Rethomics: an R framework to analyse high-throughput behavioural data

Quentin Geissmann[1*], Luis G Rodriguez[2], Esteban J Beckwith[1], Giorgio F Gilestro[1*]

1 Department of Life Sciences, Imperial College London, London, United Kingdom
2 Institute for Neuro- and Behavioral Biology, Westfälische Wilhelms University, 48149 Münster, Germany

* qgeissmann@gmail.com, giorgio@gilest.ro

Abstract

Ethomics, a quantitative and high-throughput approach to animal behaviour, is a new and exciting field. The recent development of automatised methods that can score various behaviours on a large number of animals provides biologists with an unprecedented set of tools to decipher these complex phenotypes. Analysing ethomics data comes with many challenges that are largely shared across acquisition platform and paradigms. However, there is little effort in providing a generic framework to specifically analyse multiple and long behavioural time series. We developed the `rethomics` framework, a suite of `R` packages that altogether offers utilities to: import, store, analyse and visualise behavioural data. In this article, we describe it and show an example of its application to the study of circadian rhythms and blooming field of sleep fruit flies. The `rethomics` framework is available and documented at https://rethomics.github.io.

Todo list

Introduction

Citation Needed:

The behaviour of an animal is a complex phenotypical manifestation of the interaction between its nervous system and external or internal environments. In the last few decades, our ability to record vast quantities of various phenotypical data has tremendously increased. The scoring of behaviours is certainly not an exception to this trend. Indeed, many platforms () have been developed in order to allow biologists to continuously record behaviours such as activity, position and feeding of multiple animals over long durations (days or weeks).

The availability of large amounts of data is very exciting as it paves the way for in-depth quantitative analyses. The multiplicity of model organisms, hypotheses and paradigms makes the existence of a diverse range of recording tools important. However, when it comes to the subsequent processing of the results, there is no unified, programmatic, framework that could be used as a set of building blocks in a pipeline.

Instead, tools tend to consist of graphical interfaces that import data from a single platform and provide somewhat rigid functionalities. There are, at least, three issues with this approach. First of all, state-of-the-art analysis and visualisation require a level of reproducibility, flexibility and scalability that only a programming interface can provide. Secondly, it favours replicated work as developers need to create their independent solution to similar problems. Lastly, it links analysis and visualisation to the target acquisition tool, which makes it very difficult to share cross-tool utilities and concepts.

Thankfully, behavioural data is conceptually largely agnostic of the acquisition platform and paradigm. Typically, the behaviour of each individual is described by a long time series (possibly multivariate and irregular). Importantly, individuals are labelled with arbitrary metadata defined by the experimenter (*e.g.* sex, treatment and genotype). Efficiently combining and manipulating metadata and data from hundreds of individuals, each recorded for days or weeks, is not trivial.

In the article herein, we describe `rethomics`, a framework that unifies analysis of behavioural dataset in an efficient and flexible manner. We implemented it in `R` [1] since it is widely taught and adopted by computational biologists. It is also complemented with a vast ecosystem of open-source packages. `rethomics` offers an elegant computational solution to store, manipulate and visualise large amounts of data. We expect it to fill the gap between behavioural biology and data sciences, thus promoting collaboration between computational and behavioural scientists. `rethomics` comes with a extensive documentation and a set of both practical and theoretical demos and tutorials.

Design and Implementation

`rethomics` is implemented as a collection of targeted `R` packages related to one another (Fig 1). This paradigm follows the model of modern frameworks such as the `tidyverse` [2], which results in increased testability and maintainability. In it, the different tasks of the analysis workflow (*i.e.* data import, manipulation and visualisation) are explicitly handled by different packages. At the core of `rethomics`, the `behavr` package offers a very flexible and efficient solution to store large amounts data (*e.g.* position and activity) as well as metadata (*e.g.* treatmentand genotype) in a single `data.table`-derived object [3]. Any input package will import experimental data as a `behavr` table which can, in turn, be analysed and visualised regardless of the original input platform. Results and plots integrate seamlessly into the `R` ecosystem, hence providing users with state-of-the-art visualisation and statistical tools.

