LAB-AID (Laboratory Automated Interrogation of Data): an interactive web application for visualization of multi-level data from biological experiments

Zrinko Kozic[1,4], Sam Booker[1,2,4], Owen Dando[1,2,3,4*], Peter Kind[1,2,4]

1 Simons Initiative for the Developing Brain, University of Edinburgh, Edinburgh, United Kingdom
2 Patrick Wild Centre, University of Edinburgh, Edinburgh, United Kingdom
3 UK Dementia Research Institute, University of Edinburgh, Edinburgh, United Kingdom
4 Centre for Discovery Brain Sciences, University of Edinburgh, Edinburgh, United Kingdom

* owen.dando@ed.ac.uk

Abstract

A key step in understanding the results of biological experiments is visualization of the data. Many laboratory experiments contain a range of measurements that exist within a hierarchy of interdependence. An automated way to visualise and interrogate experimental data would: 1) lead to improved understanding of the results, 2) help to determine which statistical tests should be performed, and 3) easily identify outliers and sources of batch effects. Unfortunately, existing graphing solutions often demand expertise in programming, require considerable effort to import and examine such multi-level data, or are unnecessarily complex for the task at hand. Here we present LAB-AID (**Lab**oratory **A**utomated **I**nterrogation of **D**ata), a simple tool specifically designed to automatically visualize and query hierarchical data resulting from biological experiments.

Introduction

There is a great demand for biologists to be able to visualize data from their laboratory experiments. This is particularly true when measurements exist within a hierarchy of interdependence, and when multiple sources of batch effects may influence experimental results. For example, in an investigation into the electrophysiological properties of neurons in the brain of a rodent model of a particular neurodevelopmental or neurodegenerative condition, diverse measurements are made on individual neurons whose behaviour is not independent of each other (e.g. they were subject to the same developmental program that gave rise to a functioning brain). From a single animal, multiple neurons may be probed within an individual brain slice or multiple slices. Furthermore, multiple animals be tested which may or may not be litter mates or have shared cages. Random effects (such as slice, animal or litter) at any level within this hierarchy can consistently impact groups of measurements leading to misinterpretation of the results. Moreover, data may be gathered by different investigators on several dates using diverse experimental equipment, resulting in potential batch effects that may further complicate interpretation. These random and batch effects can be difficult to disentangle from the factor being investigated. In addition, it is often important to understand the relationship or correlation between different measurements made on the same experimental unit (for example, multiple different electrophysiological measurements made on each neuron, or multiple behavioural measurements made on each animal).

Thus, it is important to thoroughly visualize and understand such data before proceeding to statistical tests of effects, otherwise misleading, inappropriate or even spurious results may be obtained. The proper visualisation of the data can also help researchers guard against common statistical errors (e.g. pseudoreplication). In addition, visualization allows for the identification of outliers at all levels within the hierarchy of measurements, and may help to determine cases of erroneous measurement or recording of data.

While numerical and statistical computing environments such as R [1] and MATLAB [2] provide a rich and powerful framework within which experimental data can be examined and plotted, their use is not necessarily straightforward for those without experience or training in programming. Dedicated scientific graphing software tools such as GraphPad Prism [3] provide graphical user interfaces which may be more approachable for the laboratory scientist, a number of online tools have recently been developed to automatically plot experimental data provided in the form of a spreadsheet or table (for example [4], [5], [6], [7], [8]), and more general online plotting tools are available (such as Plotly Chart Studio [9] and RAWGraphs [10]). However, such software may be more oriented towards examining a single experimental variable at a time, to producing publication-ready plots rather than querying data, or towards providing a wide-range of general features and plot types at the price of some complexity of use.

Hence it would be of considerable use to laboratory scientists to have a tool which could automatically plot such multi-level data from biological experiments, by which they could examine distributions, correlations between variables, outliers and batch effects. Here we present LAB-AID (**Lab**oratory **A**utomated **I**nterrogation of **D**ata), a tool specifically designed to visualize and query multi-level data resulting from biological experiments, that is easy to install, configure and use.

Materials and methods

Example data

As a demonstration of LAB-AID, we use experimental data in which the morphology, function and integration of dendritic spines was studied in layer 4 spiny-stellate cells from the somatosensory cortex of a mouse model of a monogenic neurodevelopmental disorder, and compared with the behaviour of neurons of wild-type mice. In the particular data shown, intrinsic electrophysiological properties were measured on neurons from wild-type and knockout mice; multiple cells were tested per brain slice, and multiple brain slices per experimental animal.

Technology

LAB-AID is implemented using Shiny [11], a package for the R programming language which allows interactive web applications to be easily built. Plots are produced with the ggplot2 [12], pheatmap [13] and patchwork [14] R packages, while interactive graph functionality is provided by the plotly package [15]. A full list of R packages required, and the versions that have been tested, is provided in Table 1.

Table 1. R packages used by LAB-AID.

R package	Version tested	Reference
jsonlite	1.6	[16]
magrittr	1.5	[17]
patchwork	0.0.1	[14]
pheatmap	1.0.10	[13]
plotly	4.8.0	[15]
readxl	1.1.0	[18]
reshape2	1.4.3	[19]
shiny	1.1.0	[11]
shinyjs	1.0	[20]
shinyWidgets	0.4.5	[21]
tidyverse (includes ggplot2)	1.2.1	[22]
WriteXLS	4.0.0	[23]

Input data format

Data for LAB-AID should take the form of an Excel spreadsheet (both *.xls* and *.xlsx* files are supported) or a comma-separated values (CSV) file. Data tables should be in wide format, in which each row comprises metadata and measurements of experimental variables on a single experimental unit (for example, a cell). Columns containing metadata should appear first. In the parlance of mixed modelling, these will include both fixed and random effects; for example, our example data sheet contains a column indicating the genotype of the animal from which each measured neuron was derived (a fixed effect), and also columns indicating the brain slice and animal ID for each cell (random effects). Entries in metadata columns should be provided for every experimental unit.

