

METABOLIC COST OF *OCTOPUS RUBESCENS* CHROMATOPHORE SYSTEM

by

SOFIA CAMILLE SONNER

A THESIS

submitted to

WALLA WALLA UNIVERSITY


in partial fulfillment of
the requirements for the
degree of

MASTER OF SCIENCE

18 May 2021


This thesis for the Master of Science degree
has been approved by the Department of Biological Sciences
and the Office of Graduate Studies
Walla Walla University



Major Professor


Committee Member


Cecilia Brothers
Committee Member

Committee Member


Dean of Graduate Studies


Observer of the Process - Graduate Representative


Candidate

Date

ABSTRACT

Color change is a phenomenon found in a wide diversity of animals, including arthropods, fishes, reptiles and cephalopods. Color change in cephalopods, however, is unparalleled in its rapidity and detail. It is controlled primarily by chromatophore organs, small intradermal sacs of pigment that are expanded and contracted by musculature. Although it might be assumed that color change carries an energetic cost, to date no study has directly demonstrated an energetic cost associated with color change in any animal, much less its magnitude. Investigating the energetic cost of the octopus chromatophore system may contribute knowledge to the general principles of camouflage, how it impacts other processes and energy budgets, and how it evolved. In this study, I measured the differences in oxygen uptake and chromatophore area among periods of chromatophore expansion and contraction from excised skin of *Octopus rubescens*. From these data I was able to estimate the energetic cost to a whole octopus by measuring whole body surface area and multiplying surface area by the per area metabolic rate of excised skin. I could then compare these data to routine metabolic rates of this species to put the energetic cost of the chromatophore system into context. The results indicate that the chromatophore system has a relatively high energy demand, which may imply a high energetic cost to octopus camouflage. This is the first known study to directly measure the energy demand of color change for any animal, and thus, there are many prospects for future research to expand upon this finding.

TABLE OF CONTENTS

LIST OF TABLES	5
LIST OF FIGURES	6
INTRODUCTION	7
Color Change in Animals	7
Why Octopuses Change Color	8
The Chromatophore System	9
Metabolic Cost of Chromatophores	12
Catch Muscle	14
Objectives and Hypothesis	14
METHODS	16
Octopus Collection and Housing	16
Microrespirometry	16
Design	16
Blank Testing	18
Skin Testing	19
Photograph Analysis	20
3D Scanning	20
Data Analysis and Statistics	21
Empirical Determination of Oxygen Sensor Temperature Sensitivity	21
Metabolic Rate Data Analysis	21
Calculating Chromatophore Metabolic Rate of Whole Octopus by Mass	25
Chromatophore Metabolic Rate Comparison to Whole Octopus Routine Metabolic Rates	26
RESULTS	27
Microrespirometry	27
Blank Testing	27
Metabolic Rate Determination of Excised Skin	27
3D Scanning	28
Chromatophore Metabolic Rate Comparison to Whole Octopus Routine Metabolic Rates	28
DISCUSSION	30
Additional Considerations	31
Future Research	32
ACKNOWLEDGEMENTS	34

REFERENCES	35
TABLES & FIGURES	37
APPENDIX A	55
Excised Skin Photograph Analysis Steps Using FIJI	55
APPENDIX B	67
3D Scanning Steps Using EinScan Desktop 3D Scanner	67
APPENDIX C	
R Markdowns for Data Analysis	68
Respirometer Oxygen Fastness Test	68
Empirical Temperature Response Determination	72
Chromatophore Oxygen Consumption	77
Skin Analysis	163

LIST OF TABLES

Table 1. Information on each <i>Octopus rubescens</i> used in experimentation	37
Table 2. Oxygen consumption, mean oxygen level and mean temperature during respirometry of excised octopus skin	38
Table 3. Mean area of skin covered by chromatophores during the blue periods and the non-blue period, the difference of those means, oxygen consumption per increased area of chromatophore coverage, and adjusted oxygen consumption	39

LIST OF FIGURES

Figure 1. Assembled custom respirometer as viewed from above	40
Figure 2. Assembled custom respirometer as viewed from the side	41
Figure 3. Placement of the respirometer during experimentation	42
Figure 4. Main components of the experimental setup	43
Figure 5. Difference between temperature and oxygen plotted for octopus 1r at various time offsets between the two datasets	44
Figure 6. Oxygen concentration vs. time to determine the background light correction	45
Figure 7. Example of oxygen concentration data from 5-minute periods of light in which 45s was trimmed from the beginning and 15s was trimmed from the end	46
Figure 8. Relationship between metabolic rates and oxygen levels at which they were collected for both red and blue light periods of all skin sampled	47
Figure 9. Relationship between temperature and oxygen correction multiplier to correct oxygen readings to 19°C	48
Figure 10. Oxygen seep for the custom respirometer used in experimentation	49
Figure 11. Histogram indicating the median metabolic rate estimate of chromatophores and its 95% confidence interval	50
Figure 12. Relationship between octopus mass and metabolic rate per mm ² of skin	51
Figure 13. Relationship between <i>Octopus rubescens</i> surface area and mass	52
Figure 14. Relationship between routine metabolic rate and octopus mass	53
Figure 15. Relationship between octopus mass and metabolic rate	54

INTRODUCTION

Color Change in Animals

Color change is widespread in the animal kingdom, and is exhibited in crustaceans, insects, cephalopods, amphibians, reptiles, and fishes (Stuart-Fox & Moussalli 2008). However, research regarding color change is limited due to the challenge of obtaining organisms capable of camouflage and due to issues with experimental manipulations of animal color patterns; for example, there may be differences in the spectral properties of natural and artificial colors used for color manipulations or differences in the appearance or behaviour of real animals and models (Stuart-Fox & Moussalli 2008).

While the underlying mechanisms of color change in most animals would seem to indicate that color changing requires an energetic input, to date no study has directly demonstrated an energetic cost associated with color change in any animal, much less the magnitude of any potential energetic cost. Neither chameleons nor octopuses exhibit maximum crypsis at all times, which may indicate that it is physiologically costly to do so (Hanlon et al. 1999, Stuart-Fox & Moussalli 2008). Instead, remaining only moderately rather than highly cryptic may be less expensive while still providing some advantage in the presence of an undetected predator (Hanlon et al. 1999). In one of the only attempts to date attempting to show the energy demand of color change, Trinidadian guppies (*Poecilia reticulata*) were exposed to either a changing color environment in their holding tanks, or a constant color environment. Those exposed to the changing environment changed their body coloration more often and increased their food consumption relative to those exposed to a constantly colored environment. This was

interpreted as evidence of an energetic cost associated with color change (Rodgers et al. 2013), however others have questioned if this was instead an increased energy usage due to stress induced by the changing environment. In addition to energetic costs with the process of color change itself, the acquisition and expression of color pigments by animals is thought to carry some cost since pigments must either be obtained through the environment or synthesized by the organism (Rodgers et al. 2013).