Fig 1. The `rethomics` workflow. Diagram representing the interplay between, from left to right, the raw data, the `rethomics` packages (in blue) and the rest of the `R` ecosystem.

Internal data structure

We created `behavr` (Fig 2A), a new data structure, based on the widely adopted `data.table` object, in order to address two challenges that are inherent to handling ethomics results.

Firstly, there could be very long (typically $k_i > 10^8, \forall i \in [1, n]$), multivariate (often, $q > 10$), time series for each individual. For instance, each series could represent variables that encode coordinates, orientation, dimensions, activity, colour intensity and so on, sampled several times per second, over multiple days. Therefore, the data structure must be computationally efficient – both in term of memory footprint and processing speed.

Secondly, a large number of individuals are often studied (typically $n > 100$). Each individual (i) is associated with metadata: a set of p "metavariables" that describe experimental conditions. For instance, metadata stores information regarding the date and location of the experiment, treatment, genotype, sex, *post hoc* observations and other arbitrary metavariables. It is interesting to record multiple metavariables in so far as they so they can later be used as covariates. Therefore, typically $p > 10$.

`behavr` tables link metadata and data within the same object, extending the syntax of `data.table` to manipulate, join and access metadata (Fig 2B). This approach guarantees that any data point can be mapped correctly to its parent metadata. It also allows implicit update of metadata when data is altered. For instance, when is data filtered, only the remaining individuals should be in the new metadata. It is also important that metadata and data can interoperate. For instance, when one wants to update variable according to the value of a metavariable (say, alter the variable x only for animals with the metavariable sex = "*male*"). The online tutorials and documentation provide a detailed set of examples and concrete use cases of `behavr`.

Fig 2. `behavr` table. A: Illustration of a `behavr` object, the core data structure in `rethomics`. The metadata holds a single row for each of the n individuals. Its columns, the p metavariables, are one of two kinds: either required – and defined by the acquisition platform (*i.e.* used to fetch the data) – or user-defined (*i.e.* arbitrary). In the data, each row is a "read" (*i.e.* information about one individual at one time-point). It is formed of q variables and is expected to have a very large number of reads, k, for each individual i. Data and metadata are implicitly joined on the `id` field. Note that the names used in this for variables and metavariable in this example are only plausible cases which will likely differ in practice. B: Non exhaustive list of uses of a `behavr` table (referred as `dt`). In addition to operations on data, which are inherited from `data.table`, we provide utilities designed specifically to act on both metadata and data. Commented examples are prefixed by `>`.

Data import

Citation Needed:

Data import package translate results from a recording platform (*e.g.* text files and databases) into a single `behavr` object. Currently, we provide a package to read single or multi-beam Drosophila Activity Monitor (DAM)data and another one for Ethoscope data [4]. Although the structure of the raw results is very different, conceptually, loading data is very similar. In all cases, the user is asked to generate a metadata table (one row per individual). In it, there will be both mandatory and optional columns. The mandatory ones are the necessary and sufficient information to fetch data (*e.g.* machine id, region of interest and date). The optional columns are user-defined arbitrary fields that translate experimental conditions (*e.g.* treatment, genotype and sex).

In this respect, the metadata file is a standardised and comprehensive data frame describing an experiment. It explicitly lists all treatments and individuals, which facilitates interspersion of conditions (indeed, without it, users are tempted to simplify their experimental design by, for instance, confounding device/location and treatment). Furthermore, it streamlines the inclusion and analysis of further replicates in the same workflow. Indeed, additional replicates can simply be added as new rows – and the ID of the replicate later used, if needed, as a covariate.

