The Journal of Open Source Software

Analysing 24-hour behaviour sequence data with an Rshiny application

Julien Colomb[1, 2] **and York Winter**[1]

1 Humboldt University of Berlin, Inst. of Biology, Philippstr. 13, 10099 Berlin, Germany **2** Humboldt University of Berlin, SFB1315, Inst. of Biology, Charitéplatz 1, 10117 Berlin

DOI: 10.21105/joss.02282

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @mikerspencer
- @aj2duncan

Submitted: 20 May 2020
Published: 01 June 2020

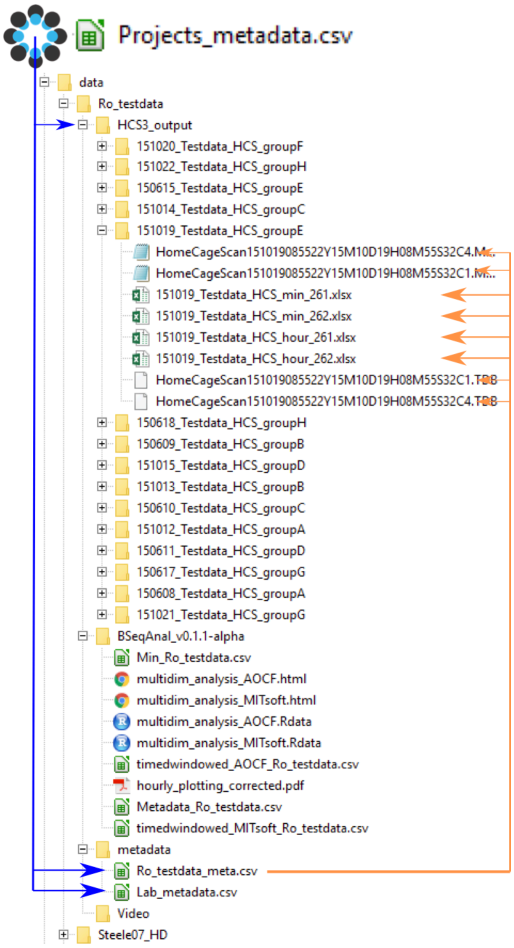
Summary

Automated mouse phenotyping via high throughput behaviour analysis of home cage behaviour has brought hope for a more effective and efficient way to test rodent models of diseases. While different software solutions track behavioural motives through time, software to analyse and archive this rich data is mostly lacking (Steele, Jackson, King, & Lindquist, 2007). Here, we present an open source, free software actionable via a web browser, that can perform state of the art multidimensional analysis of home cage behavioural sequence data. We created an open repository of the linked metadata that we treat as separate from the raw data. Data from a wild type strain of mice used to test the software is provided.

This software should facilitate the analysis of long behavioural sequence data such as extracted by machine learning and other algorithms from video based home cage monitoring.

Data input

In order to facilitate the analysis of data coming from different sources, we propose a format to organise the data (behavior sequence or binned summary data) and the metadata (information about the experiment, the lab and the animals), such that the R-Shiny applications can access the different files automatically. We designed a metadata structure according to metadata schemes developed for research data FAIRness (Martone M., 2014) and considering the needs of the analysis software (Fig. 1). We also made it a prerequisite to publish metadata about the experiment before running the shiny application on the data. Details about the metadata format and a walkthrough in the metadata production is given at: Metadata_information/readme.md. We advise any user to create the metadata during or before data acquisition and provide a folder template.