Data columns should follow the metadata columns. Often, multiple data columns will be present, representing multiple simultaneous measurements made on the experimental unit. Not all measurements need to be recorded for every experimental unit.

Installation and configuration

RShiny application installation

As LAB-AID is implemented using Shiny, it can be installed or deployed in the same way as any other Shiny application. For example, the code can be run on the user's machine using RStudio [24], deployed to a web server using Shiny Server, or, if local server resources are not available, deployed to the cloud using shinyapps.io (https://www.shinyapps.io).

JSON configuration

LAB-AID configuration is stored in a simple JSON (JavaScript Object Notation, https://json.org) file residing in the same directory as the application code, which describes the location and structure of the input data. The configuration file describes a number of data sets, each of which corresponds to a particular Excel spreadsheet or CSV file. For each data set, the following must be defined:

1. **Name**: A name for the data set, which will appear in the LAB-AID user interface.
2. **Path**: Path to the Excel or CSV file containing data for this set, relative to the location of the LAB-AID code directory.
3. **n.Factors**: The number of metadata columns in this data set.
4. **Description** (optional): A description of this particular data set which will appear on the **About** tab.

Note that while all data sets defined in the configuration file will be available for exploration in the user interface, data sets do not need to contain the same metadata columns, nor do they need contain measurements on the same experimental units.

In addition, the main application title and description are also configured in the appropriate JSON fields:

1. **Title**: An overall title for the application which will appear at the left of the main tab bar.
2. **About**: An overall description of the data sets presented which will appear on the **About** tab.

The **Configuration** tab within the application provides an interface to add and remove data sets and to modify the application title and description. Alternatively, the configuration file can be edited manually.

Availability

LAB-AID is freely available for download from GitHub under the MIT License, at https://github.com/sidbdri/LAB-AID.

Results

The LAB-AID user-interface is split into six major sections (see Fig 1), available from the main tab bar, with different functions:

1. **Data Select**: Choose, depending on their metadata, which data points will be plotted, and which highlighted. Preview data selections and highlighting.
2. **Data Plots**: View box or violin plots, histograms or density plots, or cumulative frequency distributions of chosen data, comparing between and across levels of fixed and random effects.
3. **Correlation**: View correlations between experimental variables, calculated across experimental units, and comparisons of correlations between different levels of particular effects.
4. **Downloads**: Export subsets of the input data for download.
5. **Configuration**: Configure the data sets available for investigation, and modify the application title and description.
6. **About**: Read information describing the presented data sets.

Fig 1. The LAB-AID interface is split into six major sections. The **Data Select**, **Data Plots**, **Correlation**, **Downloads**, **Configuration** and **About** sections are accessible from the main tab bar. Here the **Data Plots** tab is selected.

Data selection and highlighting

A drop-down selection box at the top of the **Data Select** tab allows the user to choose the data set (of those defined in the JSON configuration file) to be currently displayed. Below this, three further tabs give access to data selection, highlighting and preview functionality.

Next to the **Dataset** selection box is the multiple selection drop-down box **Averaging factors**, that allows the user to select metadata factors over which the measured variables can be averaged. A common use

case would be to allow measurements to be shown averaged for individual animals (i.e. animal means), but it also possible to average over different experimental conditions, such as experimental dates, or animal models. To achieve averaging over brain regions within each animal in the example data set, all factors except *slice* and *cell* should be selected. Whether individual or average measurements will be plotted can be selected in the sidebar menu on the **Data Plots** tab (see below).

Selecting data to be plotted

The **Selection** tab allows the user to choose which data are to be included in graphs (see Fig 2). A drop-down selection box is present for each column of metadata in the current data set. These allow subsets of the data to be included in or excluded from plots, depending on the metadata characteristics of each experimental unit. Individual levels of each metadata factor can be selected or deselected individually; in addition, **Select All** and **Deselect All** buttons allow for bulk selection and deselection. By default, all data in the data set is selected to be shown in plots.

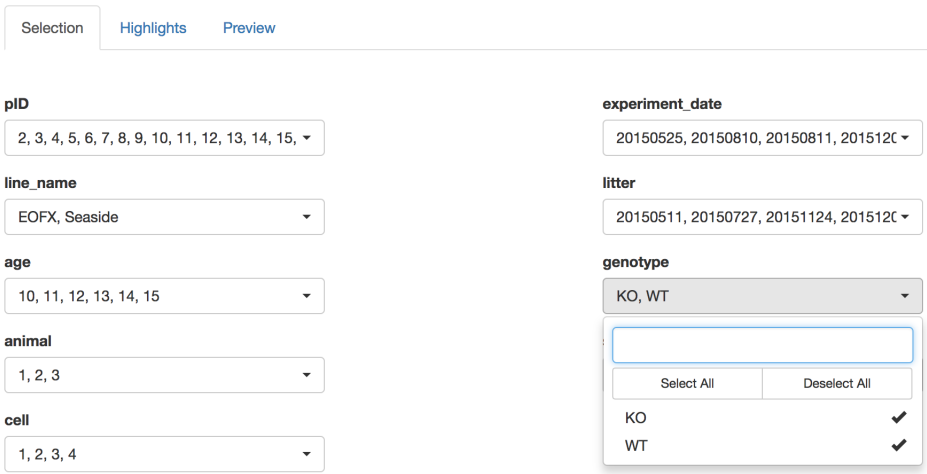

Fig 2. The Selection tab allows data subsets to be included or excluded from plots. Inclusion or exclusion is based on the metadata characteristics of each experimental unit.