Visualisation

Long time series often need to be preprocessed before visualisation. Typically, users are interested in understanding individual or population trends over time. To integrate visualisation in `rethomics`, we implemented `ggetho`, a package that offers new tools that extent the widely adopted `ggplot2` [5]. It provides preprocessing utilities as well as new layers and scales. Our tools make full use of the internal `behavr` structure to deliver efficient representations of temporal trends. It particularly applies to the visualisation of long experiments, with the ability to, for instance, display "double-plotted actograms", periodograms, annotate light and dark phases and wrap time over a given period. Importantly, `ggetho` is fully compatible with `ggplot2`.

Circadian and sleep analysis

The packages `zeitgebr` and `sleepr` provide tools to analyse circadian behaviours and sleep, respectively. Together, they offer a suite of methods to compute autocorrelogram, χ^2 [6] and Lomb-Scargle [7] periodograms and find their peaks, score sleep from inactivity (*i.e.* using the "five-minute rule"), and characterise the architecture of sleep bouts (*e.g.* number, length and latency).

Results

TODO: @esteban rewrite this please ;)

In order to illustrate the workflow of `rethomics`, we decided to provide an example that is simple and reproducible, but which could be readily modified for more complicated cases. DAM2 being widely adopted, and fruit flies being a model of choice for high throughput studies, we decided to show how `rethomics` can be used to study circadian rhythm in fruit flies, using this type data.

We gathered a subset of the data from a recent publication [8], which was gracefully made publicly available by the authors [9]. It shows that flies, which normally become arrhythmic in a Light-Light (LL) regime, are rhythmic (with an period greater than 24h) when the gene NKCCox is expressed in clock neurons. The authors have recorded the activity of both TIM/NKCCOK (NKCCox is expressed in their clock neurons) and NKCCOK/+ (the appropriate control) in LL for six days. Prior to LL, the animals were recorded for a baseline three or four days in normal LD, 12h:12h, conditions. Two repetitions were performed resulting in a total of 58 animals. The `metadata.csv` file, which describes exhaustively all individuals, as well as all the associated result files can be downloaded at https://zenodo.org/record/1172980.

TODO: phrasing

As a prerequisite, we downloaded the data, extracted the zip archive in our working directory. We start the analysis by loading the necessary `rethomics` packages (see availability section for installation instructions):

```r
library(damr)       # input DAM2 data
library(zeitgebr)   # periodogram computation
library(sleepr)     # sleep analysis
library(ggetho)     # behaviour visualisation
```

Then, the metadata file is read and linked to the `.txt` result files.

```r
metadata <- link_dam2_metadata("metadata.csv", ".")    # linking
```

```
# print(metadata)                                          # check metadata
dt <- load_dam2(metadata)                                  # loading
summary(dt)                                                # quick summary

## behavr table with:
##  58 individuals
##  8 metavariables
##  2 variables
##  1.58722e+05 measurements
##  1 key (id)
```

Preprocessing

The two replicates do not have the same baseline time and we would like to express the time relative to the important event: the transition from LD to LL. Therefore, we subtract the `baseline_days` metavariable from the `t` variable. This gives us an opportunity to illustrate the use `xmv()`, which expands metavariables as variables. In addition, we use the `data.table` syntax to create, in place, a `moving` variable. It is `TRUE` when and only when `activity` is greater than zero:

```
# baseline subtraction -- note the use of xmv
dt[,t := t - days(xmv(baseline_days))]
dt[, moving :=  activity > 0]
```

```
summary(dt)

## behavr table with:
##  58 individuals
##  8 metavariables
##  3 variables
##  1.58722e+05 measurements
##  1 key (id)
```

To simplify visualisation, we create our own `label` metavariable, as the combination of a number and `genotype`. In the restricted context of this analysis, `label` acts as a unique identifier. Importantly, we keep `id`, which is more rigorous and universal.

```
dt[, label := interaction(1:.N, genotype), meta = T]
print(dt)
```

Curation

It is important to visualise an overview of how each individual behaved and, if necessary, alter the data accordingly. For this, we generate a tile plot (Fig 3A).

```
# make a ggplot object with label on the y and moving on the z axis
```

