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

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 such data comes with several challenges that are largely shared across acquisition platform and paradigms. In the article herein, we present `rethomics`, a set of `R` packages that unifies analysis of behavioural dataset in an efficient and flexible manner. `rethomics` offers a computational solution to store, manipulate and visualise large amounts of behavioural data. We propose it as a tool to breach the gap between behavioural biology and data sciences, thus connecting computational and behavioural scientists. Our software comes with a extensive documentation as well as a set of both practical and theoretical tutorials (available at https://rethomics.github.io).

Todo list

TODO: revert line numbers before submission

Introduction

The behaviour of an animal is a complex phenotypical manifestation of the interaction between its nervous system, its internal state and its environment. In the last few decades, our ability to record vast quantities of various phenotypical data has tremendously increased [1]. The scoring of behaviours is certainly not an exception to this trend [2]. Indeed, many platforms have been developed in order to allow biologists to continuously record behaviours such as activity [3], position [4] and feeding [5,6] of single or multiple [7,8] animals over long durations (days or weeks).

The availability of large amounts of data paves the way for in-depth quantitative analyses which, in turn, leads to the characterisation of new principles and ultimately a better understanding of their underlying biology [9,10]. The multiplicity of model organisms, hypotheses and paradigms justifies the existence of a diverse palette of recording techniques 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.

Scripting interfaces are the standard in data sciences and statistics since they help delivering reproducible results [11, 12]. In addition, they can be used on remote resources such as computer clusters, which makes them more scalable in the context of 'big data' [13]. The field of bioinformatics is a good example of a community that has taken advantage of sharing standard files formats, modular command line tools [14] and software packages [15] that can be assembled to build pipelines [16]. Since many aspect of behaviour analysis are also becoming increasingly linked to data sciences, the development of such common tools and data structures would be very valuable.

At first, it may seem as though behavioural experiments are prohibitively heterogeneous – in terms of model organisms, paradigm and time scale – for a similar community to arise. However, some low-level conceptual consistencies and methodological challenges are common between experiments. For instance, the results (*i.e.* the 'data') feature a set of long time series (sometimes multivariate and irregular), but also contain a formal description of the treatment applied to each individual, the 'metadata'. Storing and accessing data and metadata efficiently involves the implementation of a nested data structure which, in principle, can be shared between acquisition platforms and experimental paradigms.

In the article herein, we describe the `rethomics` platform, an effort to promote the interaction between behavioural biology and data science. `rethomics` is implemented as a collection of packages, altogether offering a solution to import, store, manipulate and visualise large amounts of behavioural results. We also present two practical examples of its application to the analysis of circadian rhythm in fruit flies, a widely studied behaviour.

Design and Implementation

`rethomics` is implemented in `R` [17], which is widely taught and adopted by computational biologists, as a collection of packages (Fig 1). This architecture follows the model of modern frameworks such as the `tidyverse` [18], which results in increased testability and maintainability. In this model, each task of the analysis workflow (*i.e.* data import, manipulation and visualisation) is handled by a different package. At the core of `rethomics`, lies the `behavr` package, used to store large amounts data (*e.g.* position and activity) and metadata (*e.g.* treatment and genotype) in a unique `data.table`-derived object [19]. 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. Visualisation and other results are standard objects, and can therefore be further analysed inside the wide `R` package ecosystem.

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

Ethomics results can easily scale and data structure therefore gains from being computationally efficient – both in term of memory footprint and processing speed. For instance, there could be very long time series, sampled several times per second, over multiple days, for each individual. In addition, time series can be multivariate, encoding coordinates, orientation, dimensions, activity, colour intensity and so on. Furthermore, experiments may feature a large number of individuals. Each individual is also associated with some metadata: a set of '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. A large set of metavariables is an important asset since they can later be used as covariates.