Highlighting data subsets

The **Highlights** tab allows the user to choose which data, among that included in the graphs, is to be further highlighted in colour (see Fig 3). Its user interface is analogous to that of the **Selection** tab, with a drop-down selection box for column of metadata in the current data set; data points are highlighted according to the metadata characteristics of each experimental unit. By default, no data are selected to be highlighted.

Previewing data plots

The **Preview** tab presents a simple plot of the distribution of data for each experimental variable within the current data set. Individual data points are shown (see Fig 4), and overall distributions can be represented as either box or violin plots. Data are graphed according to the different levels of a single effect, chosen from the metadata columns in the input data. Preview plots respect the selection and highlighting choices made in the **Selection** and **Highlights** tabs.

Hovering over any data point in the **Preview** tab (or in the main **Data Plots** tab) will display a tooltip showing metadata attributes for that data point. In addition to the metadata derived from the input Excel or CSV file, a point ID (pID) attribute is shown, which corresponds to the relevant row in the input file, thus allowing easy location of points of interest within the input data. The pID attribute can also be used to highlight particular points, or exclude them from plots.

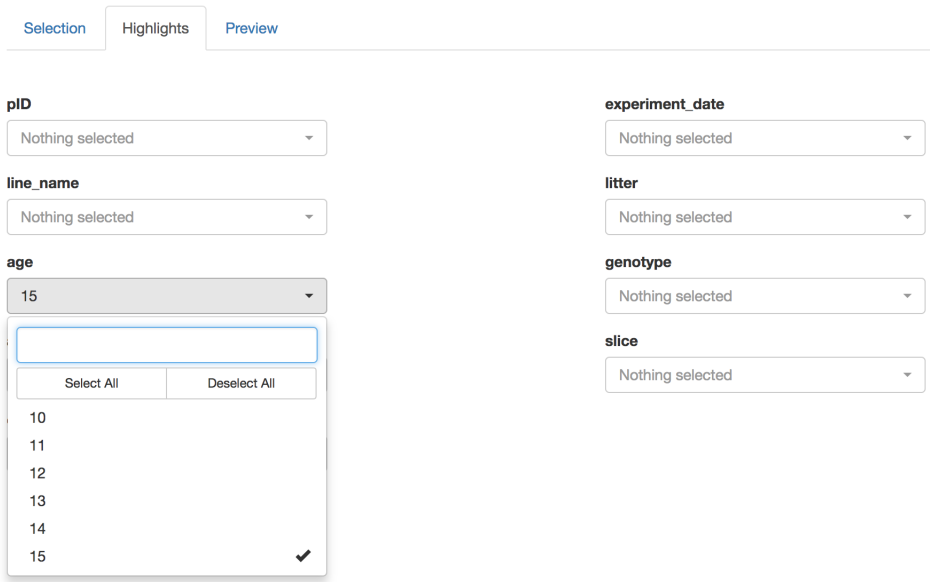

Fig 3. The Highlights tab allows data subsets to be highlighted in colour. Highlighting is based on the metadata characteristics of each experimental unit. In the example shown, data from cells originating from animals aged 15 days are selected to be highlighted.

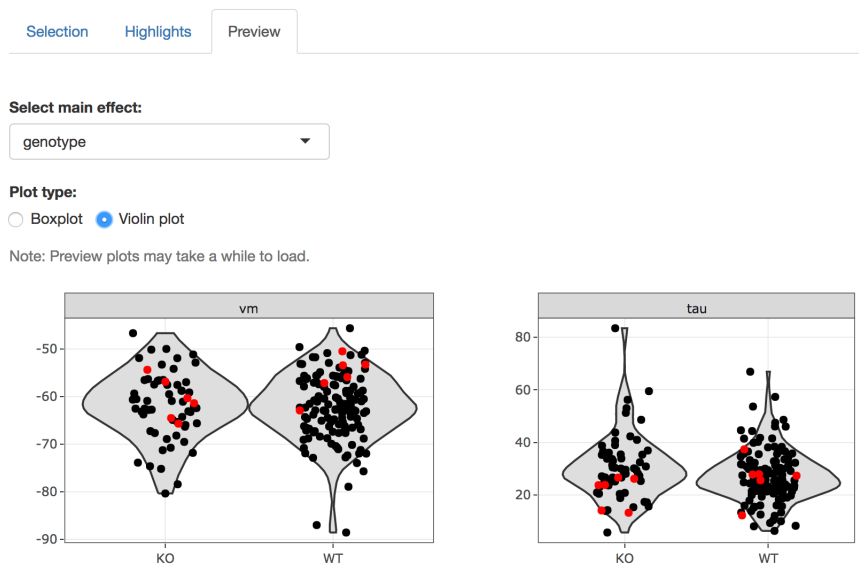

Fig 4. Data selection and highlighting choices can be inspected in the Preview tab. The Preview tab also presents an overview of the distribution of data for all experimental variables within a particular data set. Here, the data from cells originating from animals aged 15 days have been selected to be highlighted (in red).

Data plots

The behaviour of particular experimental variables can be examined in detail in the top-level **Data Plots** tab. The main area of the tab shows either (i) data distribution plots, (ii) histogram or density plots, or (iii) cumulative frequency plots, for a particular experimental variable of interest; choice between these three

modes is made via the **Data display** radio buttons. At the left of the **Data Plots** tab, a sidebar provides further control over the plots shown in the main area. Some sidebar elements are common to all plot types, whereas others change depending on the plot type.