Fig 3. Experiment quality control. Tile plot showing the fraction of time spent moving as a colour intensity. Each individual is represented by a row and time, on the x-axis, is binned in 30 minutes. A: Uncurated raw data. B: Data after the curation step. Time was trimmed and data from dead animals removed. Red '+' symbols show animals that were removed from the subsequent analysis as they had less than five complete days in LL.

```
fig3A <- ggetho(dt, aes(y = label, z = moving)) +
  # show data as a tile plot
  # that is z is a pixel whose intensity maps moving
  stat_tile_etho() +
  # add layers to draw annotations to show L and D phases
  # as white and black, respectively
  # the first layer is for the baseline (until t = 0)
  stat_ld_annotations(x_limits = c(dt[,min(t)], 0)) +
  # in the 2nd one, we start at 0 and use grey
  # instead of black as we work in LL
  stat_ld_annotations(x_limits = c(0, dt[, max(t)]),
                      ld_colours = c("white", "grey"))
```

The activity of dead or escaped animals is falsely scored as long series of zeros (see, for instance, individual 30 and 18 in Fig 3A). Our `sleepr` package offers a tool to detect and remove such artefactual data.

```
# remove data after death
dt <- sleepr::curate_dead_animals(dt, moving)
```

In addition, we can trim our data to have the same number of days across experiments and individuals.

```
# filter dt between -2d and 6d
dt <- dt[t %between% days(c(-2, 6))]
# same as above
fig3B <- ggetho(dt, aes(y = label, z = moving)) +
    stat_tile_etho() +
    stat_ld_annotations(x_limits = c(dt[, min(t)], 0)) +
    stat_ld_annotations(x_limits = c(0, dt[, max(t)]),
                        ld_colours = c("white", "grey"))
```

For the purpose of this example, we also exclude animals that died prematurely, and keep only individuals that have *at least five days in LL*. An overview of the curate data can be visualised in Fig 3B.

```
# for each id, we check for validity
```

```
valid_dt <- dt[, .(valid = max(t) > days(5)), by = id]
# a vector of all valid ids
valid_ids <- valid_dt[valid == T, id]
# filter dt with the valid ids
dt <- dt[id %in% valid_ids]
summary(dt)

## behavr table with:
##  52 individuals
##  9 metavariables
##  3 variables
##  1.19546e+05 measurements
##  1 key (id)
```

Note that as a result, we now have 52 “valid”individuals.

Double-plotted actograms

“Double-plotted actograms” are a common visualisation of periodicity and rhythmicity in circadian experiments. In S1 FigA, we show the double-plotted actograms of each animal. A representative sample of four individuals for each genotype is shown in Fig 4A.

```
# we also show a subset of this figure in 4A
figS1A <- ggetho(dt, aes(z = moving), multiplot = 2) +
            # bars n the z axis could
            # one could also use stat_tile_etho
            stat_bar_tile_etho() +
            # split plot by individual
            facet_wrap( ~ label, ncol = 4) +
            # rename the y axis
            scale_y_discrete(name = "Day")
```

Periodograms

Ultimately, in order to quantify periodicity and rhythmicity, we compute periodograms. Several methods are implemented in `zeitbebr` (χ^2, Lomb-Scargle and autocorrelation). In this example, we generate χ^2 periodograms and lay them out in a grid. Periodograms for the subset of eight animals used in Fig 4A is shown in Fig 4B. See S1 FigB for the visualisation of all individuals.

```
# only the LL data
```

```
dt_ll <- dt[t > days(1)]
# compute chi sqr periodogram
per_dt <- periodogram(moving,
                        dt_ll,
                        resample_rate = 1 / mins(10),
                        FUN=chi_sq_periodogram)

per_dt <- find_peaks(per_dt)
# we also show a subset of this figure in supplementary materials
figS1B <- ggperio(per_dt, aes(y = power, peak = peak)) +
                  # periododogram drawn as a line
                  geom_line() +
                  # the significance line in red
                  geom_line(aes(y = signif_threshold), colour = "red") +
                  # point and text at the peak
                  geom_peak() +
                  # divide plot by individual
                  facet_wrap( ~ label, ncol = 4)
```

Fig 4. Visualisation of the periodicity in activity of eight representative animals. A: Double-plotted actograms showing activity over the experiment. Transition from LD to LL happens at day 0. B: χ^2 periodograms over the LL part of the experiment matching the animals in A. The blue cross represents the first peak (if present) above the significance threshold (red line). Titles on top of each facet refer to the label allocated to each individual. See S1 Fig for all 52 animals.