`behavr` tables link metadata and data within the same object, extending the syntax of `data.table` to manipulate, join and access metadata (Fig 2A and B). This approach guarantees that any data point can be mapped correctly to its parent metadata. Furthermore, it 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 example, when one wants to update a 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

Data import packages 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 System (Trikinetics Inc.) data and another one for Ethoscope data [20]. 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 (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

To integrate visualisation in `rethomics`, we implemented `ggetho`, a package that offers new tools that extent the widely adopted `ggplot2` [21]. `ggetho` makes full use of the internal `behavr` structure to summarise of temporal trends. We implemented a set of

new 'layers' and 'scales' that 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`. For instance, `ggplot2` operations such as faceting, transforming axes and adding new layers will function natively with `ggetho`.

Results

In order to illustrate the usefulness of `rethomics`, we provide two use cases examples. The first one is a detailed and reproducible description of the loading and analysis activity data in the context of circadian rhythm, using DAM2 (Trikinetics Inc.) data. The second one shows how `rethomics` integrates with the rest of `R` with an advanced case of a multi-scale analysis of a periodic behaviour using continuous wavelet transform and ethoscope data [20].

Drosophila Activity Monitor

It uses the package `zeitgebr` which implements a suite of methods to compute autocorrelogram, χ^2 [22] and Lomb-Scargle [23] periodograms, and find their peaks.

The study of circadian rhythm, employing fruit flies as an animal model, is a well established research field recently awarded with the Nobel Prize in Physiology or Medicine. Importantly, DAM2 (the second, and most popular, version of the Drosophila Activity Monitors System) is the most widely adopted behaviour recording platform in this research field. We gathered a subset of the data from a recent publication [24], kindly made publicly available by the authors [25]. Wild type flies are highly rhythmic in Light-Dark (LD) cycles and become arrhythmic in constant light (LL). In their study, the authors gain understanding of the function of the molecular clock by showing that overexpression of the gene NKCC makes the flies rhythmic in both LD and LL, and that the endogenous period in LL is longer than 24 hours.

We took two genotypes employed in the study, one control group ($NKCC^{ox}/+$) and one where $NKCC^{ox}$ is overexpressed in all clock neurons ($TIM/NKCC^{ox}$). In particular, we re-analysed two repetitions of the same experiment in which a total of 58 animals were recorded for three to four days in LD and then subjected to constant light for six days. 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. 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):

```
library(damr)      # input DAM 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.

```
metadata <- link_dam_metadata("metadata.csv", ".")   # linking
# print(metadata)                                     # check metadata
dt <- load_dam(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 also keep `id` as an *unambiguous* unique identifier. Indeed, two animals in different experiments may have the same label, but different `id`s. in addition, if the metadata changes – for instance by the addition or removal of individuals – the label is likely to change, not the `id`, which could lead to confusion.

```
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).

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.

```
# make a ggplot object with label on the y and moving on the z axis
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). The `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, autocorrelation and Fourier. 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 NKCC^{ox}/+ flies are mostly arhythmic in LL whilst Tim/NKCC^{ox} 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)) +
          # periodododogram 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")
```

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.

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`):

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

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

```
# 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/NKCC^{ox} 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.

Multi-scale analysis of position

One of the challenges of behaviour analysis is the 'nesting' of events happening over different time scales. In other words, a behavioural variable can be modulated by the circadian rhythm, but also by co-occurring ultradian and infradian rhythms. For instance, an animal could have rhythmic bursts of activity recurring at high frequency (*e.g.* 1 min), but also a circadian (*i.e.* 24 h) regulation of the same variable, activity. In this example, both rhythms happen at time scales separated by approximately three orders of magnitudes, which makes them difficult to visualise and integrate in the same analysis. Being able to keep frequency information over multiple scales is however important in some cases. In particular, when interested in the frequency modulation of one rhythm by another – that is, if the periodicity of a high frequency rhythm itself can be driven by a lower frequency one.