Why Octopuses Change Color

Cephalopod color change is thought to have evolved due to the selective pressures that resulted from the loss of a protective external shell and thus, newly exposed body surface and ability to adopt an active lifestyle over time (Messenger 2001, Mather et al. 2010). Presumably, octopuses that had better powers of crypsis could avoid being eaten by preying fish and thereby removed from the gene pool (Mather et al. 2010).

Octopus color change abilities serve primarily three functions: predator evasion, communication, and hunting (Messenger 2001, Mather et al. 2010). Of these, color change is chiefly used as an anti-predator defense, since octopuses are prey to nearly all of the major marine carnivores (Mathger et al. 2011). Benthic octopuses display rapid adaptive camouflage so that they are able to forage and find mates while avoiding predation by visual predators (Mathger et al. 2011). Color-based evasion makes the prey not only hard to find but also exceptionally difficult to follow due to unpredictable changes in appearance (Mather et al. 2010).

Color change may also be used for interspecific and intraspecific communication (Messenger 2001). In these cases, it may serve to startle predators, attract potential mates,

or signal fighting ability (Messenger 2001). Other chromatic behavior has been recorded that does not seem to aid in concealment or communication; for example, *Octopus rubescens* changes color when attacking prey, the purpose of which is not entirely clear (Messenger 2001, Mather et al. 2010). From an evolutionary point of view, color changes that do not contribute to either signaling or camouflage are side effects, though if harmful, they likely would be selected against (Godfrey-Smith, 2016).

The Chromatophore System

Cephalopod color change, in regard to speed of change and diversity of patterns, is unparalleled by any other known animal group on Earth (Hanlon 2011). These dramatic changes in color and pattern, which occur in less than one second, are produced by modulating specialized intradermal structures including chromatophores, iridophores, and leucophores (Hanlon 2007, Hanlon 2011). This involves a variety of mechanisms; however, it has been studied most with regard to changes in chromatophore organs (Messenger 2001, Duarte et al. 2017). Chromatophores are complex structures composed of a pigment sac surrounded by an elastic membrane, to which 15–25 striated muscle fibers are directly attached around the periphery (Hanlon 2007, Messenger 2001, Kingston et al. 2015).

The pigment sac of each chromatophore organ is expanded, thus displaying the pigment, as its corresponding radial muscle fibers are contracted, and contracted and hiding the pigment when its radial muscle fibers relax (Hanlon 2007, Kingston et al. 2015). Each chromatophore is less than one millimeter in diameter when expanded and contains just one color of pigment (Godfrey-Smith 2016). Pigments contained in the

center of each chromatophore may be yellow, orange, red, brown, or black, although this varies among species, and usually an animal has three different colors of chromatophore (Messenger 2001, Godfrey-Smith 2016). Chromatophore organs are arranged in the dermis in layers with a vertical hierarchy, in which each layer carries a different pigment color (Hadjisolomou & El-Haddad 2017).

Due to the lack of research on cephalopod brains, little is known about how octopuses control their changes in color and texture (Mather et al. 2010).

Chromatophores seem to be controlled both locally and through input from the central nervous system (Hanlon 2011, Ramirez & Oakley 2015). The radial muscles that control chromatophores are selectively activated by motor neurons (Messenger 2001, Hanlon 2007). Although the brain seems to play a role in camouflage, the ability and extent of this is unknown (Mather et al. 2010). The nervous system has descending control, although there is not believed to be a feedback loop in which the octopus monitors the patterns it displays by communicating the skin's sensing to the brain; rather, this information may remain local (Mather et al. 2010, Godfrey-Smith 2016). Brain control circuitry for camouflage may be of great variety, since this includes control of posture, skin texture, pattern, and prompt display changes (Mather et al. 2010). However, the functioning of this system is barely understood, despite being a key evolutionary mechanism (Hanlon 2011, Mather et al. 2010).

Despite the involvement of cephalopod eyes for detecting light, and the central nervous system for controlling chromatophore activity, chromatophores may also be controlled locally by the peripheral nervous system and there is evidence that chromatophores themselves are photosensitive (Kingston et al. 2015, Ramirez & Oakley

2015). The visual transduction components rhodopsin, retinochrome, and G_{α} proteins have been found in the pigment cell membranes, radial muscle fibers, and sheath cells of squid and cuttlefish chromatophores (Kingston et al. 2015). These components may indicate extraocular photoreception, since phototransduction in the retina is thought to rely on these same components, though the specific sequence of events is not clear (Kingston et al. 2015). Nevertheless, this indicates that cephalopod dermal tissues, specifically chromatophores, contain the basic components required for a distributed light-detecting system for non-visual functions (Kingston et al. 2015).

Furthermore, *Octopus bimaculoides* skin can both sense light and respond directly via its chromatophores, even when the skin is detached from the body (Ramirez & Oakley 2015). Light detection causes the chromatophores to expand significantly and repeatedly in excised skin when exposed to bright light; this behaviour is called light-activated chromatophore expansion, or LACE (Ramirez & Oakley 2015). The LACE responses show that octopus skin is intrinsically light sensitive, independent of the eyes (Ramirez & Oakley 2015).

R-opsin, a key light-sensing protein in the eyes of octopuses and other animals, is expressed in the skin of *O. bimaculoides*; its role in light sensing suggests that it detects light in octopus skin and underlies LACE (Ramirez & Oakley 2015). Thus, LACE seems to rely on the same r-opsin-based phototransduction cascade found in octopus eyes (Ramirez & Oakley 2015). Additionally, LACE behaviour is maximally sensitive to blue light, which closely matches the known spectral sensitivity of opsin from octopus eyes (Ramirez & Oakley 2015). *O. bimaculoides* LACE shows that the latency for LACE is shortest between 470 and 480 nm (Ramirez & Oakley 2015). The predicted λ_{\max} , the

wavelength of maximum absorption in the spectrum, for octopus LACE is 480 nm, and the λ_{max} for octopus eye opsin is 474 nm (Ramirez & Oakley 2015). Since photoreceptors in the skin and eyes peak close to the same wavelengths, the same opsin phototransduction may underlie LACE (Ramirez & Oakley 2015, Godfrey-Smith 2016). Thus, a common molecular mechanism for light detection in eyes may have been co-opted for light sensing in octopus skin and used for LACE (Ramirez & Oakley 2015).

