1 Installation

`rethomics` is still under heavy development, so it cannot be uploaded to the CRAN (Comprehensive R Archive Network), where most stabe packages are. Instead, we can install R using Hadley Wickham's popular `devtools`[1]. Once you have `devtools` installed, it should be straightforward to install `rethomics`

```
> library(devtools)
> install_github("gilestrolab/rethomics",subdir = "rethomics")
```

Check for error messages. Then ensure the package is installed by loading it

```
> library(rethomics)
```

2 Loading data

2.1 Data structure

In `rethomics` the goal is to store *all* the behavioural data in one single dataframe, which is standard for subsequent statitical analysis. In such a dataframe, every row correspond to a single measurment; that is he position of one animal at one time. Every column describes a statistical variable such as t, and X and Y positions, but they can also hold information about arbitrary conditions such as treatment, sex, age, genotype and so on.

By convention:

- t, the variable holding the time, is always *in seconds*
- X and Y, are *relative to the width of the region* they originate from, and origin is top-left

  For a single **experiment**, and when tracking a single animal (i.e. in a single **region**), your data table could look like:

```
> library(rethomics)
> data(multiple_iterative_y_mazes)
> single_animal <- multiple_iterative_y_mazes[experiment_id == "female_FALSE_11.db"]
> print(single_animal)
```

```
          experiment_id region_id    sex sleep_deprived id               path
  1: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
  2: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
  3: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
  4: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
  5: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
 ---
```

[1] installation instructions are available here https://github.com/hadley/devtools/blob/master/R/run-examples.r

```
4167: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
4168: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
4169: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
4170: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
4171: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
            t         x           y           w           h phi has_interacted
   1:  12.166 0.5505208 0.565104167 0.005729167 0.001562500   0          FALSE
   2:  12.200 0.5510417 0.564062500 0.008854167 0.003645833   0          FALSE
   3:  12.233 0.5520833 0.563020833 0.009375000 0.005729167   0          FALSE
   4:  12.266 0.5515625 0.562500000 0.011458333 0.006770833   0          FALSE
   5:  12.300 0.5510417 0.561979167 0.011458333 0.007812500   0          FALSE
  ---
4167: 151.633 0.6270833 0.002604167 0.009375000 0.003645833   0          FALSE
4168: 151.666 0.6265625 0.002604167 0.007812500 0.003645833   0          FALSE
4169: 151.833 0.6369792 0.001562500 0.003645833 0.001562500   0          FALSE
4170: 151.966 0.6328125 0.001562500 0.004687500 0.001562500   0          FALSE
4171: 152.100 0.6125000 0.003125000 0.004687500 0.004687500  90          FALSE
```

The first thing you may notice is that the first few columns seem unnecessary as they have constant values. This is because they describe variables that vary between individuals, and we have only one individual.

In order to unerstand the need for additionnal columns, we can load data from multiple experiements:

```
> library(rethomics)
> data(multiple_iterative_y_mazes)
> print(multiple_iterative_y_mazes)
```

```
            experiment_id region_id    sex sleep_deprived id               path
    1: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    2: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    3: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    4: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    5: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
   ---
40426:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40427:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40428:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40429:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40430:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
             t         x           y           w           h phi has_interacted
    1:  12.166 0.5505208 0.565104167 0.005729167 0.001562500   0          FALSE
    2:  12.200 0.5510417 0.564062500 0.008854167 0.003645833   0          FALSE
    3:  12.233 0.5520833 0.563020833 0.009375000 0.005729167   0          FALSE
    4:  12.266 0.5515625 0.562500000 0.011458333 0.006770833   0          FALSE
    5:  12.300 0.5510417 0.561979167 0.011458333 0.007812500   0          FALSE
```

```
   ---
40426: 191.966 0.3661458 0.005208333 0.025520833 0.011458333  36          FALSE
40427: 192.000 0.3651042 0.004687500 0.024479167 0.011458333  37          FALSE
40428: 192.033 0.3645833 0.004166667 0.023958333 0.011979167  35          FALSE
40429: 192.066 0.3625000 0.006770833 0.021875000 0.011979167   0          FALSE
40430: 192.100 0.3625000 0.006250000 0.021875000 0.010937500   0          FALSE
```

The data will *always* have two columns: **experiment_id** and **region_id**. Together, these columns form a *key*, that is combinations of experiement *and* region represent unique animals. in orther words, we can identify, unambiguously, any animal from its region and experiement identifier. Instead of using the legacy data.frames, `rethomic` takes advantage of Matt Doyle's powerful `data.table` package[2]. This makes it very easy and efficient to work with large amount of behavioural data. Common operation could involve filtering data and computing variable per condition or per individual. Let us go through sevral examples:

```
> library(rethomics)
> data(multiple_iterative_y_mazes)
> print(multiple_iterative_y_mazes)
```

```
           experiment_id region_id    sex sleep_deprived id                path
    1: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    2: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    3: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    4: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
    5: female_FALSE_11.db         1 female          FALSE 11 female_FALSE_11.db
   ---
40426:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40427:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40428:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40429:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
40430:   male_FALSE_25.db         1   male          FALSE 25   male_FALSE_25.db
            t         x           y           w           h phi has_interacted
    1:  12.166 0.5505208 0.565104167 0.005729167 0.001562500   0          FALSE
    2:  12.200 0.5510417 0.564062500 0.008854167 0.003645833   0          FALSE
    3:  12.233 0.5520833 0.563020833 0.009375000 0.005729167   0          FALSE
    4:  12.266 0.5515625 0.562500000 0.011458333 0.006770833   0          FALSE
    5:  12.300 0.5510417 0.561979167 0.011458333 0.007812500   0          FALSE
   ---
40426: 191.966 0.3661458 0.005208333 0.025520833 0.011458333  36          FALSE
40427: 192.000 0.3651042 0.004687500 0.024479167 0.011458333  37          FALSE
40428: 192.033 0.3645833 0.004166667 0.023958333 0.011979167  35          FALSE
40429: 192.066 0.3625000 0.006770833 0.021875000 0.011979167   0          FALSE
40430: 192.100 0.3625000 0.006250000 0.021875000 0.010937500   0          FALSE
```

[2]tutorial available at todo

```
> #WE can simply call this data table `dt'
> dt <- multiple_iterative_y_mazes
> # keeping only females
> dt_female <- dt[sex == 'female',]
> # excluding any data point before 30 secondes (i.e. keeping >= 30s)
> dt_currated <- dt[t >= 30,]
> # Computing, per animal, the time spent in the experiment
> summary_dt =  dt[,.(time_spent = max(t) - min(t)),
+                 by=key(dt)]
> print(summary_dt)

        experiment_id region_id time_spent
1: female_FALSE_11.db         1    139.934
2:  female_FALSE_2.db         1    149.300
3:  female_TRUE_11.db         1    235.200
4:  female_TRUE_21.db         1    147.866
5:   female_TRUE_4.db         1    181.000
6:   female_TRUE_5.db         1    184.567
7:   female_TRUE_7.db         1    231.133
8:   male_FALSE_25.db         1    184.634
```

Much more can be acchived using `data.table`, so I would strongly recco-ment to, at least, read the package introduction[3].

2.2 Loading one file

2.3 Fancier queries

2.4 Loading from network drive

2.5 Analysing data

2.6 Sleep annotation

2.7 Visualisation

2.7.1 Overview plot

2.7.2 Ethograms

2.8 Bout analysis

[3]