Among the common controls, the **Data** radio buttons determine whether the data plotted is for individual experimental units (i.e. for each row in the input data file), or if it should be averaged over the levels of metadata factors selected in **Averaging factors** selection box under the **Data Select** tab. The last four selection boxes control which variable will be plotted (**Output variable**) and how data should be sub-divided for comparison. The **Main effect** selection box determines the metadata factor of primary interest; how this is represented depends on the type of plot being shown, and will be further described below. The **X facet** and **Y facet** selection boxes allow for the further subdivision of the data into separate columns and rows of graphs, respectively, according to the levels of two additional metadata factors.

Distributions

If the **Distribution plots** radio button is selected, then individual data points (or averages) for the chosen experimental variable are plotted, along with overall distributions (see Fig 5); data selection and highlighting choices made in the **Data Select** tab are respected. In each graph displayed, data are separated on the x-axis into separate levels of the main effect chosen in the left sidebar, and the overall distribution shape is shown for each. If no selections are made in the **X facet** and **Y facet** selection boxes, then a single graph is shown containing all data. In the sidebar, radio buttons specific to this plot type control whether overall distributions will be represented as box or violin plots.

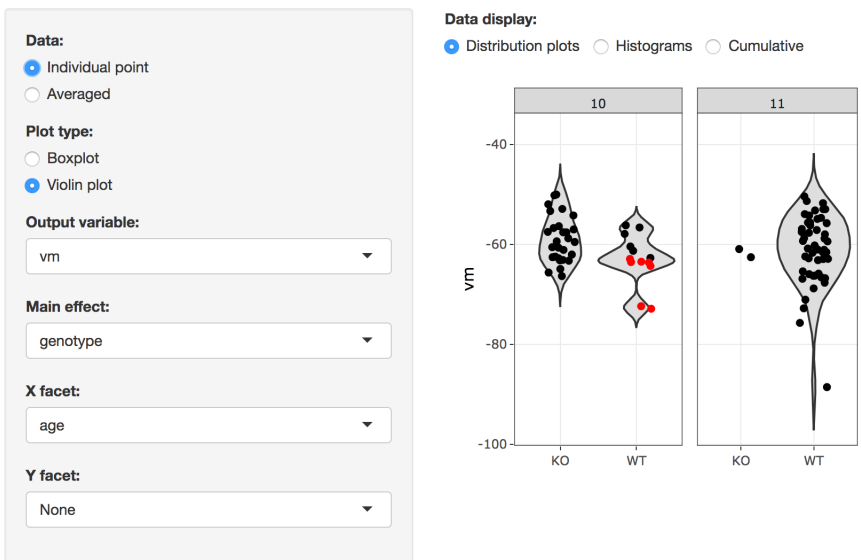

Fig 5. Distribution plots show individual data points together with box or violin plots for a particular experimental variable. Here, in our example data set, neuronal resting membrane potential is plotted across the two levels of the Genotype effect (i.e. for wild type or knockout mice). Age has been chosen as the X facet, and hence the data are subdivided into a row of separate graphs for each value of this factor. Data points derived from animals born in a particular litter have been chosen to be highlighted.

Histograms

When the **Histograms** radio button is selected, data for the chosen experimental variable are plotted as a histogram or density plot, as determined by the **Plot type** radio buttons in the sidebar (see Fig 6). If data are plotted as a histogram, then the **Bins** slide control in the sidebar determines the number of bins. Data

are separated by colour of bars (histogram) or line (density plot) into different levels of the main effect chosen in the sidebar. If no selections are made in the **X facet** and **Y facet** selection boxes, then a single histogram or density plot is shown representing all data, otherwise the data can be subdivided into multiple histograms or density plots according to the levels of one or two extra factors.

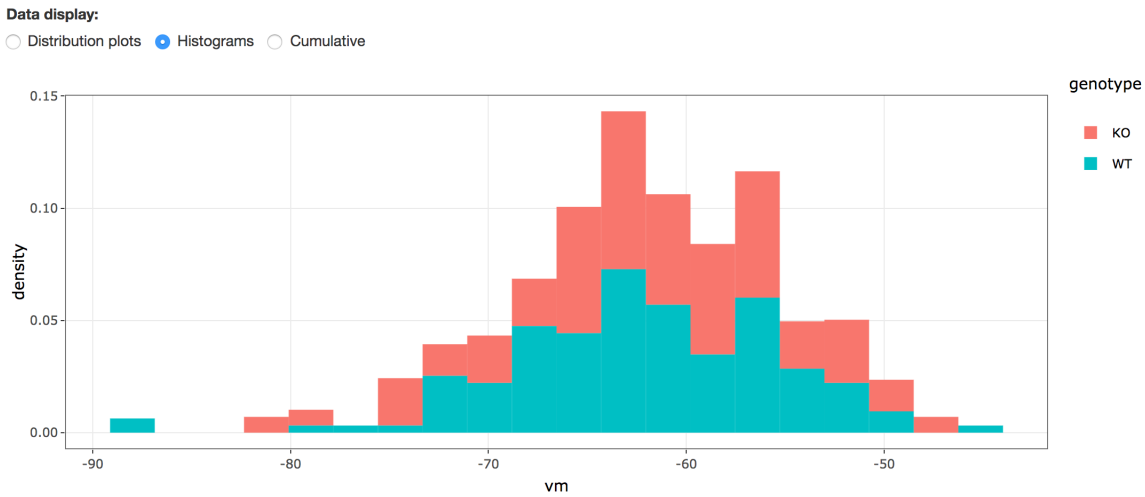

Fig 6. Data can also be represented as histogram or density plots. Here, a histogram of neuronal resting membrane potential is plotted for the two levels of the Genotype effect (for wild type or knockout mice).

Cumulative frequency distributions

Finally, if the **Cumulative** radio button is selected, data for the chosen experimental variable are plotted as a cumulative frequency distribution (see Fig 7). Data are separated by colour of line into different levels of the main effect chosen in the sidebar. As before, a single plot encompassing all data can be shown, or the data can be subdivided according to one or two additional factors. There are no plot-type specific controls in the sidebar for cumulative frequency distributions.