The problem of understanding time series at different scales in not uncommon in fields such as economics [26], climate sciences [27] and ecology [28] where variables are governed by multiple underlying rhythms (*e.g.* tidal, daily, yearly and multi-yearly). One approach is to study a variable of interest in the time/period domain using, for instance, continuous wavelet transform (CWT) [29].

To illustrate how this analysis can be adapted to the study of behaviour, we studied the position of 40 males and 40 virgins females in their glass tubes. The male data was collected from our previous study [20] (controls in figure 5M-P). We used the package `scopr`, part of `rethomics`, to load five days of ethoscope positional data, which we sampled it at 0.1 Hz. Our variable of interest is the position of animals in their tube (from the food end, 0, to the cotton end, 1). Fig 6A-C shows the raw position data at two different scales for two representative animals.

Fig 6. Wavelet analysis of positional data. A: Raw position data for a representative male (top) and virgin female (bottom) *Drosophila* over five days, in black. The thick coloured lines show the local average position. The green rectangles in the background shows the two time windows selected for B and C. B: Close up of A, showing position over one hour, in the beginning of the L phase ($t \in [24, 25]h$). C: Close up of A, showing position over one hour, in the middle of the L phase ($t \in [30, 31]h$). D: Continuous wavelet transform spectrogram for the two representative animals. E: Average spectrogram across 40 males and 40 females. In D and E, the lines on the right shows the marginal power spectra corresponding to the shown spectrograms (average on the period domain).

In order to compute CWT, we used the `WaveletComp` package [30]. We then averaged the result of the five consecutive days in the time/period domain over one circadian day (Fig 6D and E).

As expected, we observe strong, high period (12 h and 24 h), rhythms of the position. In addition, a large amount of signal is detected for low period (around 60 s)

events. This correspond to the position of animals walking along (back and forward) their tubes in a very paced manner.

Interestingly, in females, this low period pace appears to be frequency modulated during the L phase, suggesting a slower walking speed around ZT6 h. In contrast, males shows only a high frequency rhythm around the phase transitions (L to D and D to L). Surprisingly, the peak of high frequency rhythm imply a faster pace in males (approximately 60 s) than in females (approximately 120 s) – indicating that, when active, males walk faster than females.

This non-exhaustive proof of principle illustrates how analysis of behavioural data can be taken further by adapting the wide range of numerical tools already available in the `R` ecosystem.

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.

Several users, in different research groups, have already adopted and are contributing to the future development framework. Several new packages in the `rethomics` framework are currently envisaged. They include utilities to input new behaviour tracking methods and analyse position and multi-animal interactions.

Supporting information

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

Acknowledgements

We would like to thank people who have directly or indirectly contributed to the this manuscript. In particular, Han Kim, for his invaluable comments on the early versions of `rethomics` and his dedicated contribution to the tutorials; Maite Ogueta, for making the results data available; Alice French, Hannah Jones and Diana Bicazan for their comments as early users; Marcus Gosh, for his feed back on the manuscript; Brenna Williams, for her help to support multi-beam DAMS; Patrick Krätschmer, for his time discussing the conceptual framework; and Robert Dickinson, for his support.

TODO: @esteban names of the BA users

References

1. Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, et al. Big Data: Astronomical or Genomical? PLOS Biology;13(7):e1002195. doi:10.1371/journal.pbio.1002195.

2. Reiser M. The ethomics era? Nature Methods;6(6):413–414. doi:10.1038/nmeth0609-413.

3. Faville R, Kottler B, Goodhill GJ, Shaw PJ, van Swinderen B. How deeply does your mutant sleep? Probing arousal to better understand sleep defects in *Drosophila*. Scientific Reports;5:8454. doi:10.1038/srep08454.

4. Pelkowski SD, Kapoor M, Richendrfer HA, Wang X, Colwill RM, Creton R. A novel high-throughput imaging system for automated analyses of avoidance behavior in zebrafish larvae. Behavioural Brain Research;223(1):135–144. doi:10.1016/j.bbr.2011.04.033.