Dispersed light sensitivity in cephalopod skin raises questions of how it evolved to underlie LACE, how the skin senses light, and to what extent this ability contributes to rapid changes in the skin's color and tone (Ramirez & Oakley 2015). Presence of light-sensitive skin in cephalopods indicates that chromatophore responses from LACE may play a role in modulating body patterning for camouflage, in addition to control exerted by the central nervous system (Ramirez & Oakley 2015). Moreover, the widespread presence of distributed light-sensing in molluscs, appearing in groups such as bivalves, gastropods and chitons, suggests that it could be an ancestral molluscan trait co-opted in the cephalopod lineage to mediate body-patterning in response to light (Ramirez & Oakley 2015). The evolutionary history of dispersed light sensing throughout the phylum may be clarified with a better understanding of the mechanisms that govern it (Ramirez & Oakley 2015).

Metabolic Cost of Chromatophores

Metabolism can be generally defined as the sum of the physicochemical processes within an organism that transform energy and materials into living structures and activities (Glazier 2014). The rate of metabolism is most often estimated by the rate of

oxygen consumption since oxidative phosphorylation generates ATP in aerobic organisms (Glazier 2014, Bayne & Newell 1983). Understanding biological processes requires knowledge of metabolic rates to discern how organisms function by allocating energy in environmentally responsive and evolutionarily adaptive ways (Glazier 2014).

Research regarding respiration can help increase understanding of the flow of energy through individual organisms and populations (Bayne & Newell 1983).

Cephalopods must strategically allocate their resources to the energetic demands of various biological processes; thus, there is potential for a tradeoff since the organism has a limited supply of energy to fuel growth, reproduction, maintenance, and activity (Wells & Clarke 1996). Although octopus metabolic rates are high by invertebrate standards, and they are a comparatively successful group, understanding their energetics could reveal possible constraints on what a cephalopod can do (Wells et al. 1983, Wells & Clarke 1996).

It is widely assumed that colour change carries some metabolic costs (Duarte et al. 2017, Rodgers et al. 2013). The anatomy of chromatophores may indicate a metabolic cost since each chromatophore has 15–25 radial muscle fibers that are innervated by neurons; both muscle and neurons are energetically expensive to operate (Messenger 2001). When studying octopus crypsis in 1999, Hanlon et al. remarked “it must be neurophysiologically expensive to operate those hundreds of thousands of chromatophores in synchrony with visual input, and to do so continually when they are out of the den” (Hanlon et al. 1999). Investigating this mechanism may reveal general principles of how camouflage works, allow for better understanding of how it impacts other processes and animal energy budgets, and potentially give insight into octopus

evolutionary history (Duarte et al. 2017).

Catch Muscle

Although the existence of metabolic costs due to color change may be assumed, it is possible that mechanisms exist to reduce or temporarily eliminate these costs. Catch is a mechanical contractile state that occurs in some invertebrate smooth muscles, in which high force is maintained as twitchin molecules tether between thin and thick filaments (Butler & Siegman 2010). This allows for low energy expenditure while maintaining force for many hours, allowing the muscle to resist stretch in oysters, scallops, clams, and mussels (Butler & Siegman 2010). It is possible that the radial muscles surrounding chromatophores operate using a similar mechanism to reduce energy expenditure for camouflage. Additionally, papillary muscles sustain a “catch-like” tension in the absence of neural stimulation in cuttlefish, which seems to reduce the energy expenditure required for long-term camouflage (Gonzalez-Bellido et al. 2018).

Objectives and Hypothesis

This is the first known research to directly measure the energetic demand of color change in any animal. Since cephalopods exhibit unparalleled camouflage abilities, it is important to understand the metabolic demand of chromatophore use in order to understand camouflage itself. Thus, the purpose of this research was to determine the metabolic cost of chromatophore use in a common octopus species in the northeastern Pacific Ocean, *Octopus rubescens*. I measured the difference in oxygen uptake between periods of chromatophore expansion and contraction from excised octopus skin by using

LACE; photographs were used to determine the areas of chromatophores when expanded and contracted, which were applied to measurements of *O. rubescens* surface area to determine the area occupied by chromatophores on a whole octopus. These data were compared to the routine metabolic rate of *O. rubescens* taken during periods of little chromatophore use, which served to contextualize the magnitude of the demand of the chromatophore system to the overall energy demand of *O. rubescens*. I hypothesized that there would be no change in oxygen uptake between chromatophore expansion and contraction, which could be explained by a catch-like mechanism present in chromatophore radial muscles which would nearly eliminate the system's energy expenditure during camouflage.

METHODS

Octopus Collection and Housing

From June to August 2020, 12 male and 5 female *Octopus rubescens* ranging in size 26.5 to 390 grams were collected by SCUBA from Driftwood Beach, Whidbey Island, Washington (Table 1). The octopuses were found in discarded glass bottles on the ocean floor, which were placed in sealed plastic bags and brought to the surface. Each octopus was transferred out of its original glass bottle and into a red plastic bottle, then placed in a large cooler filled with oxygenated seawater that was transported to Rosario Beach Marine Lab (RBML) in Anacortes, Washington.

The mass and sex of each octopus was recorded at RBML. Each octopus was housed in an individual 27.3 L (30.8 cm x 24.5 cm x 36.2 cm) clear plastic tank with flowing seawater at approximately 13°C. Octopuses were kept with an approximately natural day:night cycle and fed one purple shore crab (*Hemigrapsus nudus*) each day. Octopuses were observed daily for signs of illness such as lack of appetite or movement, and neither senescent nor egg-brooding octopuses were used in experimentation. After skin was sampled in late August, the octopuses were returned to the location at which they were collected.

Microrespirometry

Design

A custom respirometer was built by sandwiching a 3D printed panel, made using polylactic acid (PLA), between two glass microscope slides (Fig. 1 & 2). The middle panel was layered with silicone high vacuum grease on its top and bottom to seal the

respirometer. The panel had a hole in the middle that acted as the main chamber for respirometry, which was connected by two channels to a shallow circular area that contained a stir bar on a spindle. The stir bar was 3D printed with metal bars embedded in each arm. Shallow channels were added lengthwise to the respirometer to hinder the introduction of bubbles. All parts were assembled underwater in a small tank of oxygenated artificial seawater made using Instant Ocean. This allowed the respirometer to entirely fill with seawater and keep bubbles excluded as it was assembled. The respirometer was held together by custom 3D printed rectangular retainers across the width of each side.

For experimentation, the small respirometer was placed on a custom 3D printed platform (Fig. 3). A neodymium bar magnet attached to a DC-motor was located below the stir bar chamber of the respirometer to constantly spin the stir bar during experimentation. Above the stir bar chamber, a Peltier thermo-electric cooler was placed so that the cooling side was in direct contact with the glass of the respirometer. The heating side of the Peltier thermo-electric cooler was attached with thermal conductive paste to an aluminum water cooled block heatsink, which had plastic tubing attached that pumped water up through the heatsink from a bucket on the floor. The heatsink was glued into a 3D printed compartment attached by a hinge mechanism to the 3D printed platform, allowing the heatsink and Peltier to be lifted when the respirometer was added then placed down on top of it.