Correlations

The top-level **Correlations** tab allows the investigation of associations between experimental variables (see Fig 8). Pearson correlations are calculated between each pair of variables, calculated over all experimental units by default (but respecting the data selections made on the **Data Select** tab). Selecting an effect allows the user to subset the correlation calculation to those experimental units belonging to a particular level of a metadata factor (the effect and level are chosen from selection boxes in the sidebar at the left of the main plot). Correlations are drawn as a heat map, where colours run from blue (strong negative correlation) to red (strong positive correlation).

If the **Pairwise** check box is selected then two heat maps are displayed, allowing the comparison of variable correlations calculated from different subsets of the data, defined by two levels of the chosen metadata factor (see Fig 9). This allows changes in correlation between experimental conditions to be examined.

Finally, when the **Clustered** check box is selected then hierarchical clustering of experimental variables is performed, grouping together variables which show similar patterns of correlation against all the other variables (see Fig 10). Otherwise, variables are displayed along the x- and y-axes of the heatmap in the order that they are defined in the input data file.

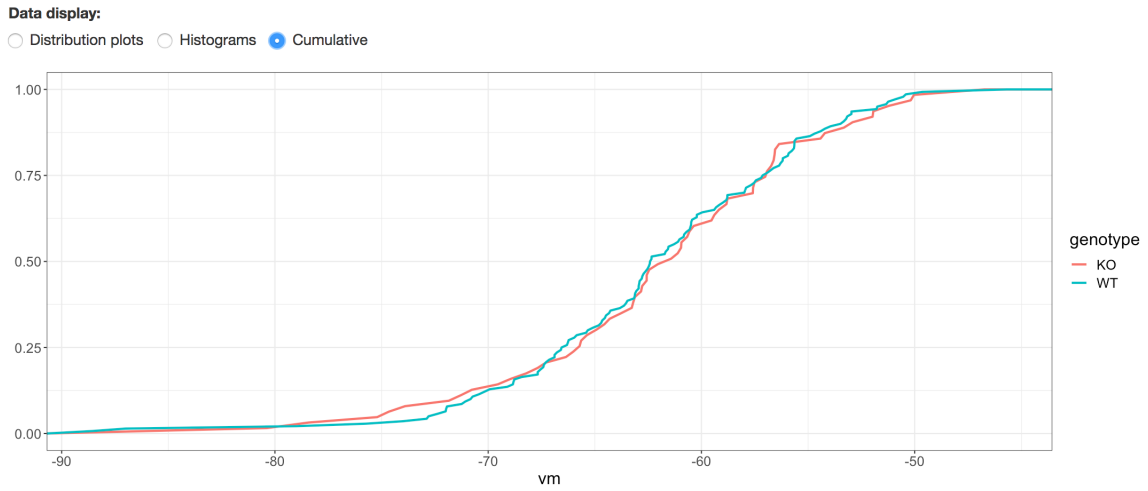

Fig 7. Representation of data as a cumulative frequency distribution. In our example data set, neuronal resting membrane potential is plotted for the two levels of the Genotype effect (for wild type or knockout mice).

Downloads

The top-level **Downloads** tab allows subsets of the input data to be exported as a new Excel spreadsheet or CSV file (see Fig 11). The two boxes at the top of the page determine what combination of metadata and experimental variable columns will be present in the exported file. The table at the bottom of the page previews the data rows that will be exported. Boxes at the head of each column allow the data to be filtered prior to export. Data rows can either be filtered via selecting particular values of the metadata factors, or by selecting ranges of the experimental variables. Finally, data are exported via the **Download .xlsx** button as an Excel spreadsheet, or **Download .csv** button as a CSV file.

Configuration

Rather than having to manually edit LAB-AID's JSON configuration file, the top-level **Configuration** tab allows the user to add and remove data sets from within the application (see Fig 12). To add a new data set, the user can upload a CSV file or an Excel spreadsheet containing data which adheres to the format previously described. Text fields allow the data set name, number of experimental factors, and data set description to be entered; while the data set description is optional, both data set name and the number of experimental factors must be provided for the data set to be successfully loaded. Newly added data can be immediately accessed on the **Data selection** tab. To remove a data set, the user can simply select the data set from the **Remove dataset** drop-down menu and press the **Remove** button.

At the bottom of the tab there are options to change the application title, which is displayed at the left of the main tab bar, and the overall application description text, which appears on the **About** tab. Note that changes to text on the **About** tab will take effect the next time the application is launched.

About

The top-level **About** tab displays text describing the data sets currently loaded in LAB-AID (see Fig 13). The texts to be displayed are specified in LAB-AID's JSON configuration file, as described above.