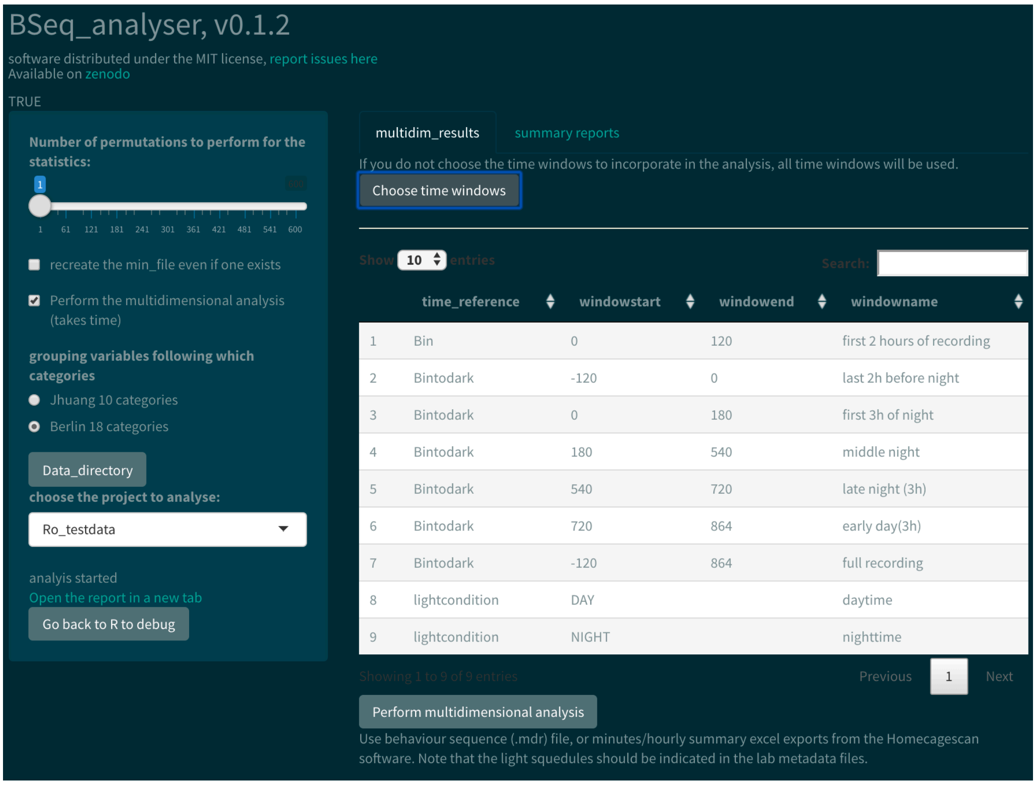
```
Projects_metadata.csv
data
  Ro_testdata
    HCS3_output
      151020_Testdata_HCS_groupF
      151022_Testdata_HCS_groupH
      150615_Testdata_HCS_groupE
      151014_Testdata_HCS_groupC
      151019_Testdata_HCS_groupE
        HomeCageScan151019085522Y15M10D19H08M55S32C4.M...
        HomeCageScan151019085522Y15M10D19H08M55S32C1.M...
        151019_Testdata_HCS_min_261.xlsx
        151019_Testdata_HCS_min_262.xlsx
        151019_Testdata_HCS_hour_261.xlsx
        151019_Testdata_HCS_hour_262.xlsx
        HomeCageScan151019085522Y15M10D19H08M55S32C1.TDB
        HomeCageScan151019085522Y15M10D19H08M55S32C4.TDB
      150618_Testdata_HCS_groupH
      150609_Testdata_HCS_groupB
      151015_Testdata_HCS_groupD
      151013_Testdata_HCS_groupB
      150610_Testdata_HCS_groupC
      151012_Testdata_HCS_groupA
      150611_Testdata_HCS_groupD
      150617_Testdata_HCS_groupG
      150608_Testdata_HCS_groupA
      151021_Testdata_HCS_groupG
    BSeqAnal_v0.1.1-alpha
      Min_Ro_testdata.csv
      multidim_analysis_AOCF.html
      multidim_analysis_MITsoft.html
      multidim_analysis_AOCF.Rdata
      multidim_analysis_MITsoft.Rdata
      timedwindowed_AOCF_Ro_testdata.csv
      hourly_plotting_corrected.pdf
      Metadata_Ro_testdata.csv
      timedwindowed_MITsoft_Ro_testdata.csv
    metadata
      Ro_testdata_meta.csv
      Lab_metadata.csv
    Video
  Steele07_HD
```

Figure 1: Data and metadata structure. The master project_metadata file available online links the address of the metadata files and the data folder (blue arrows). The experiment metadata file links to each data file (for clarity, only one folder is shown here). The format of the data was either .xlsx summary files (with minutes or hour time resolution) or the output files .mbr (behavior sequence) and .tbd (position) of the proprietary HomeCageScan (CleverView) software. Note that the current software did not use the .tbd files. The master file, provided path information to the analysis software. Reports are stored in a folder indicating the software name and version. Derived data files are saved in a folder named after the software name, but not its version.

We have used raw time series of behavior categories produced by the (prioprietary) software HomeCageScan (CleverSys), that had been run on sideview videos of mice placed individually in common lab cages for 22h. The software could be extended to also work with behaviour sequence data from other origin (e.g. using equivalent open source software (Jhuang et al., 2010)). Such application is facilitated since our analysis software only requires the behavioural time series data but not any data summaries.

