 1) and metadata (Figure 2) to be in comma-separated variable (csv) format; most plate readers can export data as a .csv file. Metadata need "row" and "col" columns to indicate the location of well. The package has functions for data extraction (extract_filename), formatting (data2plateformat, plate2df, matrix96 and plate_metadata), visualization (heatplate) and data analysis (reduceblank, estimate, dfsummary, pvalue).

The function extract_filename is useful for extracting specific information from file names, such as compound name, plate number etc., which can be used for automated data analysis. This function provides a very easy way to pass data of this type into the analysis by simply editing file names.

The function data2plateformat converts the data (eg: readings from a 96 well plate) to an appropriate matrix format with suitable row and column names. The plate2df function can convert such data into a data frame with 'row' and 'col' columns to indicate well position. The function matrix96 can convert both character and numeric columns to a matrix. This function can automatically determine the type of multi-well from which the data is coming, and it have provision to convert negative and NA values to 0, if needed. The plate_metadata function combines plate-specific information (such as compound used, standard concentration, dilution of samples, etc.), to produce unique plate metadata.

For data visualization, the 'heatplate' function can be used. This can plot both heat plots (Figure 3) and categorical plots (Figure 4) automatically, depending on the data type. The 'heatplate' function also has provisions for adjusting the displayed well size, for visual esthetics. Any multi-well plate data can be structured for the plots (function 'heatplate') by using functions 'data2plateformat', 'plate2df', 'matrix96' alone or in combinations (depending on the input data).

The function 'reduceblank' can help to reduce blank values from the readings. This function can handle separate blanks for different datasets. The function 'estimate' can be used to estimate unknown variable (eg: concentration) based on standard curve. The function 'dfsummary' is really versatile for handling multiple data sets. It can group samples and summarize data sets separately. It has additional controls for handling outliers and omitting unwanted data sets. Function 'pvalue' can be used to test significance by t-test. It has provisions for asserting control group and level of significance.

Availability and installation

Bioassays is supported on Windows and macOS. The package can be installed using install.packages command in R. The source code, vignette, datasets and detailed examples on how to use the package are available on CRAN (https://CRAN.R-project.org/package=bioassays) and GitHub (https://github.com/anwarbio/bioassays).

Palakkan, A. A., (2020). Bioassays: a new package in R for analyzing cellular assays. *Journal of Open Source Software*, 5(54), 2402. https://doi.org/10.21105/joss.02402

Acknowledgment

Figures

```
#>   X    X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12
#> 1 A 0.659 0.649 0.598 0.601 0.541 0.553 0.568 0.519 0.576 0.575 0.583 0.504
#> 2 B 0.442 0.455 0.586 0.563 0.525 0.548 0.511 0.503 0.533 0.559 0.529 0.535
#> 3 C 0.278 0.266 0.491 0.562 0.510 0.473 0.467 0.433 0.382 0.457 0.475 0.510
#> 4 D 0.197 0.199 0.452 0.456 0.421 0.431 0.409 0.401 0.458 0.412 0.408 0.403
#> 5 E 0.177 0.174 0.447 0.437 0.392 0.412 0.368 0.396 0.397 0.358 0.360 0.393
#> 6 F 0.141 0.137 0.277 0.337 0.294 0.279 0.257 0.263 0.262 0.292 0.280 0.300
```

Figure 1: Input data format from a 96 well plate reading.

```
#>   row col position type       id concentration dilution
#> 1   A   1      A01 STD1      STD            25       NA
#> 2   A   2      A02 STD1      STD            25       NA
#> 3   A   3      A03   S1   Sample            NA       NA
#> 4   A   4      A04   S1   Sample            NA       NA
#> 5   A   5      A05   S1   Sample            NA       NA
#> 6   A   6      A06   S1   Sample            NA       NA
```

Figure 2: Metafile data format.
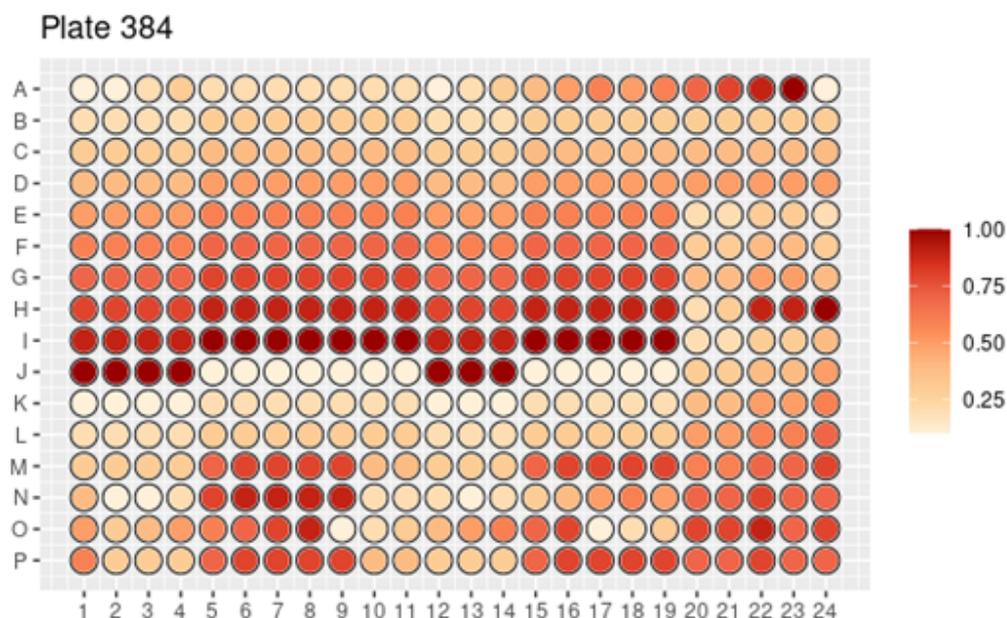
Figure 3: Heat map of 384 well (normalized values).