Population statistics

As shown in the original study, double-plotted actograms and periodograms suggest that NKCCOX/+ flies are mostly arhythmic in LL whilst Tim/NKCCOX appear to have a consistent, long-period rhythm. To visualise this at the population scale, we can plot an average periodogram (see Fig 5A):

```
# display periodogram
fig5A <- ggperio(per_dt, aes(y = power - signif_threshold,
                             colour = genotype)) +
          # periododogram shown as a line for population mean
          # and bootstrap error bars
          stat_pop_etho(method = ggplot2::mean_cl_boot) +
          # rename x and y axis
          scale_y_continuous(name = "Relative power") +
          scale_x_hours("Period")
```

To further quantify this difference, we can show the number of rhythmic animals – *i.e.* individuals for which a peak was found – in each group (see Fig 5B). Then, we can compare the average value of the peak for the rhythmic animals (see Fig 5C). First of all, we compute a summary per individual (`by=id`):

Fig 5. Population statistics on circadian phenotype. A: Average periodograms. The aggregated relative power of the periodogram of all animals. The solid lines and the shaded areas show population means and their 95% bootstrap confidence interval, respectively. B: Frequencies of rhythmic animals. Number of rhythmic animals (*i.e.* with a significant peak) in each genotypes. Dark and clear fillings indicate rhythmic and arhythmic animals, respectively. C: Peak periodicity power and average. Values of the peak period for animals with a significant peak (*i.e.* rhythmic). Individual animals are shown by dots whose size represent relative power of the peak period. The error bars are 95% bootstrap confidence interval on the population mean.

```r
summary_dt <-
    per_dt[,
          .(
           first_peak_period = period[peak == 1],
           { # {} can be used for tmp variables
             signif = signif_threshold[peak == 1]
             power = power[peak == 1]
             first_peak_rel_power =  power - signif
            },
           is_rhythmic = any(peak == 1)
           ),
           by=id]

# rejoin metadata
summary_dt <- rejoin(summary_dt)
```

`summary_dt` is just a regular data frame with one row per individual, containing both metadata and our summary statistics. It can therefore be used directly by `ggplot` and other tools: 170 171 172

```r
# standard ggplot
fig5B <- ggplot(summary_dt, aes(x = genotype,
                                fill = genotype,
                                alpha = is_rhythmic
                                )) +
              geom_bar(colour="black")
```

```r
# standard ggplot
fig5C <- ggplot(summary_dt, aes(y = first_peak_period,
                                x = genotype)) +
              # draw the mean of each genotype group
              stat_summary(fun.y = mean, geom = "point", shape=3) +
              # draw bootstrap confidence intervals
              stat_summary(fun.data = mean_cl_boot, geom = "errorbar") +
              # shows all individuals as points
              # the size of the point expresses the power of the peak
              geom_jitter(aes(colour = genotype,
                              size = first_peak_rel_power),
                              alpha = 0.67) +
              # We would like to convert time in hour
              scale_y_hours("Period")
```

R provides one of the richest statistical toolbox available, which allows users to go deeper in the analysis of the extracted variables. One could, for instance, perform a χ^2 test on the number of rhythmic *vs* arhythmic flies in both genotypes. To address the same question, we fit a binomial generalised linear model:

```
fit <- glm(is_rhythmic ~ genotype, summary_dt, family = "binomial")

summary(fit)$coefficients

##                      Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)         -1.504077  0.5527708 -2.720978 6.508902e-03
## genotypeTim/NKCCOX   4.143135  0.9172057  4.517127 6.268433e-06
```