We used an unpublished dataset based on 11 wild type female mice recorded twice (at the ages of 3 and 7 months, respectively) for about a day, and a published dataset from another study (Luby et al., 2012; Steele et al., 2007). Other datasets were tested but the data is not made public here. For the new dataset, sample size was decided independently of this study and one animal was excluded for lack of data for the second time point. Mice were recorded in the same order at the two time points, and had undergone different behavioural tests before and between the two home cage monitoring events.

Data analysis



BSeq_analyser, v0.1.2

software distributed under the MIT license, report issues here
Available on zenodo

TRUE

Number of permutations to perform for the statistics: 1

□ recreate the min_file even if one exists

☑ Perform the multidimensional analysis (takes time)

grouping variables following which categories

○ Jhuang 10 categories

● Berlin 18 categories

Data_directory

choose the project to analyse: Ro_testdata

analyis started

Open the report in a new tab

Go back to R to debug

multidim_results | summary reports

If you do not choose the time windows to incorporate in the analysis, all time windows will be used.

Choose time windows

Show 10 entries

Search:

	time_reference	windowstart	windowend	windowname
1	Bin	0	120	first 2 hours of recording
2	Bintodark	-120	0	last 2h before night
3	Bintodark	0	180	first 3h of night
4	Bintodark	180	540	middle night
5	Bintodark	540	720	late night (3h)
6	Bintodark	720	864	early day(3h)
7	Bintodark	-120	864	full recording
8	lightcondition	DAY		daytime
9	lightcondition	NIGHT		nighttime

Showing 1 to 9 of 9 entries

Previous 1 Next

Perform multidimensional analysis

Use behaviour sequence (.mdr) file, or minutes/hourly summary excel exports from the Homecagescan software. Note that the light squedules should be indicated in the lab metadata files.

Figure 2: Preview of the shiny GUI. On the left panel, the user chooses variables: project to analyse, behaviour categorisation to use, whether to recreate the minute summary file from the raw data, whether a machine learning analysis should be performed, and the number of permutations to perform (if a machine learning analysis is performed). The user can choose which time windows to incorporate in the analysis. Pushing the "Perform multidimensional analysis button" starts analysis and produces the report. By switching to the summary_reports tab, one can also produce a time series representation of each behaviour category.

Briefly, we merged the 45 categories that were originally generated by the home cage scan software into 10 (Jhuang et al., 2010) or 18 meta-categories (see https://github.com/jcolomb/HCS_analysis/blob/master/analysis/Rcode/grouping_variables.R). The time series data was synchronised to the light off event and split into different time windows, in order to account for circadian rhythm linked effects. The square root of the frequency of each behaviour meta-category (percentage of time spent performing that behaviour) was calculated for each time window. We ended up with 10 to 124 variables per session.

In order to test for differences between different groups of animals, we used a non-parametric test on the first component of a PCA (a p-value and effect size was calculated). As another option, we applied a machine learning algorithm. For this, we used a support vector machine trained on one part of the data to predict potential group differences from the other part of the data (we used a 2-out validation for sample size below 15 per groups). The accuracy of this prediction was then compared to the distribution of accuracies obtained while shuffling the groups randomly (these were computer intensive calculations).

An html report is created from the multidimensional analysis and can be visualised directly in the application. The application can be used directly with one of the test datasets included in the repository (Ro_testdata with minute summary data or Ro_testdata_mbr with sequence raw data).

Putative future development

While the shiny app can fully perform the analysis, we also provide the R code to run the scripts step by step. In addition to facilitating code debugging, this allows users to include additional analysis steps (e.g. paired analysis) or change figures. We also provide additional analysis for visualisation and a more complex analysis using the raw (sequence) data. In particular, one script analyses the behaviour sequence itself, reporting the percentage of time a behaviour was performed just before another one (Fig. 3 shows the eight behaviours occurring just before or after a "land vertical" behaviour in the test data).

A server version of the application was not yet possible, because the application needs to access folders (with ShinyFiles), something browsers are not allowed to do. One could develop a different version of the application that would upload (from the computer) or copy the data (from another server) before the analysis is done. This might facilitate metadata analyses combining data from different labs (but linked in the single master metadata file), in the future.

Conclusion

The software presented here demonstrates the power of combining data management with analysis to process data more efficiently and effectively. This approach avoids most pitfalls of multivariate analysis such as p-hacking and harking, as well as human errors in data processing.

By making this software available and linking it to re-usable (FAIR) open datasets, we hope to initiate community activity for tools facilitating long-term animal behaviour sequence analysis.