Plate 384
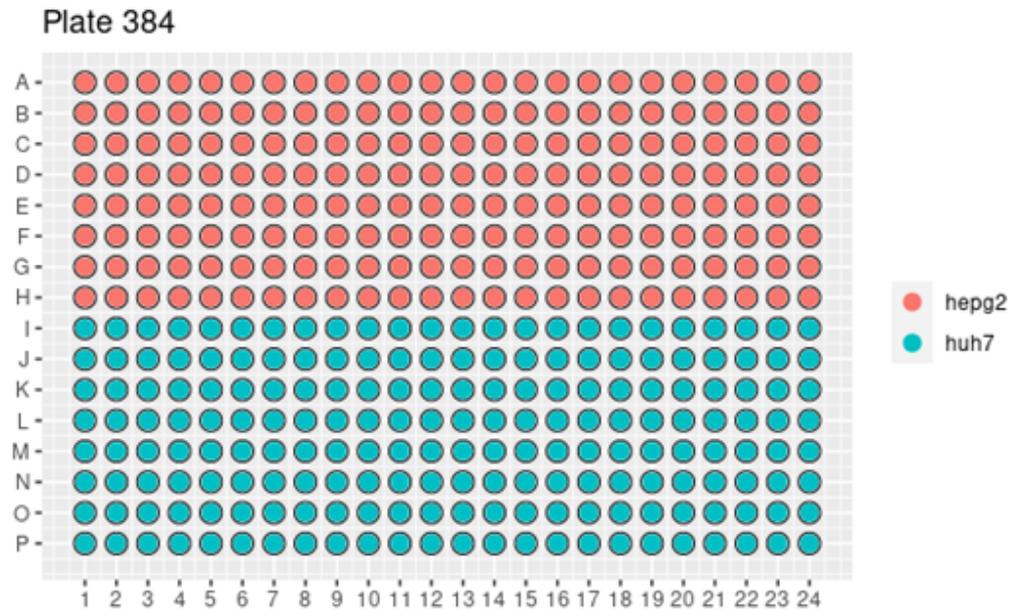
Figure 4: Categorical plot of 384 well plate.

References

Burton, E., Yakandawala, N., LoVetri, K., & Madhyastha, M. S. (2007). A microplate spectrofluorometric assay for bacterial biofilms. *J Ind Microbiol Biotechnol*, *34*(1), 1–4. doi:10.1007/s10295-006-0086-3

Hofmann, A., Preston, S., Cross, M., Herath, H. M. P. D., Simon, A., & Gasser, R. B. (2019). DRfit: A java tool for the analysis of discrete data from multi-well plate assays. *BMC Bioinformatics*, *20*(1), 262. doi:10.1186/s12859-019-2891-5

Lin, A., Salvador, A., & Carter, J. M. (2015). Multiplexed microsphere suspension array-based immunoassays. *Methods Mol Biol*, *1318*, 107–18. doi:10.1007/978-1-4939-2742-5_11

Markossian, S., Sittampalam, G. S., Grossman, A., Brimacombe, K., Arkin, M., Auld, D., Austin, C. P., et al. (2004). Assay guidance manual. In. Eli Lilly & Company; the National Center for Advancing Translational Sciences. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK53196/

Masuzzo, P., Huyck, L., Simiczyjew, A., Ampe, C., Martens, L., & Van Troys, M. (2017). An end-to-end software solution for the analysis of high-throughput single-cell migration data. *Sci Rep*, *7*, 42383. doi:10.1038/srep42383

Pant, K., Bruce, S., Sly, J., Klug Laforce, M., Springer, S., Cecil, M., Andrus, E., et al. (2016). Bacterial mutagenicity assays: Vehicle and positive control results from the standard Ames assay, the 6- and 24-well miniaturized plate incorporation assays and the Ames II™ assay. *Environ Mol Mutagen*, *57*(6), 483–96. doi:10.1002/em.22014

R Core Team. (2014). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org/

Song, E., Yu, M., Wang, Y., Hu, W., Cheng, D., Swihart, M. T., & Song, Y. (2015). Multi-color quantum dot-based fluorescence immunoassay array for simultaneous visual detection

of multiple antibiotic residues in milk. *Biosens Bioelectron*, *72*, 320–5. doi:10.1016/j.bios.2015.05.018

Wang, C. K., Weeratunga, S. K., Pacheco, C. M., & Hofmann, A. (2012). DMAN: A java tool for analysis of multi-well differential scanning fluorimetry experiments. *Bioinformatics*, *28*(3), 439–40. doi:10.1093/bioinformatics/btr664