A thin film platinum resistance temperature detector (PT-100 RTD) was placed underneath the respirometer so that it was in direct contact with the glass of the main chamber. A FireSting O₂ oxygen sensor was connected to the platform that held the

respirometer to determine oxygen within the respirometer with a small piece of oxygen sensor foil that was glued to the inside of the main chamber.

Above the respirometer setup, a panel of six red LEDs were angled towards the main chamber of the respirometer and taped into place. A larger array of 49 blue LEDs each rated at a luminous intensity of 19,380 millicandela (mcd) were adhered to a convex magnifying lens which was angled towards the main chamber of the respirometer on a stand.

The respirometry components were set up underneath a dissecting microscope and the platform was taped into place with duct tape (Fig. 4). A digital SLR camera was attached to the microscope.

Blank Testing

To determine the temperature sensitivity of the oxygen sensor, oxygen concentration was measured in an empty respirometer in which the temperature was stepped from 20.5°C to 16.5°C in 0.1°C steps and apparent change in oxygen concentration was related to temperature. To investigate the oxygen seep (the amount of oxygen flowing into the respirometer from outside of it) of the custom respirometers used in experimentation, the respirometer was sealed containing seawater with a low oxygen content, created by bubbling 100% nitrogen gas, rather than oxygen, into the seawater before respirometer assembly. Oxygen concentration was measured for 7 hours to observe the rate of oxygen seep.

Skin Testing

Before skin sample data were taken, the equipment was calibrated using a respirometer that contained only 100% oxygenated seawater and no octopus skin. For each trial, one piece of skin was excised from one octopus at a time. To anesthetize each octopus before skin excision, the octopus was removed from its seawater tank and placed in a 2L beaker filled with 1 L of oxygenated seawater, and 22.5 mL 95% ethanol was gradually added. Once unresponsive, the octopus was removed from the beaker and placed onto a clean table. Immediately, forceps were used to grasp the top layer of the skin at the base of an arm and pull it up and away from the underlying layers. Dissecting scissors were used to cut an approximately 4 mm by 4 mm piece of skin. The octopus was returned to well-oxygenated seawater for recovery. The skin removal procedure took less than ten minutes in total. After recovery, each octopus was returned to its normal laboratory seawater tank.

Once removed, the piece of excised skin was adhered to a microscope slide using Loctite Ultra Gel Control high strength gel cyanoacrylate. The microscope slide had a piece of oxygen sensor foil glued to it prior to skin excision and was assembled as the bottom slide of the respirometer.

The respirometer containing the skin was immediately placed in the experimental setup as previously described. The setup was located in a small room with no windows, and the door was shut after placement of the respirometer so that it was completely dark. Power was turned on to an Arduino Uno which turned on the DC motor controlling the stir bar, the Peltier thermoelectric device to keep the water temperature within the respirometer consistent, the temperature sensor measurements, and the lights. At

approximately the same time recording oxygen concentration in the respirometer was started using the Pyroscience FireStingO2 oxygen optode system. After a one-minute period of red light, the lights alternated between blue in addition to red, and only red, each for five-minute periods. Thus, the presence of constant red light served as a control. Photographs of the excised skin were taken each minute for two intervals of blue light and the intervening period of only red light. Temperature and oxygen data were continuously automatically recorded.

Photograph Analysis

The photographs were taken of the excised skin and used to measure total area of skin and area of chromatophore coverage under different light regimes using ImageJ as described in Appendix A.

3D Scanning

Two octopuses used in this experiment were preserved in formalin and used to determine the relationship between octopus surface area and wet mass (Table 1). Wet masses of the octopuses were taken while alive. Octopuses were then placed into 2.5% ethanol in seawater to be sedated, then ethanol concentration was raised to 10% to euthanize octopuses. Octopuses were then fixed in 12% formalin. One at a time, each arm of these preserved octopuses was removed and scanned using an EinScan desktop 3D scanner. Once all of the arms were removed, the remaining body of each octopus was scanned. Scans were processed using the methods as described in Appendix B. An estimate of the surface area for each octopus was determined.

Data Analysis and Statistics

Empirical Determination of Oxygen Sensor Temperature Sensitivity

Using the oxygen concentration data collected while decreasing temperature, the relationship between deviation from 19°C and the change in measured oxygen concentration was determined. The oxygen concentration changed approximately 7.3 $\mu\text{mol O}_2$ per degree over the range of testing. See Appendix C: Empirical temperature response determination.

Metabolic Rate Data Analysis

Since the temperature and oxygen data were measured with two separate instruments, the recorded data started at slightly different times and thus, required alignment. Furthermore, oxygen pressure, rather than concentration, was directly measured. Oxygen pressure increases as temperature increases. The temperature was set to 19°C; however, in reality, it varied slightly which caused the oxygen sensor to respond to changes in the oxygen pressure. Since oxygen measurement by the optode system was temperature-sensitive, and these two factors fluctuated up and down in synchrony, they both peaked at the same time and the peaks were used to align the data.

The difference between temperature and oxygen data was plotted to determine the correct offset between these datasets (Fig. 5). The more accurate alignment was indicated by stable lines, and less accurate alignment was indicated by bigger waves. Plots were produced at one-second interval offsets from -6 seconds, in which the oxygen sensor started before the Arduino by 6 seconds, to 6 seconds, in which the oxygen sensor started

after the Arduino by 6 seconds. I visually determined which offset produced the fewest waves and thus, the most accurate alignment. The temperature was then corrected by using the empirically determined temperature correction factor (Appendix C).

In addition to correcting for temperature, oxygen measured had to be corrected for background light. Because we used an optical light measurement, the blue LEDs could cause changes in the oxygen measurements. The Pyroscience FireStingO2 measures fluorescence at 710 nm, while the blue LEDs used to illuminate the skin had maximum luminance intensity at 470 nm, so little overlap likely occurred between measurement wavelengths and illumination wavelengths. However, tests with the respirometer setup demonstrated there was enough overlap to cause some small changes to oxygen readings. When the respirometer was placed into the setup for each trial, the lights may have shifted slightly, making their distance from the oxygen sensor and therefore the intensity on the optode differ from one trial to the next. This is evident in the ambient light measurements of the oxygen sensor. Brighter background light would have lower apparent oxygen measurements, and lower background light would have higher oxygen measurements.

The PyroScience FireStingO2 takes measurements of background light levels and reports these in raw millivolt values. Background light levels during periods when the blue light was off ranged from 4–7 mV, while background light levels during periods when the blue light was on ranged from 45–65 mV. Oxygen measurements during periods when the background light level was above a threshold of 20 millivolts (mV) were divided by a percentage of the total value from 92% to 100% in 1% increments (Appendix C, Fig. 6). Therefore, the number 1 would correspond to the raw data with no

background light correction (Fig. 6). The background light correction was individually determined for each trial by selecting the alignment in which the beginning and end of the oxygen concentrations during each light period were aligned with that of the previous and subsequent light period, respectively (Fig. 6). This allowed for the least difference in background light among the alternating periods of light.