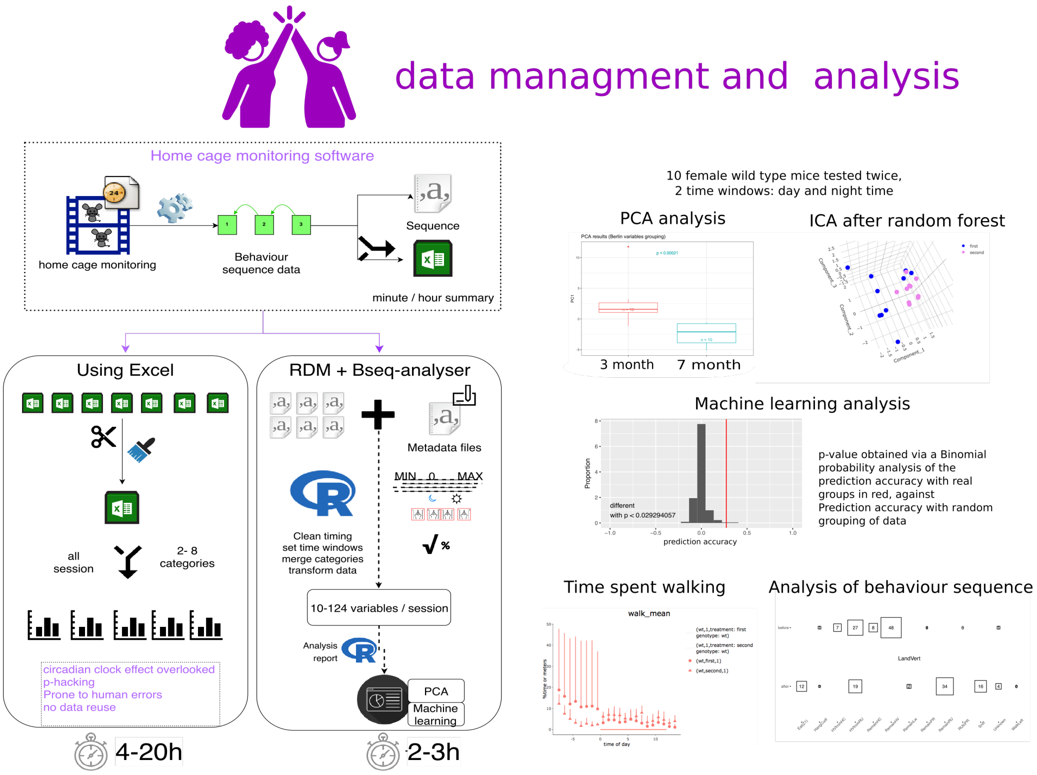


Figure 3: Visual abstract. Left: HomeCageScan software (CleverSys) analyses video data to produce a time series of behaviour categories, as well as spreadsheets with pre-analysis data. Previous studies used the summary data to perform summary analysis. Our software used the raw behaviour sequence data, as well as metadata spreadsheets the user had to provide. R code was used to synchronise the time series and cut datasets to a common length for all sessions, to merge categories, and to produce summaries for several time windows of observation. The summary data was then analysed and a report was saved on disc. The process takes about 3 hours (mostly used to create the metadata files), and uses multivariate analysis. Right: example of analysis output using a dataset from wild type mice recorded twice. A PCA analysis could tell the two groups apart, while the machine learning algorithm we used (SVM) had difficulties to do so. Hourly summaries for each behaviour category can also be visualised in the application. More complex analysis might be performed from the same data (see text).

Acknowledgements

The authors want to thank members of the Winter lab and the NeuroCure Animal Outcome Core Faciltiy team: Andrei Istudor for his suggestion to integrate all used variables in the software report, Patrick Beye for discussions on the analysis design, Melissa Long for performing the home cage monitoring experiment and help with the creation of the metadata, and Vladislav Nachev for scientific inputs and discussion. In addition, we want to thank Prof. Andrew D. Steele and Prof. Oliver D. King for fruitful discussion and access to code and data, and the team of Cleversys Inc. for their help with the decoding of the HomeCageScan software outputs.

Funding

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 327654276 – SFB 1315, and Project number 39052203 - EXC 257: NeuroCure.