The result shows a strong positive effect of genotype Tim/NKCCOX on the probability of being rhythmic (p-value 6.27×10^{-06}):

Lastly, we can generate a table that compute arbitrary population statistics for each genotype:

```
result_dt <- summary_dt[,
          .(
            mean_period = mean(first_peak_period, na.rm = T) / hours(1),
            sd_period = sd(first_peak_period, na.rm = T) / hours(1),
            n_rhythmic = sum(is_rhythmic),
            n = .N,
            percent_rhythmic = 100 * sum(is_rhythmic) / .N
            ),
          by = genotype
          ]
result_dt

##       genotype mean_period sd_period n_rhythmic  n percent_rhythmic
## 1:   NKCCOX/+    25.22500  3.598495          4 22         18.18182
## 2: Tim/NKCCOX    26.21429  2.363568         28 30         93.33333
```

This example shows how `rethomics` can be used from the raw data to making publication-quality figures and statistics. We were able to comprehensively analyse the data from a circadian experiment with a few line of code. This workflow applies particularly to much larger data sets and provides a large degree of flexibility that will allow biologists to tune their analysis to their specific questions.

```
## Error in FUN(X[[i]], ...):  object 'first_peak_rel_power' not found
## Error in FUN(X[[i]], ...):  object 'first_peak_rel_power' not found
## Error in cowplot::plot_grid(fig5, legend, nrow = 2, rel_heights =
c(15, :  object 'fig5' not found
## Error in print(fig5_final):  object 'fig5_final' not found
```

Availability and Future Directions

All packages in the `rethomics` framework are available under the terms of the GPLv3 license and listed at https://github.com/rethomics/. Extensive installation instructions as well as reproducible demos and tutorials are available at https://rethomics.github.io/.

All packages are continuously integrated and unit tested on several version of `R` to minimise the risk of present and future issues.
 * How we have already used rethomics in two impactful papers
 * Other inputs
 * Position analysis
 * GUI
 * ...

Supporting information

S1 Fig. Complete version of Fig 4. See Fig 4 for legend.

Acknowledgements

TODO: Thank people here

Han Kim
Maite
Hannah, Alice, Diana
Marcus Gosh, Patrick Krätschmer Rob

References

1. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing;. Available from: `https://www.R-project.org/`.
2. Wickham H. tidyverse: Easily Install and Load the 'Tidyverse';. Available from: `https://CRAN.R-project.org/package=tidyverse`.
3. Dowle M, Srinivasan A. data.table: Extension of 'data.frame';. Available from: `https://CRAN.R-project.org/package=data.table`.
4. Geissmann Q, Rodriguez LG, Beckwith EJ, French AS, Jamasb AR, Gilestro GF. Ethoscopes: An open platform for high-throughput ethomics. PLOS Biology;15(10):e2003026. doi:10.1371/journal.pbio.2003026.
5. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer;.
6. Sokolove PG, Bushell WN. The chi square periodogram: Its utility for analysis of circadian rhythms. Journal of Theoretical Biology;72(1):131–160. doi:10.1016/0022-5193(78)90022-X.
7. Ruf T. The Lomb-Scargle Periodogram in Biological Rhythm Research: Analysis of Incomplete and Unequally Spaced Time-Series. Biological Rhythm Research;30(2):178–201. doi:10.1076/brhm.30.2.178.1422.
8. Buhl E, Bradlaugh A, Ogueta M, Chen KF, Stanewsky R, Hodge JJL. Quasimodo mediates daily and acute light effects on Drosophila clock neuron excitability. Proceedings of the National Academy of Sciences;113(47):13486–13491. doi:10.1073/pnas.1606547113.
9. Ogueta M, Stanewsky R. LL Behaviour of TIM Gal4 > NKCC OX and NKCC OX / + flies;. Available from: `https://zenodo.org/record/1172980`.