For the data from each 5-minute period of light, 45 seconds was trimmed from the beginning and 15 seconds was trimmed from the end (Appendix C, Fig. 7). This was due to an adjustment period that exhibited unstable oxygen readings during the transitions between red and blue light periods and vice versa.

Next, the oxygen consumption in $\mu\text{mol O}_2 \text{ hr}^{-1}$ was calculated (Appendix C). The oxygen sensor reported oxygen concentration in $\mu\text{mol O}_2 \text{ L}^{-1}$ every two seconds. Each measurement of oxygen was subtracted from the next measurement, taken two seconds later, to determine the change in oxygen consumption in $\mu\text{mol O}_2 \text{ L}^{-1} 2\text{s}^{-1}$. This number was multiplied by 1800 (the number of two-second intervals in one hour) to give $\mu\text{mol O}_2 \text{ L}^{-1} \text{ hr}^{-1}$. This number was then multiplied by 0.000612 L (the volume of the respirometer) to determine $\mu\text{mol O}_2 \text{ hr}^{-1}$. These values were multiplied by 1000 to report values in $\text{nmol O}_2 \text{ hr}^{-1}$.

Next, the metabolic rate had to be corrected for oxygen level as there was a statistically significant positive relationship between oxygen concentration in the respirometer and measured oxygen consumption rate (Fig. 8, Appendix C). To do this the relationship between oxygen consumption rates and oxygen levels was determined using a linear regression (Fig. 8, Appendix C). Each oxygen consumption measurement was then corrected to a pO_2 of $240 \mu\text{mol O}_2 \text{ L}^{-1}$. To do this, residuals for each data point from

the linear regression were added to the prediction of metabolic rate at $240 \mu\text{mol O}_2 \text{ L}^{-1}$ (Appendix C).

Two of the trials included a small bubble in the respirometer that required correction as well. The diameter of each bubble was measured using FIJI (FIJI Is Just ImageJ) by calibrating the measurement tool using the known measurement of the temperature probe within the respirometer. The volume of each bubble was estimated from its diameter assuming that it was a sphere. The “correct bubble” function in the R package “respirometry” (Birk, 2020) was used to adjust to the realized volume of the respirometer (Appendix C).

After the bubble correction, the average oxygen consumption from the red light period, when chromatophores were more constricted, was subtracted from that of the average of the two blue light periods, when chromatophores were more expanded (Table 2). Data from three octopuses, 7, 14 and 15, in which the chromatophore area decreased during blue light periods were removed since they did not exhibit LACE (Table 1). Finally the change in oxygen consumption rate was divided by the change in chromatophore coverage area to arrive at the oxygen consumption per area chromatophore coverage (Table 3).

Oxygen was measured at 19°C since this was near room temperature. In trials, when the peltier was set to cool the water below room temperature, condensation formed on the respirometer. The condensation required periodic removal by being wiped off, which could slightly alter the position of the respirometer. For more accurate data and consistent positioning of the respirometer, it was maintained at room temperature for all trials. Thus, metabolic rates were adjusted from 19°C to 11°C , the temperature at which

previous whole-organism metabolic rates were measured, assuming a Q_{10} of 2 (Table 3, Appendix C).

A Shapiro-Wilk test was performed to test the normality of the data, and because the assumption of normality was not supported, a Wilcoxon signed rank exact test was then performed to determine whether the increase in metabolic rate under blue light was greater than 0. I determined the median metabolic rate estimate in $\text{nmol O}_2/\text{mm}^2/\text{hr}$ and its 95% confidence interval using bootstrap resampling (Figure 11).

Calculating Chromatophore Metabolic Rate of Whole Octopus by Mass

The relationship between octopus mass and surface area was determined by using the previously described surface area data determined from 3D scans of preserved octopuses. The nls (nonlinear least squares) function in R was used to fit the following equation to the data: $\text{surface area} = M \times \text{mass}^t$, in which the function fits M coefficient and t power.

For a range of masses from 25g to 400g, the metabolic rate during maximal chromatophore usage was estimated. For each mass, the surface area was predicted from the previous function and divided in half to conservatively account for lower chromatophore density on an octopus's ventral side. This surface area was multiplied by the mean chromatophore metabolic rate per area, and the upper and lower 95% confidence interval.

Chromatophore Metabolic Rate Comparison to Whole Octopus Routine Metabolic Rates

Routine metabolic rate data previously measured in 97 *Octopus rubescens* at 11°C by Kirt Onthank was used for comparison to chromatophore system metabolic rate estimates. Five additional metabolic rate measurements of *Octopus rubescens* paralarvae were added from Seibel 1997. These five measurements were corrected to 11°C assuming a Q_{10} of 2 (Appendix C). The scaling exponent among these previous measurements was determined using a linear regression of the log-log transformed data. This was then used in conjunction with the nls() function to determine the relationship between metabolic rate and octopus mass (Fig. 14).

RESULTS

Microrespirometry

Blank Testing

Empirical determination of the oxygen sensor temperature sensitivity resulted in the determination of the relationship between deviation from 19°C and the change in measured oxygen concentration (Figure 9). The relationship between temperature and oxygen adjustment multiplier was found to be:

$$M = 3.22 \times 10^{-6} T^{3.68} + 0.834$$

Where M is the oxygen adjustment multiplier which would be multiplied by the measured oxygen to correct for temperature and T is the temperature in degrees Celsius.

The oxygen seep of the respirometer was determined (Figure 10). In the range of oxygen levels at which respirometry occurred in this experiment, from 164 to 253 $\mu\text{mol O}_2 \text{ L}^{-1}$, the oxygen seep was at a maximum of 0.2 $\mu\text{mol O}_2 \text{ L}^{-1} \text{ min}^{-1}$. This is not substantial compared to the metabolic rates recorded in this study.

Metabolic Rate Determination of Excised Skin

Metabolic rate of excised octopus skin samples was determined by respirometry while illuminated with blue light, and therefore chromatophores activated, and with red light, and therefore chromatophores mostly inactivated. There was a significant relationship between metabolic rates and oxygen levels at which they were collected (Figure 8, linear regression, $p = 0.0106$). Metabolic rates were therefore corrected based on this linear relationship to an oxygen concentration of 240 $\mu\text{mol O}_2 \text{ L}^{-1}$.

The difference in metabolic rate of the skin between periods of chromatophore activation and chromatophore inactivation was significantly greater than 0 (Appendix C, Wilcoxon signed rank exact test, $p = 0.02768$). The mean value of oxygen required by chromatophore use is 22.5 nmol O₂ per mm² (95% CI 5.0 – 46.8 nmol O₂ per mm²). The median value of oxygen required by chromatophore use is 9.9 nmol O₂ per mm² (95% CI 3.45 – 18.05 nmol O₂ per mm², Figure 11).