Dependencies

The software was buildt on R ressources (R Core Team, 2019). This work would not have been possible without the tidyverse environment (Wickham, 2017a, 2017b), packages for interactive processing (Chang, Cheng, Allaire, Xie, & McPherson, 2017; Pedersen, 2016; Sievert et al., 2017), statistical analysis (Harrell Jr, Charles Dupont, & others., 2017; Helwig, 2015; Hothorn, Winell, Hornik, van de Wiel, & Zeileis, 2019; Kassambara, 2019; Liaw & Wiener, 2002; Meyer, Dimitriadou, Hornik, Weingessel, & Leisch, 2017; Park & Hastie, 2018) and graphical interface (Auguie, 2017; Murrell, 2016; Sievert et al., 2017). It also depended on the osfr package, which was still in development (Hartgerink, Nagraj, Hafen, & Colomb, 2017) and loaded via the devtools package (Wickham & Chang, 2017). We used the packrat package (Ushey, McPherson, Cheng, Atkins, & Allaire, n.d.) to dock the project.

References

Auguie, B. (2017). *GridExtra: Miscellaneous functions for "grid" graphics*. Retrieved from https://CRAN.R-project.org/package=gridExtra

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2017). *Shiny: Web application framework for r*. Retrieved from https://CRAN.R-project.org/package=shiny

Harrell Jr, F. E., Charles Dupont, & others. (2017). *Hmisc: Harrell miscellaneous*. Retrieved from https://CRAN.R-project.org/package=Hmisc

Hartgerink, C., Nagraj, V., Hafen, R., & Colomb, J. (2017). *Osfr: API to the open science framework*.

Helwig, N. E. (2015). *Ica: Independent component analysis*. Retrieved from https://CRAN.R-project.org/package=ica

Hothorn, T., Winell, H., Hornik, K., van de Wiel, M. A., & Zeileis, A. (2019). *Coin: Conditional inference procedures in a permutation test framework*. Retrieved from https://CRAN.R-project.org/package=coin

Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A. D., & Serre, T. (2010). Automated home-cage behavioural phenotyping of mice. *Nature Communications*, *1*(6), 1–9. doi:10.1038/ncomms1064

Kassambara, A. (2019). *Rstatix: Pipe-friendly framework for basic statistical tests*. Retrieved from https://CRAN.R-project.org/package=rstatix

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. Retrieved from http://CRAN.R-project.org/doc/Rnews/

Luby, M. D., Hsu, C. T., Shuster, S. A., Gallardo, C. M., Mistlberger, R. E., King, O. D., & Steele, A. D. (2012). Food Anticipatory Activity Behavior of Mice across a Wide Range of Circadian and Non-Circadian Intervals. (P. A. Bartell, Ed.)*PLoS ONE*, *7*(5), e37992. doi:10.1371/journal.pone.0037992

Martone M. (2014). *Data Citation Synthesis Group: Joint Declaration of Data Citation Principles*. Retrieved from https://www.force11.org/group/joint-declaration-data-citation-principles-final

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2017). *E1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien*. Retrieved from https://CRAN.R-project.org/package=e1071

Murrell, P. (2016). *RGraphics: Data and functions from the book r graphics, second edition*. Retrieved from https://CRAN.R-project.org/package=RGraphics

Park, M. Y., & Hastie, T. (2018). *Glmpath: L1 regularization path for generalized linear models and cox proportional hazards model*. Retrieved from https://CRAN.R-project.org/package=glmpath

Pedersen, T. L. (2016). *ShinyFiles: A server-side file system viewer for shiny*. Retrieved from https://CRAN.R-project.org/package=shinyFiles

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., & Despouy, P. (2017). *Plotly: Create interactive web graphics via 'plotly.js'*. Retrieved from https://CRAN.R-project.org/package=plotly

Steele, A. D., Jackson, W. S., King, O. D., & Lindquist, S. (2007). The power of automated high-resolution behavior analysis revealed by its application to mouse models of Huntington's and prion diseases. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(6), 1983–8. doi:10.1073/pnas.0610779104

Ushey, K., McPherson, J., Cheng, J., Atkins, A., & Allaire, J. (n.d.). *Packrat: A dependency management system for projects and their r package dependencies*. Retrieved from https://github.com/rstudio/packrat/

Wickham, H. (2017a). *Tidyverse: Easily install and load the 'tidyverse'*. Retrieved from https://CRAN.R-project.org/package=tidyverse

Wickham, H. (2017b). *Stringr: Simple, consistent wrappers for common string operations*. Retrieved from https://CRAN.R-project.org/package=stringr

Wickham, H., & Chang, W. (2017). *Devtools: Tools to make developing r packages easier*. Retrieved from https://CRAN.R-project.org/package=devtools