5. Itskov PM, Moreira JM, Vinnik E, Lopes G, Safarik S, Dickinson MH, et al. Automated monitoring and quantitative analysis of feeding behaviour in *Drosophila*. Nature Communications;5:4560. doi:10.1038/ncomms5560.

6. Ro J, Harvanek ZM, Pletcher SD. FLIC: High-Throughput, Continuous Analysis of Feeding Behaviors in Drosophila. PLOS ONE;9(6):e101107. doi:10.1371/journal.pone.0101107.

7. Swierczek NA, Giles AC, Rankin CH, Kerr RA. High-throughput behavioral analysis in *C. elegans*. Nature Methods;8(7):592–598. doi:10.1038/nmeth.1625.

8. Pérez-Escudero A, Vicente-Page J, Hinz RC, Arganda S, de Polavieja GG. idTracker: tracking individuals in a group by automatic identification of unmarked animals. Nature Methods;11(7):743–748. doi:10.1038/nmeth.2994.

9. Brown AE, Bivort Bd. The study of animal behaviour as a physical science. bioRxiv; p. 220855. doi:10.1101/220855.

10. Berman GJ. Measuring behavior across scales. BMC Biology;16:23. doi:10.1186/s12915-018-0494-7.

11. Peng RD. Reproducible Research in Computational Science. Science;334(6060):1226–1227. doi:10.1126/science.1213847.

12. Stodden V, Guo P, Ma Z. Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. PLOS ONE;8(6):e67111. doi:10.1371/journal.pone.0067111.

13. Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Ullah Khan S. The rise of "big data" on cloud computing: Review and open research issues. Information Systems;47:98–115. doi:10.1016/j.is.2014.07.006.

14. Roumpeka DD, Wallace RJ, Escalettes F, Fotheringham I, Watson M. A Review of Bioinformatics Tools for Bio-Prospecting from Metagenomic Sequence Data. Frontiers in Genetics;8. doi:10.3389/fgene.2017.00023.

15. Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, et al. Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods;12(2):115–121. doi:10.1038/nmeth.3252.

16. Leipzig J. A review of bioinformatic pipeline frameworks. Briefings in Bioinformatics;18(3):530–536. doi:10.1093/bib/bbw020.

17. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing;. Available from: `https://www.R-project.org/`.

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

19. Dowle M, Srinivasan A. data.table: Extension of 'data.frame';. Available from: `https://CRAN.R-project.org/package=data.table`.

20. 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.

21. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer;.

22. 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.

23. 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.

24. 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.

25. Ogueta M, Stanewsky R. LL Behaviour of TIM Gal4 > NKCC OX and NKCC OX / + flies;. Available from: `https://zenodo.org/record/1172980`.

26. Aguiar-Conraria L, Joana Soares M. Business cycle synchronization and the Euro: A wavelet analysis. Journal of Macroeconomics;33(3):477–489. doi:10.1016/j.jmacro.2011.02.005.

27. Lau KM, Weng H. Climate Signal Detection Using Wavelet Transform: How to Make a Time Series Sing. Bulletin of the American Meteorological Society;76(12):2391–2402. doi:10.1175/1520-0477(1995)076¡2391:CSDUWT¿2.0.CO;2.

28. Cazelles B, Chavez M, Berteaux D, Ménard F, Vik JO, Jenouvrier S, et al. Wavelet analysis of ecological time series. Oecologia;156(2):287–304. doi:10.1007/s00442-008-0993-2.

29. Grossmann A, Morlet J. Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape. SIAM Journal on Mathematical Analysis;15(4):723–736. doi:10.1137/0515056.

30. Schmidbauer ARaH. WaveletComp: Computational Wavelet Analysis;. Available from: `https://CRAN.R-project.org/package=WaveletComp`.