There was no apparent relationship between octopus mass and metabolic rate per mm² of skin (Figure 12; linear regression, $p = 0.8881$).

3D Scanning

Estimates of surface areas were determined for octopuses that were 33 and 377 grams. The surface areas were 14,826 and 96,929 mm², respectively. The relationship between *Octopus rubescens* surface area and mass was fit to the equation:

$$\text{Surface area} = 1,001 (\text{mass}^{0.77})$$

Where surface area is expressed in mm². This relationship makes it possible to estimate the surface area based on a given mass of *Octopus rubescens* (Figure 13).

Chromatophore Metabolic Rate Comparison to Whole Octopus Routine Metabolic Rates

The relationship between whole octopus routine metabolic rate and octopus mass was found to be:

$$\text{Routine metabolic rate} = 7.433 (\text{mass}^{0.751})$$

Where octopus mass is in grams and routine metabolic rate is expressed in $\mu\text{mol O}_2 \text{g}^{-1} \text{h}^{-1}$. As octopus mass increases, so does routine metabolic rate; mass-specific metabolic rate decreases (Figure 14).

The relationship between estimated chromatophore metabolic rate and routine metabolic rate was determined (Figure 15). The estimated chromatophore metabolic rate is substantial when compared to the whole octopus routine metabolic rate.

DISCUSSION

This study is the first to estimate the metabolic demand of the octopus chromatophore system, and is also the first to directly measure the metabolic demand of color change in any animal. The results indicate that the chromatophore system has a high energy demand, which may imply a high energetic cost to octopus camouflage. Due to the involvement of the nervous and muscular systems involved, it is likely that cephalopod color change is one of the most energetically expensive forms of color change, so this estimate may represent the upper bound of the cost of color change in the animal kingdom.

The difference in chromatophore metabolic rate under blue light, as opposed to red light, was greater than 0. This, in conjunction with the photographs of chromatophores and measurements of their area, signifies that energy was being consumed due to light-activated chromatophore expansion (LACE). Furthermore, this study suggested that the estimated chromatophore metabolic rate is nearly as great as the whole octopus routine metabolic rate. Thus, it seems that chromatophore use in *O. rubescens* is energetically expensive and may necessitate a substantial part of this species' energy budget. This suggests a high energetic cost to camouflage, since camouflage is primarily controlled by the chromatophore system.

This study determined that there is no relationship between *O. rubescens* mass and metabolic rate per mm² of skin. This indicates that metabolic rate of chromatophore use per unit area of skin is consistent across the sizes of octopuses used in this study (26 g – 390 g). In future research, it would be helpful to verify this finding using a broader

range of masses for this species, especially as juveniles.

Additional Considerations

This research puts forth a conservative estimate of metabolic rate due to chromatophore expansion; the true metabolic rate is likely higher since most of the corrections and methods used confer an underestimate. For example, the 3D scanning methods determined an underestimate of octopus surface area, which can be seen in the 3D scans of each arm- not all of their detail or texture is displayed in the scans, and the arms of the preserved octopuses were often curled. In addition, the octopuses scanned in this study were fixed in formalin, which pulls the skin taut over the body, and would also give an underestimation of surface area. In future research, it may be more accurate to determine octopus surface area using computed tomography (CT) scanning on whole preserved octopuses rather than using a 3D scanner on parts of preserved octopuses.

The oxygen seep of the custom respirometers used in this study was low. Additionally, the oxygen seep was corrected for by comparing the difference in the periods of blue light to the intervening period of red light. By taking averages from both periods of blue light, the oxygen seep would be the same as the period of red light between them. Furthermore, during periods of blue light, the blue lights were turned on in addition to the red lights to avoid any underlying effects between exposure to the two different colors.

Future Research

This study gives potential insight into cephalopod energy budgets since it suggests high energy expenditure, and thus high priority, in chromatophore use for color change in *O. rubescens*. However, this research was performed using excised skin *in vitro*, so it is a possibility that the operation of the chromatophore system, and thus its metabolic rate, may differ *in vivo*.

This is the first known study to directly measure the energy demand of color change for any animal. As such, it is also the first to investigate the metabolic cost of chromatophore use, so there are many different avenues that could be explored in future research. Further studies could use LACE to investigate the metabolic rate of the chromatophore system with other species of octopus or other cephalopods to determine whether these findings are applicable to other species. Further research could also investigate the energy consumed by chromatophores on a whole octopus *in vivo*, which would include central nervous system input. Though this would provide valuable knowledge, it would be challenging since it is difficult both to determine a method to encourage consistent camouflage and to isolate color change as the sole varying factor in live organisms.

Although the underlying mechanism of chromatophores is known, it is unknown how octopuses instantly determine which colors and textures to exhibit, so this arena is rich with opportunities for future research to contribute to knowledge of rapid camouflage and the chromatophore system. Chromatophore expansion seems to happen in waves among chromatophores, so it could be possible that chromatophores communicate with each other to signal or trigger expansion in some way. Additionally,

further investigation could determine the energy consumption of each individual chromatophore and whether a chromatophore's size or color impacts its energy consumption. Future studies could also investigate whether chromatophore expansion and contraction are affected by ocean acidification or other seawater conditions.

ACKNOWLEDGEMENTS

“If we are to accomplish anything in this world, it will in equal measure be due to the work and achievements of others” - Nelson Mandela

I would like to thank everyone who has generously contributed their encouragement, time, and expertise to this research. First and foremost, I would like to thank my major professor, Dr. Kirt Onthank, for providing continual encouragement, support, and expertise and for making it not only possible but also enjoyable to conduct research during a pandemic. I would like to thank the members of my committee- Dr. David Cowles, Dr. C.J. Brothers, and Dr. Roy Campbell- for their valuable guidance throughout the development of this project. I would like to thank those who assisted during my research season, especially Dr. Jim Nestler, Dr. Alan Verde, Jessica D’Auria, and Jaydee Sereewit, for their assistance with octopus collection dives and data collection. I would like to thank Dr. Lloyd Trueblood for the valuable insight that he contributed to this research, beginning at its earliest stages. I would like to thank Stephanie Marsh and the Biology faculty for being positive role models and for relentlessly encouraging my growth as a scientist during my time at Walla Walla University. Finally, I am grateful to Walla Walla University and Rosario Beach Marine Laboratory for providing not only funding and a research facility, but also an environment that challenged me to grow personally, academically, and professionally.