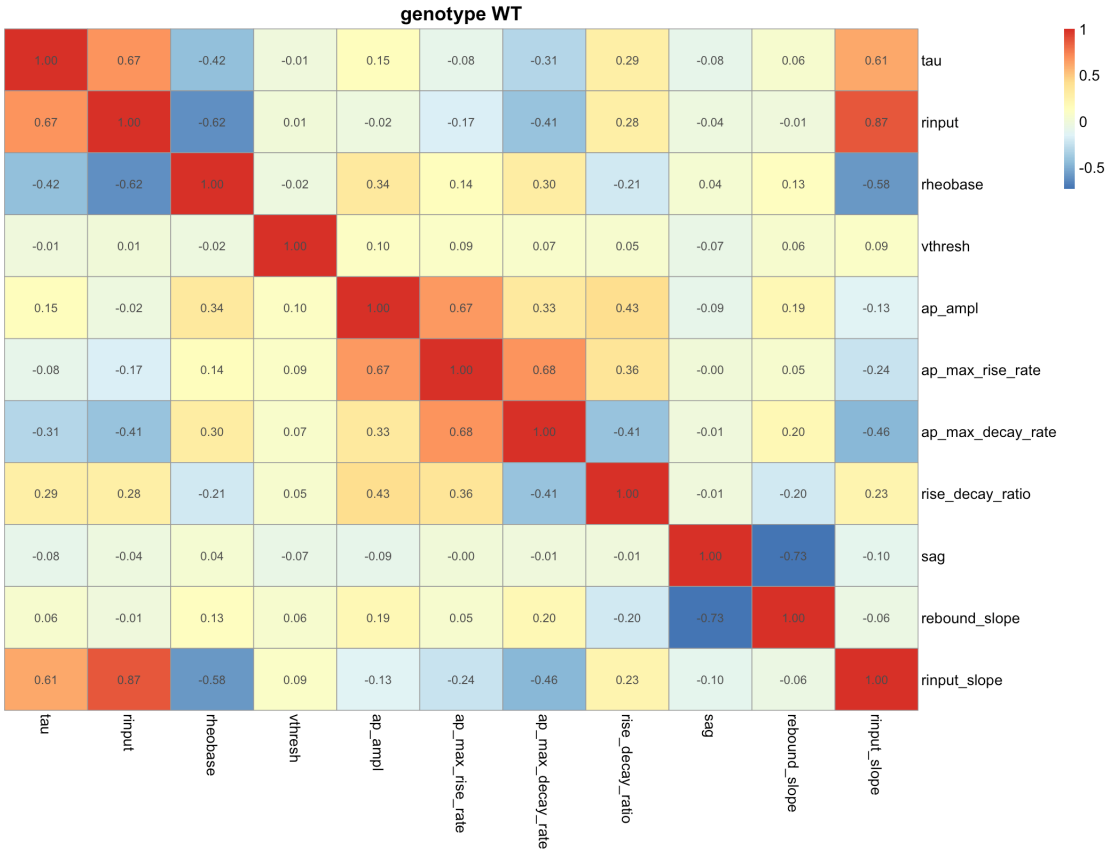

Fig 8. Correlation heat maps show associations between experimental variables. Here correlations between intrinsic electrophysiological properties are shown for neurons originating from wild-type animals. For example, membrane time constant and input resistance are positively correlated, but sag amplitude and rebound slope are strongly anti-correlated.

Use cases

We briefly highlight, in the context of the demonstration electrophysiological data presented here, a number of situations that LAB-AID makes simple to examine:

1. After noting a neuron that is an outlier with respect to one electrophysiological measurement, one can choose that data point – by its unique pID – to be highlighted, and thus immediately see if the neuron is also an outlier with respect to other experimental variables.

2. By highlighting data points originating from particular animals or litters, for example (or those data points which are commonly identified by any other metadata factor), one can examine whether that animal or litter appears to be driving one or more apparent phenotypes.

3. By excluding data points originating from particular animals or litters (or, again, points which are commonly identified by any other metadata factor), one can examine whether correlations between electrophysiological measurements on neurons are robust to the removal of data originating from that animal or litter.

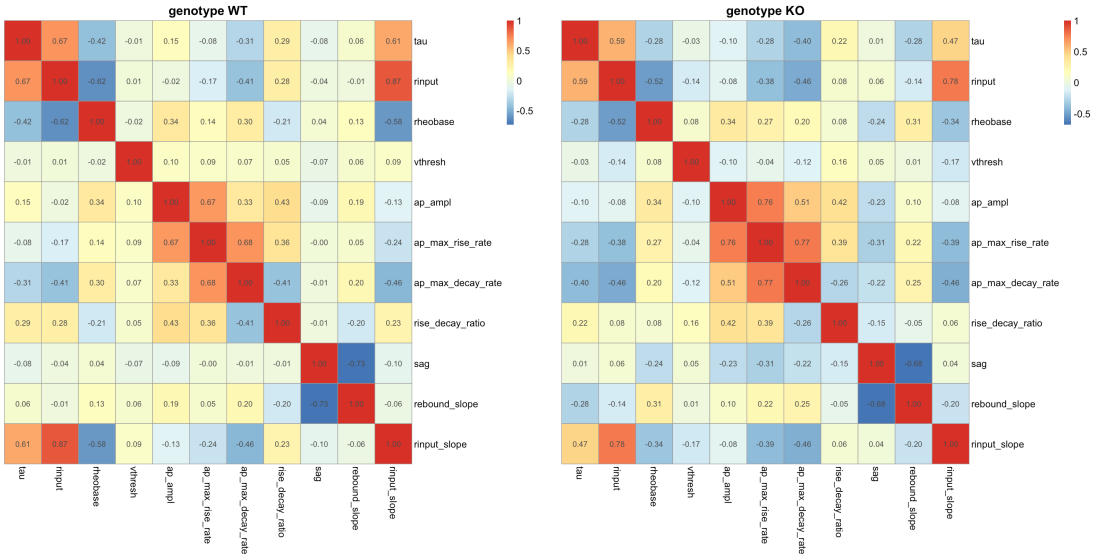

Fig 9. Paired correlation plots allow changes in variable associations to be examined. Here correlations between intrinsic electrophysiological properties are shown (i) for neurons originating from wild-type animals, and (ii) for neurons originating from knockout mice.

