**DAM analysR**

*Author: Patrick Kratschmer*

**Introduction**

This repository contains work in progress on an R-based analysis package for the Drosophila Activity Monitor (DAM) system.

In order to run it, R needs to be installed on the local computer, and preferably RStudio as an IDE. This is a quick guide to do so:

1. Download R

* R can be downloaded from CRAN: <https://cran.r-project.org/> Simply follow the instructions (and note that XQuartz is needed for Mac users).

1. Download RStudio

* RStudio can be downloaded from: <https://www.rstudio.com/>

Once R and RStudio are installed on the local computer, there are two (so far) extra packages that need to be installed before using DAM\_analysR. These are:

* xlsx
* plyr

To install these packages, and any other, type install.packages("<packagename>") into the console in RStudio. Once this is done, start a new R project in RStudio, which sets the working directory to be the new RProject directory. Now the DAM\_analysR GitHub repository can be downloaded and all its contents be copied into that RProject directory on the local computer; otherwise, for those using git and GitHub, the remote repo (DAM\_analysR) can be forked and set as the working directory in RStudio.

Once the working directory in RStudio contains the contents of DAM\_analysR, the analysis of DAM raw output data can begin. First, run the raw data from the DAM through DAM FileScan; the output files from this need to be copied into the RStudio working directory. Second, source analysR.R by source("analysR.R"). Third, run analysR with the name of the raw DAM output file as the first argument, and the length of the experiment in minutes as the second argument (the default experiment duration is 1440 min, i.e. one day). Store the output of analysR in a variable. Below, I go into detail of what the analysR output contains, and how the respective elements are calculated. To calculate the sleep parameters (24 so far) for each fly, first source sleep\_parameters.R. Then run this function with the output of analysR as the first input argument, and the name of the output Excel spreadsheet as the second argument in parentheses (don't forget the .xlsx file extension). sleep\_parameters will write the results of its caculations into an Excel spreadsheet that will be created in the working directory. Below, I explain the meaning of the sleep\_parameters, and how they are calculated.

**analysR.R**

analysR.R is the base function used to work through the DAM FileScan output. Its first input is the DAM FileScan output, which is a text file (don't forget parentheses and file extension). Optionally, if the experiment was run for a different duration than 1440 min, the length of the experiment in minutes can be passed into analysR as the second argument. analysR's output is a list of 11 elements. Each element can be accessed by the $ operator:

1. light\_regime is a list of 1's and 0's, where 1 indicates light-on, and 0 light-off. Its length is specified by the days\_in\_minutes argument to analysR. Hence, each element of the light\_regime vector indicates whether during a given minute of the experiment the light was switched on or off.
2. transitions is calculated by light\_dark\_transitions.R, which is sourced from within analysR. This function takes two arguments, light\_regime and days\_in\_minutes. It walks through light\_regime until it encounters a transition from 0 to 1 or vice versa, and stores the index of that transition in two separate vectors, one for dark-light transitions, and one for light-dark transitions, respectively. Note that the minute of transition stored is always the 1, i.e. for light-dark transitions, the last minute in light\_regime associated with a 1 is stored, while for dark-light transitions, the first minute associated with a 1 is stored. This is because even if the lights were on for just 1 s during a whole 1 min bin, in the DAM raw data, this minute will be associated with a 1. Hence, by storing the indices of the 1's, we know that for light-dark transitions, this was definitely the last minute during which the lights were on, while for dark-light transitions, this was definitely the first minute during which the lights were on.
3. DAM\_raw is the raw output from the DAM system, which is exactly the content of the text file you pass into analysR as the first argument.
4. DAM\_raw\_clean extracts columns 12 to 44 from DAM\_raw, which corresponds to the light regime (column 12 of DAM\_raw), and the beam crosses per minute for each channel (32 channels in total, resulting in 33 columns). This is done because the first 11 columns are not used for downstream analyses.
5. DAM\_raw\_clean2 binds an extra row of 1's to the beginning of DAM\_raw\_clean, which is important for later calculations of sleep episodes: these are calculated in part by finding a transition from activity to inactivity; if a fly was inactive at the start of the experiment, this can only be interpreted as sleep (if this inactivity lasted for at least 5 min) if we add a row of pseudo-activity to the beginning of each channel. In addition, DAM\_raw\_clean2 has the first column removed, which is stored in the vector light\_regime as stated above, and not carried forward for downstream calculations.
6. DAM\_raw\_clean3 has an additional column added to the start of DAM\_raw\_clean2, which contains a sequence of numbers from 0 to days\_in\_minutes, the second input argument to analysR. This is done to "index" every row in DAM\_raw\_clean3.
7. analogue\_to\_binary takes DAM\_raw\_clean3[, 2:33] (because the first column, the days\_in\_minutes, is not used for this calculation), and transforms any 0 in the data frame into a 1, and any non-0 into a 0. Hence, every minute of inactivity is now represented by a 1, while every minute during which the fly crossed the infrared beam at least once is represented by a 0. This makes analogue\_to\_binary an inactivity representation. Note that days\_in\_minutes is appended as the first column in analogue\_to\_binary after the above calculation is done, so analogue\_to\_binary has 33 columns.
8. five\_min\_bouts is calculated via the function sleep\_define.R (sleep\_define.R must be in the working directory). This function is sourced from within analysR. sleep\_define runs through every column of analogue\_to\_binary (again, except the first one, which is days\_in\_minutes), and looks for five consecutive 1's, i.e. periods of five minutes of inactivity. It does so via a sliding window of length 5, sliding down every column of analogue\_to\_binary from the first row to the fourth-last row (the window would otherwise exceed the length of the column). Whenever it finds a window of five consecutive 1's, it saves a 1 at position 1 of the sliding window in five\_min\_bouts. E.g., if a fly didn't cross the infrared beam between minutes 223 and 227, a 1 is put at position 223 in five\_min\_bouts. As a result, five\_min\_bouts will consist of 0's and 1's, with every 1 indicating that at this moment in time during the experiment (e.g. at minute 223), the fly didn't move for five consecutive minutes, i.e. it didn't move during this minute (e.g. minute 223), and neither during the next four, indicating a period of five minutes of inactivity. If this is interpreted as a single sleep bout, then five\_min\_bouts is a dataframe where every 1 indicates the beginning of a five minute sleep bout (which is how this dataframe got its name). A nine minute sleep bout in five\_min\_bouts would be represented by five 1's in a row, a 300 min sleep bout by 296 1's in a row, an n min sleep bout therefore by (n - 4) 1's in a row, where n >= 5.
9. sleep\_start\_list is calculated by the function sleep\_start.R, which is sourced from within analysR. sleep\_start uses a sliding window of length 2 to slide down each column of five\_min\_bouts. Every time it encounters a transition from 0 to 1, it identifies this as the start of a sleep episode, and saves the index of the 1 in a vector. The result (i.e. sleep\_start\_list) is a list of vectors, where each vector contains the indices of sleep starts for one channel (i.e. one fly) for the course of the whole experiment (defined by days\_in\_minutes in analysR).
10. sleep\_end\_list is calculated by the function sleep\_end.R, which is also sourced from within analysR. Like sleep\_start, sleep\_end also uses a sliding window of length 2 to slide through each column of five\_min\_bouts. It is important to note how sleep\_end calculates the index of sleep end. When the window encounters a transition from 1 to 0, it saves the index of the first 1 plus 4 (due to the nature of what a 1 in five\_min\_bouts represents). E.g., if a fly exhibits a sleep bout of 7 minutes at minute 726, five\_min\_bouts will have 1's at positions 726, 727, and 728. Hence, a transition from 1 to 0 will be noticed by sleep\_end when the window is placed over positions 728 and 729. Because the 1 at position 728 in five\_min\_bouts indicates a sleep bout of 5 minutes, the index that is saved by sleep\_end is 728 + 4 = 732.
11. sleep\_bout\_length is calculated by subtracting every element in sleep\_start\_list from every corresponding element in sleep\_end\_list, plus 1. Hence, every list element in sleep\_bout\_length is a list itself and corresponds to one channel. This sub-list contains the length of every sleep bout exhibited by a specific fly throughout the course of the whole experiment. To get the corresponding start and end times of each sleep bout, sleep\_start\_list and sleep\_end\_list must be indexed by the same index as the bout is associated with in sleep\_bout\_length.

**sleep\_parameters.R**

Once the output of analysR has been stored in a variable, this variable can be passed into sleep\_parameters. Note that sleep\_parameters.R needs to be sourced before calling it. The second input argument to sleep\_parameters is the name of the Excel file that will be written by this function; this name needs to be in double quotes and end with the .xlsx file extension. This Excel file will be written into the working directory; its content is a matrix where each column indicates a separate channel and each row a different sleep parameter. Note that if only a subset of the 32 channels in a monitor is to be analysed, this can be specified in the third input argument to sleep\_parameters; the default is channels 1 to 32. Moreover, the length of the experiment is set to 1440 min, but this can also be changed in the fourth input argument. Lastly, sleep\_parameters can be run for data in light-dark, dark-light, light-light, and dark-dark; in case of light-dark and dark-light, the function will automatically extract the time of transition and use it for downstream calculations. However, in the case of light-light and dark-dark, the user has to input the time of transition (theoretical transition) manually as the fifth input argument ('mytransition') to sleep\_parameters: by default, the value is 721, which represents a theoretical 12:12 pattern. Altogether, there are 24 separate sleep parameters calculated by sleep\_parameters, which are listed below. For further explanation of how they are calculated, please consult the R script directly:

* latency\_first\_day is the time point at which the first sleep bout during the day occurred.
* offset\_first\_day is the time point at which the first sleep bout during the day ended.
* length\_first\_day\_bout is the length of the first day sleep bout.

Note that there is a special case when a fly starts sleeping only once during the day, and does not wake up until the light-dark transition. This is dealt with as follows: if this fly started sleeping at least 5 min before the light-dark transition, this bout is counted as a day sleep bout (and potentially also as a night sleep bout, if it lasted for at least 5 min into the dark period, see below), with latency\_first\_day being the start of this bout and offset\_first\_day the time of the light-dark transition. In cases where a fly either not started sleeping at all during the day, or only did so less than 5 min before the light-dark transition, latency\_first\_day is set equal to the time of the light-dark transition, while offset\_first\_day will be assigned NaN.

* latency\_first\_night is the time point at which the first night sleep bout occurred.
* offset\_first\_night is the time point at which the first night sleep bout ended.
* length\_first\_night\_bout is the length of the first night sleep bout.

As similar special case as described above for day sleep bouts ca also occur for night sleep bouts: a fly may start sleeping during the day, and only wake up from this sleep bout during the night (such bouts are from here on called transition bouts). If the night part of a transition bout lasts for at least 5 min, this is counted as a night sleep bout, and latency\_first\_night is set to the time of the light-dark transition, while offset\_first\_night will be the time this transition bout ended. If a fly either did not sleep at all during the night, or only shows a transition bout that lasted less than 5 min during the dark period, latency\_first\_night will be set to the length of the experiment (days\_in\_minutes), and offset\_first\_night will be assigned NaN.

* latency\_first\_max\_day is the time point at which the longest sleep bout during the day occurred.
* offset\_first\_max\_day is the time point at which the longest day sleep bout ended.
* number\_of\_max\_day\_bouts shows how many maximum sleep bouts during the day a fly exhibited; this is relevant if a fly sleeps n > 1 times for a length that represents the longest sleep bout during the day.
* length\_first\_max\_day\_bout is the length of this/these maximum day sleep bout/s.

Again, there is the complication of transition bouts: if there is a transition bout whose day part represents a longer duration than any other day bout, the day part of this transition bout will be the maximum day bout. In this case, latency\_first\_max\_day is the time this transition bout started, while offset\_first\_ma\_day is equal to the time of the light-dark transition. In the rare case where there is a transition bout whose day part is exactly as long as another, previously occurring day sleep bout, latency\_first\_max\_day and offset\_first\_max\_day will not change, because these parameters refer to the *first* maximum day bout, and clearly the day part of a transition bout, if as long as another maximum day bout, will be the last one occurring during the day, by definition. In this case, only number\_of\_max\_day\_bouts will be incremented by 1. If there is either no day bout at all, or a transition bout of length less than 5 min, latency\_first\_max\_day, offset\_first\_max\_day, and length\_first\_max\_day\_bout are assigned NaN, while number\_of\_max\_day\_bouts is 0.

* latency\_first\_max\_night is the time at which the first maximum night bout occurred.
* offset\_first\_max\_night is the time at which the first maximum night bout ended.
* number\_of\_max\_night\_bouts shows how many equally long maximum night sleep bouts occurred.
* length\_first\_max\_night\_bout represents the lengths of this/these maximum night bout/s.

The complication transition bouts pose to these parameters is twofold: first, as above, if there is a transition bout whose night part lasts for a duration longer than the maximum night bout, the night part of this transition bout will be counted as the maximum night bout, with latency\_first\_max\_night equal to the time of the light-dark transition, and offset\_first\_max\_night equal to the end of this transition bout. Second, because we are calculating the latency and offset of the *first* maximum night bout, in this case such a transition bout, rather its night part, will actually be the first bout (not the last as for day bouts). Hence, if the night part of a transition bout exactly as long as the maximum night bout, latency\_first\_max\_night and offset\_first\_max\_night will refer to this transition bout in the same way as explained above, for cases where the night part of a transition bout is longer than all other maximum night bouts. If there is either no night sleep at all, or only a transition bout whose night part lasts for less than 5 min, latency\_first\_max\_night, offset\_first\_max\_night, and length\_first\_max\_night\_bout are assigned NaN, while number\_of\_max\_night\_bouts is 0.

* latency\_first\_max\_overall\_bout is the time at which the first overall (no differentiation between light-on and light-off) maximum bout started.
* offset\_first\_max\_overall\_bout is the time at which this first overall maximum bout ended.
* number\_of\_max\_overall\_bouts is the number of equally long maximum bouts overall.
* length\_first\_max\_overall\_bout is the length of this/these overall maximum bout/s.

The calculation for the above parameters does not split the data into light-on and light-off. Hence, if a transition bout is the longest bout a fly exhibited during an experiment, this bout, without being split into dat and night parts, will be shown as the overall maximum bout.

* total\_day\_bouts is the number of day sleep bouts.
* total\_night\_bouts is the number of night sleep bouts.
* total\_bouts\_overall is the number of (*not* split into day and night phases) sleep bouts throughout the whole experiment.
* transition\_bouts is the number of transition bouts exhibited by a fly. By definition, this can only be 1 or 0.
* last\_day\_offset is the time point at which the last day sleep bout ended.
* last\_night\_offset is the time point at which the last night sleep bout ended.

If there is a transition bout that lasts for at least 5 min during light-on, last\_day\_offset will be equal to the time to the light-dark transition.