REFERENCES

- Bayne, B. L., & Newell, R. C. (1983). *Physiological Energetics of Marine Molluscs*. The Mollusca, 407–515. doi:10.1016/b978-0-12-751404-8.50017-7
- Birk, Matthew A. (2020). *Respirometry: Tools for Conducting and Analyzing Respirometry Experiments*. R package version 1.2.1. <https://CRAN.R-project.org/package=respirometry>
- Butler, T. M., & Siegman, M. J. (2010). *Mechanism of Catch Force: Tethering of Thick and Thin Filaments by Twitchin*. Journal of Biomedicine and Biotechnology, 2010, 1–20. doi:10.1155/2010/725207
- Duarte, R. C., Flores, A. A. V., & Stevens, M. (2017). *Camouflage through colour change: mechanisms, adaptive value and ecological significance*. Philosophical Transactions of the Royal Society B: Biological Sciences, 372(1724), 20160342. doi:10.1098/rstb.2016.0342
- Glazier, D. S. (2014). *Is metabolic rate a universal “pacemaker” for biological processes?* Biological Reviews, 90(2), 377–407. doi:10.1111/brv.12115
- Godfrey-Smith, Peter. *Other Minds: The Octopus, the Sea, and the Deep Origins of Consciousness*. Farrar, Straus and Giroux, 2016.
- Gonzalez-Bellido, P. T., Scaros, A. T., Hanlon, R. T., & Wardill, T. J. (2018). *Neural Control of Dynamic 3-Dimensional Skin Papillae for Cuttlefish Camouflage*. iScience, 1, 24–34. doi:10.1016/j.isci.2018.01.001
- Hadjisolomou, S. P., & El-Haddad, G. (2017). *SpotMetrics: An Open-Source Image-Analysis Software Plugin for Automatic Chromatophore Detection and Measurement*. Frontiers in Physiology, 8. doi:10.3389/fphys.2017.00106
- Hanlon, R., Forsythe, J., & Joneschild, D. (1999). *Crypsis, conspicuousness, mimicry and polyphenism as antipredator defences of foraging octopuses on Indo-Pacific coral Reefs, with a method of quantifying crypsis from video tapes*. Biological Journal of the Linnean Society, 66.
- Hanlon, R. (2007). *Cephalopod dynamic camouflage*. Current Biology, 17(11), R400–R404. doi:10.1016/j.cub.2007.03.034
- Hanlon, Roger. (2011). *Camouflage and Signaling in Cephalopods*. iBiology.
- Kingston, A. C. N., Kuzirian, A. M., Hanlon, R. T., & Cronin, T. W. (2015). *Visual phototransduction components in cephalopod chromatophores suggest dermal photoreception*. Journal of Experimental Biology, 218(10), 1596–1602. doi:10.1242/jeb.117945

- Mather, J.A., Anderson, R.C., Wood, J.B. (2010). "Chapter 6: Appearances." *Octopus: The Ocean's Intelligent Invertebrate*. Timber Press, Inc., pp.89-97.
- Mathger, Lydia M, et al. "Rapid Adaptive Camouflage in Cephalopods." *Animal Camouflage: Mechanisms and Function*, edited by Martin Stevens and Sami Merilaita, by Roger T. Hanlon and Chuan-Chin C. Chiao, Cambridge University, 2011, pp. 145–152.
- Messenger, J. B. (2001). *Cephalopod chromatophores: neurobiology and natural history*. *Biological Reviews*, 76(4), 473–528. doi:10.1017/s1464793101005772
- Ramirez, M. D., & Oakley, T. H. (2015). *Eye-independent, light-activated chromatophore expansion (LACE) and expression of phototransduction genes in the skin of Octopus bimaculoides*. *Journal of Experimental Biology*, 218(10), 1513–1520. doi:10.1242/jeb.110908
- Rodgers, G. M., Gladman, N. W., Corless, H. F., & Morrell, L. J. (2013). *Costs of colour change in fish: food intake and behavioural decisions*. *Journal of Experimental Biology*, 216(14), 2760–2767. doi:10.1242/jeb.080879
- Seibel, B. A., Thuesen, E. V., Childress, J. J., & Gorodezky, L. A. (1997). *Decline in Pelagic Cephalopod Metabolism With Habitat Depth Reflects Differences in Locomotory Efficiency*. *The Biological Bulletin*, 192(2), 262–278. doi:10.2307/1542720
- Stuart-Fox, D., Moussalli, A. (2008). *Camouflage, communication and thermoregulation: lessons from colour changing organisms*. *Philosophical Transactions of the Royal Society B: Biological Sciences*. <http://doi.org/10.1098/rstb.2008.0254>
- Wells, M.J., Clarke, A. (1996). *Energetics: The Cost of Living and Reproducing for an Individual Cephalopod*. The Royal Society. 1083-1104.
- Wells, M. J., O'Dor, R. K., Mangold, K., & Wells, J. (1983). *Diurnal changes in activity and metabolic rate in octopus vulgaris*. *Marine Behaviour and Physiology*, 9(4), 275–287. doi:10.1080/10236248309378598
- Wells, M. J., O'Dor, R. K., Mangold, K., & Wells, J. (1983). *Feeding and metabolic rate in octopus*. *Marine Behaviour and Physiology*, 9(4), 305–317. doi:10.1080/10236248309378600

TABLES & FIGURES

Octopus	Octopus in R	Sex	Mass (g)	Preservation
S1r	1	male	136	
T2r	2	male	153	
U4r	3	female	59	
J5	4	female	85	
K6	5	male	155	
L7	6	male	46	
W9r	7	female	41	
X10r	8	male	188	
M11	9	male	152	
Y13r	10	male	218	
Z14r	11	male	85	
N15	12	male	390	
O16	13	male	189	
23	14	female	33	Preserved
R25	16	female	26.5	
V26	17	male	308	
N/A	N/A	male	377	Preserved

Table 1. Information on each *Octopus rubescens* used in experimentation. Letters indicate the order in which skin samples were taken, Z being the last. Numbers in the “Octopus” column are the individual numbers used for the identification and records of each octopus. A small r indicates a skin sample that was redone after the first attempt. The two preserved octopuses were later used in the determination of the surface area and mass relationship for *Octopus rubescens*.