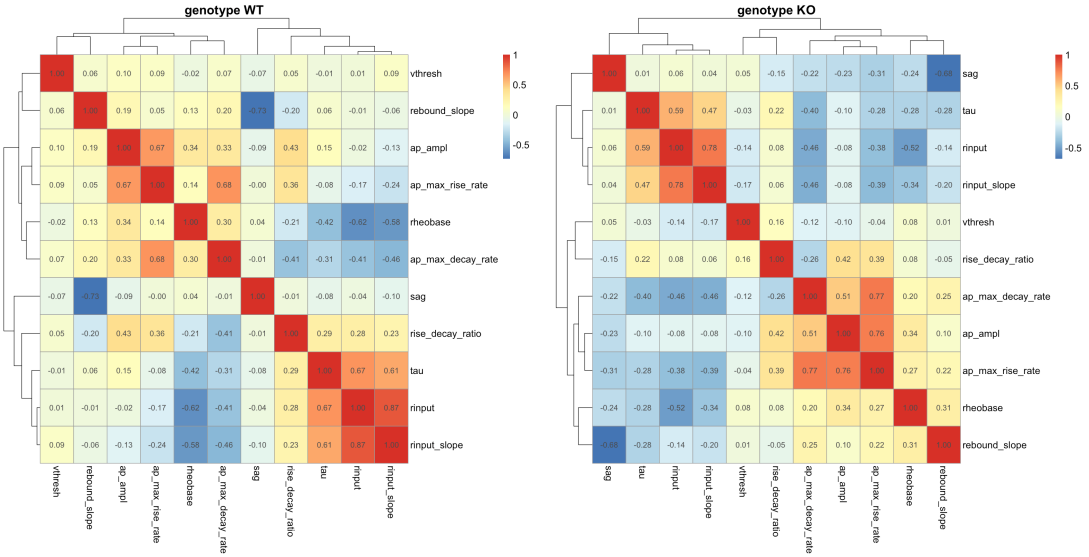

Fig 10. Clustering groups strongly-related experimental variables. Correlations between intrinsic electrophysiological properties for wild-type and knockout neurons are clustered, grouping variables which show similar patterns of correlation.

Conclusion

There is a great demand from biologists to be able to visualize the data from their laboratory experiments. Here we have presented LAB-AID, a simple, yet powerful, tool specifically designed to automatically visualize hierarchical data from biological experiments, that is freely available and easy to install and configure.

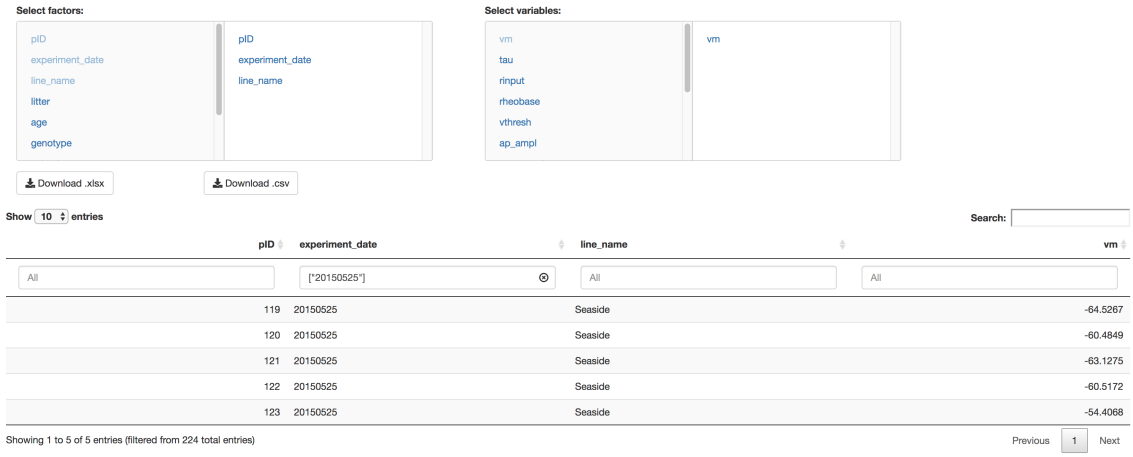

Fig 11. Subsets of the input data can be exported as Excel or CSV files. Here, three metadata columns and one experimental variable column have been selected for export. Filtering has been employed so that only data from a particular experimental date will be output.

References

1. R Core Team. R: A Language and Environment for Statistical Computing; 2018. Available from: `https://www.R-project.org/`.
2. MATLAB. version 9.6 (R2019a); 2019. Available from: `https://mathworks.com/`.
3. Prism. version 8.0.0; 2018. Available from: `https://www.graphpad.com`.
4. Spitzer M, Wildenhain J, Rappsilber J, Tyers M. BoxPlotR: A web tool for generation of box plots. Nature Methods. 2014;11(2):121–122. doi:10.1038/nmeth.2811.
5. Pallmann P, Hothorn LA. Boxplots for grouped and clustered data in toxicology. Archives of Toxicology. 2016;90(7):1631–1638. doi:10.1007/s00204-015-1608-4.
6. Weissgerber TL, Garovic VD, Savic M, Winham SJ, Milic NM. From Static to Interactive: Transforming Data Visualization to Improve Transparency. PLoS Biology. 2016;14(6):1–8. doi:10.1371/journal.pbio.1002484.
7. Weissgerber TL, Savic M, Winham SJ, Stanisavljevic D, Garovic VD, Milic NM. Data visualization, bar naked: A free tool for creating interactive graphics. Journal of Biological Chemistry. 2017;292(50):20592–20598. doi:10.1074/jbc.RA117.000147.
8. Postma M, Goedhart J. Plotsofdata—a web app for visualizing data together with their summaries. PLoS Biology. 2019;17(3):1–8. doi:10.1371/journal.pbio.3000202.
9. Plotly Technologies Inc. Collaborative data science; 2015. Available from: `https://plot.ly`.
10. Mauri M, Elli T, Caviglia G, Uboldi G, Azzi M. RAWGraphs: A Visualisation Platform to Create Open Outputs. In: Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter. CHItaly '17. New York, NY, USA: ACM; 2017. p. 28:1–28:5. Available from: `http://doi.acm.org/10.1145/3125571.3125585`.
11. Chang W, Cheng J, Allaire J, Xie Y, McPherson J. shiny: Web Application Framework for R; 2018. Available from: `https://CRAN.R-project.org/package=shiny`.

