# **Required R package**

The functions need R package “chron” be installed for converting string to time. To install “chron” package use the command *install.packages(“chron”)*.

# **Function Input**:

## count.oscillation

This is the main function to count oscillations. The program automatically determines where to draw the horizontal line to count the spikes and do the other calculations as well. Sometimes the function may have trouble identifying a good place to draw the horizontal line. In this case the function [*count.osc.drawline*](#_count.osc.drawline) is recommended.

**dataset** : dataset that contains the oscillation data. Please refer to “[Input file format](#_Input_file_format:)” for dataset format. It is important that the input data follow the same format.

**s** : this designates the sample in the dataset to be analyzed with the entered sample being s=1 which corresponds to the 3rd column in the input data.

**lag.time1**  : this defines the duration for which AUC is calculated since the start of the first transient.

**save.plot** : A Boolean variable indicating if a pdf file will be created to save the plot.

## count.osc.drawline

This function counts oscillations based on user provided value to draw the horizontal line(s). In addition to the three input parameters mentioned above, this function requires several additional input parameters:

**mid.point0** : this is the y-axis value which determines where to draw the horizontal line(s) to count the oscillations. More than one value can be provided for different y values at different times (as defined by “at.time”).

**at.time** : if more than one mid.point0 values were provided, at.time should be specified to indicate where the next mid.point value will be applied. See the [example code](#_Example_code:) for details.

**length.at.height** : the y-axis value at which the duration of the first spike is measured. If left as the default value of NA the function will automatically pick the value

**file1** : file name of the file where summary statistics will be saved.

**save.plot** : A Boolean variable indicating if a pdf file will be created to save the plot.

# **Function Output**:

**length of 1st Osc**  : Duration of the first spike in minute

**Total AUC** : Total area under the curve recorded

**AUC w 1st peak**  : Area under the curve from the start of the first transient until “lag.time” (default to 1 hour) afterward including the first transient

**AUC w/o 1st peak**  : Area under the curve from the start of the first transient until “lag.time” (default to 1 hour) afterward excluding the first transient

**Delay b/f 1st**  : Number of minutes before the start of the first transient

**n.osc (all)**  : Total number of oscillation recorded

**time.first.last (all)**  : Duration of oscillations from the first transient until the last

**osc.per.10min (all)**  : Number of oscillations per 10 minutes based on all oscillations observed

**meanPeak (all)**  : Mean height of the spikes based on all oscillations

**n.osc (1hr)**  : Number of Oscillation until 1 hour after the start of the first transient

**time.first.last (1hr)**  : Duration of oscillations from the first transient until 1 hour later

**osc.per.10min (1hr)**  : Number of oscillations per 10 minutes based on the first 1 hour of oscillations

**meanPeak (1hr)**  : Mean height of the spikes based on the first 1 hour of oscillations

**n.osc (2hr)**  : Number of Oscillation until 2 hours after the start of the first transient

**time.first.last (2hr)**  : Duration of oscillations from the first transient until 2 hours later

**osc.per.10min (2hr)**  : Number of oscillations per 10 minutes based on the first 2 hours of oscillations

**meanPeak (2hr)**  : Mean height of the spikes based on the first 2 hours of oscillations

# **Input file format:**

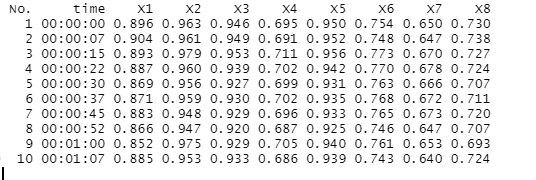
The samples are in columns and data from each time point are in rows.

The first column is recording number.

The second column is the time of recording (column header must be “time”). The time must be in the format of “hh:mm:ss”, e.g., “00:00:00” is the start of the recording time.

The third column and on is the readings of the samples. One column corresponds to one sample and one row corresponds to one time point.

Here is a screenshot of an example input file.



# **Example code:**

### read in example data file and convert dat$time to time.

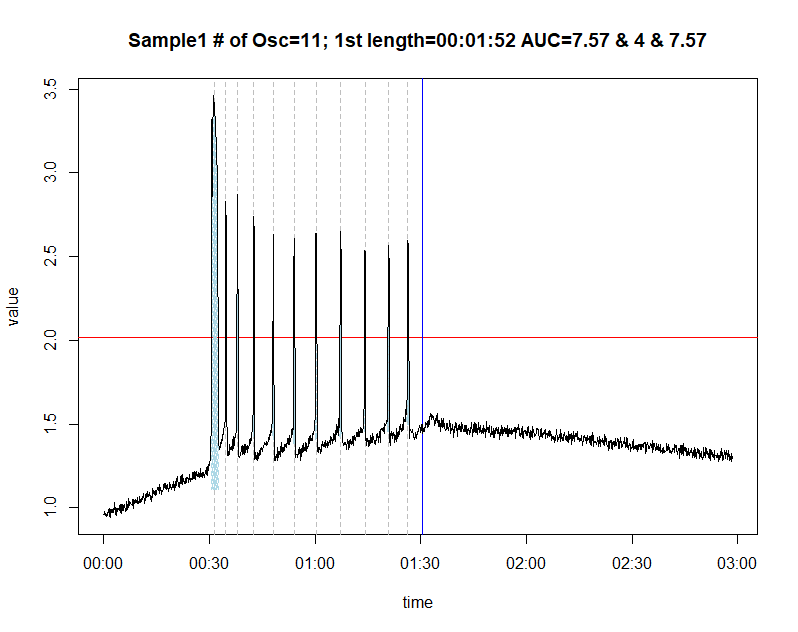
read.table(‘example.txt’, header=T, as.is=T)->dat

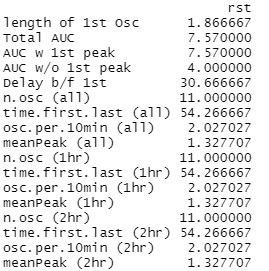
dat$time <- chron(times=dat$time)

### calling count.oscillation function to analyze the 1st sample

count.oscillation(dat, s=1)

Output:





### calling count.osc.drawline function to analyze the 3rd sample. Three “*mid.point0*” and two “*at.time*” values are provided. The function will count the oscillation going through two segments of horizontal lines: 1) at y=2.15 from time 00:00:00 to 01:30:00 and 2) at y=2.2 from time 01:30:00 to 02:00:56, and y=2.15 from 02:00:56 to the end of the recording time.

count.osc.drawline(3, mid.point0=c(2.15,2.2,2.15), at.time = c('01:30:00','02:00:56'), length.at.height=2)

Output:

