Manuscript\_Word

Januar Harianto

13/10/2017

# Introduction

Metabolic rate is a fundamental trait in organisms, associated with virtually all aspects of biology from individual traits such as growth and feeding rate (Kiørboe and Hirst, 2014), to ecological interactions (Seibel and Drazen, 2007), community dynamics (Barneche et al., 2014), and the behaviour of whole ecosystems (Brown et al., 2004). The most common method of measuring metabolic rate is by respirometry, a technique that quantifies the uptake of oxygen over time of an organism (e.g. fish, Kelly et al., 2014), group of organisms (e.g. phytoplankton, Padfield et al., 2016), or part of an organism (e.g. mussel tissue; Stapp et al., 2017).

Respirometry has a long history in experimental biology, having been used for at least a hundred years when Ege & Krogh (1914) studied gas exchange in goldfish. There are at least four broad methodological approaches in respirometry; closed, intermittent flow, flowthrough and open. In closed respirometry, decrease is measured within a hermetically sealed chamber of known volume, sometimes set within a closed loop to allow circulation or mixing of the environment within the chamber. The oxygen recordings may be continuous through use of an oxygen probe, periodic through withdrawing water or gas samples at set intervals, or a two-point measurement consiting of the initial and final concentrations. Typically, metabolic rates are estimated from respirometry by assuming a linear relationship between variables, and estimates of metabolic rate are straightforward in constant volume respirometry with the equation (Lighton, 2008):

where is the slope of the regression that relates O2 concentration to time, or in the case of a two-point measurement, the difference in concentration divided by time, and is the volume of the container.

Intermittent flow respirometry is similar to closed, however, periodically (typically after a set time period or decrease) the chamber is flushed with new water or air, returning it to initial conditions, then sealed again and the experiment repeated (Svendsen et al., 2016). Intermittent flow is essentially closed respirometry, but with an straightforward way of doing repeated measures. Depending on the metabolic rate metric being investigated, final respiration rate may be calculated as the mean of the measures (e.g. citation), or the lowest or highest rates recorded in any trial (e.g. Carey et al., 2016).

Flowthrough respirometry involves a closed chamber, but with a regulated flow of air or water through it at a precisely determined rate. After equilibrium has been achieved, the oxygen concentration differential between the inflow and outflow channels, along with the flow rate allows calculation of the oxygen extracted from the flow volume per unit time (Lighton, 2008):

where is the rate of consumption in milligrams per minute, and are the incurrent and excurrent concentrations, and is the flow rate of water through the system.

A final method is open respirometry, in which an open tank or semi-enclosed area is used, but the input or mixing rate of oxygen from the surroundings is known or found to be negligible relative to oxygen consumption of the specimens (Leclercq et al., 1999). It is seldom used, but for some applications it is a sufficient and practical methodology (e.g. Gamble et al., 2014). The common equation used for open respirometry is (Leclercq et al., 1999):

where is the slope of the regression that relates concentration to time, and is the air-sea oxygen flux as determined by Fick's Law.

With modern measurement techniques available, researchers are collecting increasingly large, high-resolution respirometry data, often capturing multiple response variables such as maximum metabolic rate (MMR, or ) or standard (or resting) metabolic rate (SMR, or ), or critical oxygen tension (; Yeager & Ultsch, 1989), over long periods (e.g. 20 h, Norin & Malte, 2012). In most cases, processing the data involves an *ad hoc* selection of data points and them manually performing calculations in a spreadsheet program (e.g. Microsoft Excel) or an integrated development environment (IDE, e.g. R and Matlab). These approaches can sometimes be repetitive and difficult, where software like Excel struggle with datasets over a few thousand data points, while IDEs require a degree of expertise and come with substantial learning curves. Some dedicated software are available to perform such calculations, but most are proprietary and require licensing requirements or even hardware dongles (e.g. AutoResp by Loligo Systems), complicating or preventing their use on multiple machines.

Recently, a number of free R software packages have appeared that are designed for respirometry, such as respirometry (<https://CRAN.R-project.org/package=respirometry>), rMR (<https://CRAN.R-project.org/package=rMR>), and to a certain extent, LoLinR (Olito et al., 2017). Both respirometry and rMR provide useful tools to help set up aquatic respirometry experiments, but are not focused on the analysis of data. In particular, the rMR package has functions for the analysis of metabolic rate over intervals and the calculation of , but does not specifically focus on any respirometry method. For LoLinR, the package provides a statistically robust way of automatically detecting a best-fit regression which are applicable for closed and flow-through respirometry. However, LoLinR's processing time increases exponentially with large datasets, with data containing more than 500 observations taking a few minutes to run (Olito et al., 2017), which makes LoLinR difficult to use for large amounts of data.

Here we describe the R package respR, a collection of functions desgined to provide a structured workflow for the analysis of respirometry-related data [@tab:tab1]. The package contains utilities to: (1) analyse closed, intermittent, flow-through and open respirometry data, (2) scale to volume and/or mass, and (3) automatically and rapidly detect MMR, SMR, or "best-fit" results in large datasets with the help of traditional rolling regression techniques and k-means clustering. Other smaller, but useful, functions are also avaliable, and are desribed in more detail in our online html vignette (<https://januarharianto.github.io/respR/>). While the package has a strong focus on aquatic respirometry, the principals are the same in air respirometry, so as long as the user understands the caveats, there is no reason why some of our functions could not be used with these respirometry data (full support for air-respirometry is in our roadmap for future versions). The package is also suitable for both marine and freshwater respirometry -- some oxygen concentration conversions require a salinity value, but in freshwater respirometry this should simply be set to zero. We demonstrate the utility of the package by analysing example data collected using closed, intermittent and flow-through respirometry.

# Package overview

respR can be automatically installed from the GitHub repository using the devtools package:

devtools::install\_github("januarharianto/respR")  
library(respR)

To explore respR and its main functions we have provided example data that can be loaded directly. The first data object, urchin2013, contains measurements of oxygen consumption in 16 individual sea urchins (*Heliocidaris erythrogramma*) and 2 "blank" measurements of background respiration (Harianto, *unpublished*). The intermittent data object contains three repeated measures of oxygen consumption in *H. erythrogramma*, collected in a single session with intermittent respirometry (Carey et al. 2016, *JEB*). The flowthrough data object contains measurements of oxygen consumption, via flow-through respirometry, in a species of chiton, *Mopalia lignosa* (Carey, *unpublished*). Detailed information about all example data, including their source and methods, can be obtained with the ? command in the R console (e.g. ?urchin2013).

## Data import and exploration

Data should be formatted correctly before use in respR. The function check.input() performs error checks on a data frame of any size, and can extract a two-column data frame for subsequent analyses. Data must be numeric and of equal length, because respR has limited support for date/time data (e.g. POSIXct and POSIXlt classes) and performing calculations on different time formats have many caveats that are beyond the scope of the package. The function also checks that time data is sequential without duplicates, and warns the user when time data is not evenly-spaced. If a two-column data frame object is used, the function automatically plots the data. The locations (i.e. row numbers) of the errors can be identified so that necessary corrections can be made.

check.input(urchin2013) # check entire data frame  
check.input(urchin2013, xcol = 1, ycol = 3) # check columns 1 and 3 only

It should be noted that invoking check.input is optional - the main functions in our package will readily accept any data frame as long as data are numeric. Running check.input is a qualitative step that simply allows for data exploration and flags potential issues about the data before it is further analysed.

## Estimating rates of change in concentration

The functions calc.rate(), auto.rate() and pcrit()