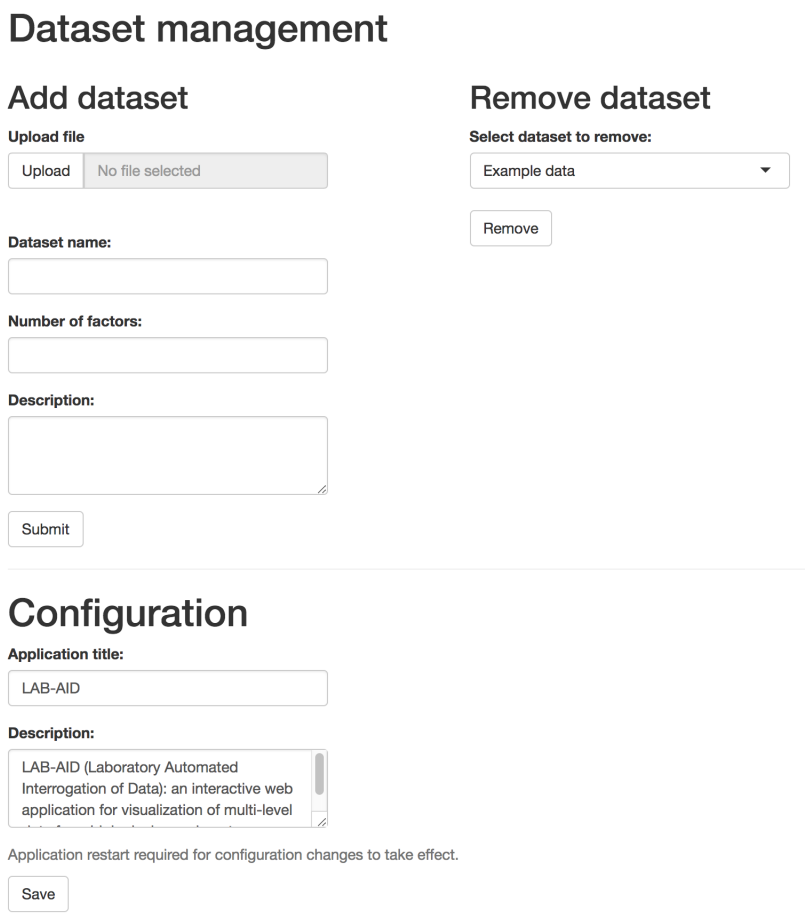

Fig 12. The Configuration tab allows data sets to be added and removed. The application title and description can also be edited.

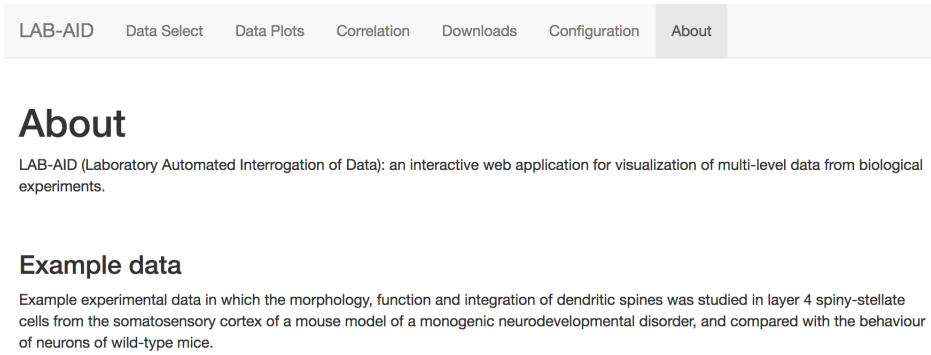

Fig 13. The About tab describes the data sets presented. Both overall and per-data set description texts are defined in LAB-AID's JSON configuration file.

12. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2016. Available from: `http://ggplot2.org`.

13. Kolde R. pheatmap: Pretty Heatmaps; 2018. Available from:

https://CRAN.R-project.org/package=pheatmap.

14. Pedersen TL. patchwork: The Composer of ggplots; 2017. Available from: https://github.com/thomasp85/patchwork.

15. Sievert C. plotly for R; 2018. Available from: https://plotly-book.cpsievert.me.

16. Ooms J. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:14032805 [statCO]. 2014;.

17. Bache SM, Wickham H. magrittr: A Forward-Pipe Operator for R; 2014. Available from: https://CRAN.R-project.org/package=magrittr.

18. Wickham H, Bryan J. readxl: Read Excel Files; 2018. Available from: https://CRAN.R-project.org/package=readxl.

19. Wickham H. Reshaping Data with the reshape Package. Journal of Statistical Software. 2007;21(12):1–20.

20. Attali D. shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds; 2018. Available from: https://CRAN.R-project.org/package=shinyjs.

21. Perrier V, Meyer F, Granjon D. shinyWidgets: Custom Inputs Widgets for Shiny; 2019. Available from: https://CRAN.R-project.org/package=shinyWidgets.

22. Wickham H. tidyverse: Easily Install and Load the 'Tidyverse'; 2017. Available from: https://CRAN.R-project.org/package=tidyverse.

23. Schwartz M. WriteXLS: Cross-Platform Perl Based R Function to Create Excel 2003 (XLS) and Excel 2007 (XLSX) Files; 2015. Available from: https://CRAN.R-project.org/package=WriteXLS.

24. RStudio Team. RStudio: Integrated Development Environment for R; 2018. Available from: http://www.rstudio.com/.