Octopus #	Oxygen Consumption ($\mu\text{molO}_2 \text{ hr}^{-1}$)				Mean Oxygen Level ($\mu\text{molO}_2 \text{ L}^{-1}$)			Mean Temperature ($^{\circ}\text{C}$)		
	Blue 1	Red 1	Blue 2	Difference	Blue 1	Red 1	Blue 2	Blue 1	Red 1	Blue 2
1	78.63	28.00	58.19	40.41	166.26	165.05	164.31	19.2	19.1	19.2
2	98.09	5.26	84.14	85.85	188.71	190.78	187.14	19.2	19.1	19.2
3	40.17	7.36	26.16	25.80	213.18	211.57	211.72	19.2	19.1	19.2
4	83.18	124.66	103.99	-31.08	229.66	226.34	216.18	19.2	18.7	19.2
5	164.87	119.59	140.36	33.03	226.79	215.88	201.79	19.1	18.8	19.1
6	68.57	56.93	44.71	-0.28	228.02	223.00	220.28	18.9	19.0	18.8
8	103.07	34.79	58.53	46.01	248.49	241.52	236.20	19.1	19.0	19.1
9	37.27	42.58	45.90	-1.00	230.93	228.25	224.42	19.2	19.1	19.2
10	99.65	39.29	36.81	28.94	253.14	244.57	241.27	19.1	19.0	19.1
11	71.96	12.04	37.06	42.47	239.81	235.14	233.42	19.2	19.1	19.2
12	72.89	59.91	56.24	4.66	218.44	211.13	207.52	19.2	19.0	19.1
13	87.68	39.19	64.04	36.66	214.91	208.39	204.54	19.2	19.1	19.1
16	101.72	67.25	69.11	18.17	224.74	217.47	213.77	19.2	19.1	19.1
17	75.58	39.16	73.50	35.38	203.38	199.82	196.55	19.2	19.1	19.2

Table 2. Oxygen consumption, mean oxygen level and mean temperature during respirometry of excised octopus skin. Levels are given for the first period of blue light stimulation (Blue 1) when chromatophores would be activated, the first period without blue light (Red 1) when chromatophores would not be activated, and the second blue light stimulation period (Blue 2). The difference between the mean of both blue light periods and the blue light off period is also given.

Octopus #	Mass (g)	Mean area blue (mm ²)	Mean area red (mm ²)	Difference mean area (mm ²)	Oxygen consumption (nmolO ₂ hr ⁻¹)	Oxygen consumption per area (nmolO ₂ (mm ²) ⁻¹ hr ⁻¹)	Adjusted oxygen consumption per area (nmolO ₂ (mm ²) ⁻¹ hr ⁻¹)
1	136	3.18	1.68	1.50	40.41	26.87	15.43
2	153	1.94	1.22	0.72	85.85	119.77	68.79
3	59	0.96	0.48	0.48	25.80	54.16	31.11
4	85	2.68	1.88	0.80	-31.08	-38.92	-22.36
5	155	1.49	1.36	0.12	33.03	265.81	152.67
6	46	1.28	0.82	0.46	-0.28	-0.62	-0.35
8	188	2.15	0.68	1.46	46.01	31.43	18.05
9	152	1.40	0.62	0.78	-1.00	-1.29	-0.74
10	218	2.60	0.45	2.15	28.94	13.47	7.74
11	85	2.71	0.14	2.57	42.47	16.56	9.51
12	390	4.02	1.59	2.43	4.66	1.92	1.10
13	189	3.50	1.41	2.09	36.66	17.53	10.07
16	26.5	0.93	0.32	0.62	18.17	29.47	16.92
17	308	3.32	0.52	2.80	35.38	12.64	7.26

Table 3. Mean area of skin covered by chromatophores during the blue periods (Mean area blue) and the non-blue period (Mean area red), and the difference of those means, which is the increase in chromatophore coverage between periods. In addition, oxygen consumption per increased area of chromatophore coverage is given, and oxygen consumption adjusted from 19°C to 11°C assuming a Q₁₀ of 2 to compare to previously measured routine metabolic rates taken at 11°C.

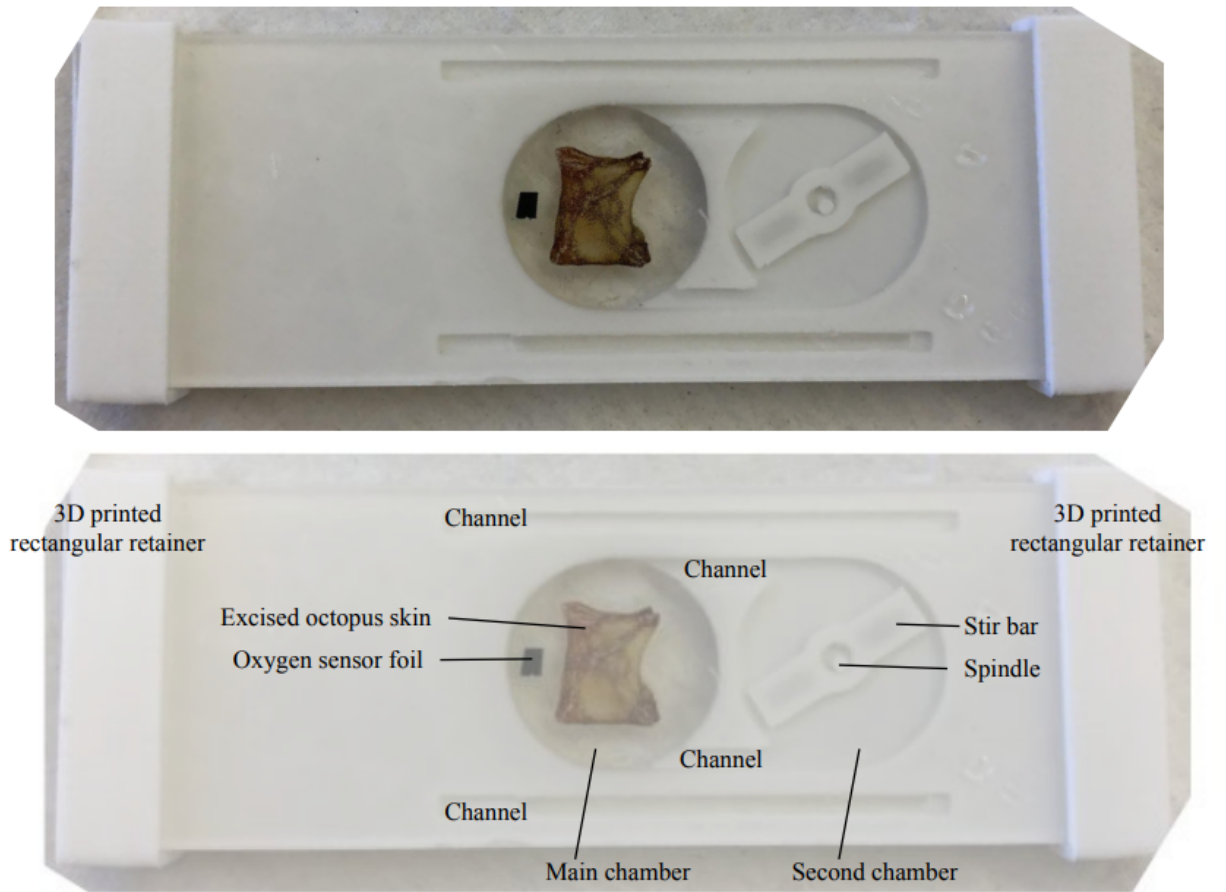


Figure 1. Assembled custom respirometer as viewed from above. The top image is a picture of the respirometer and in the bottom image its parts have been labeled.

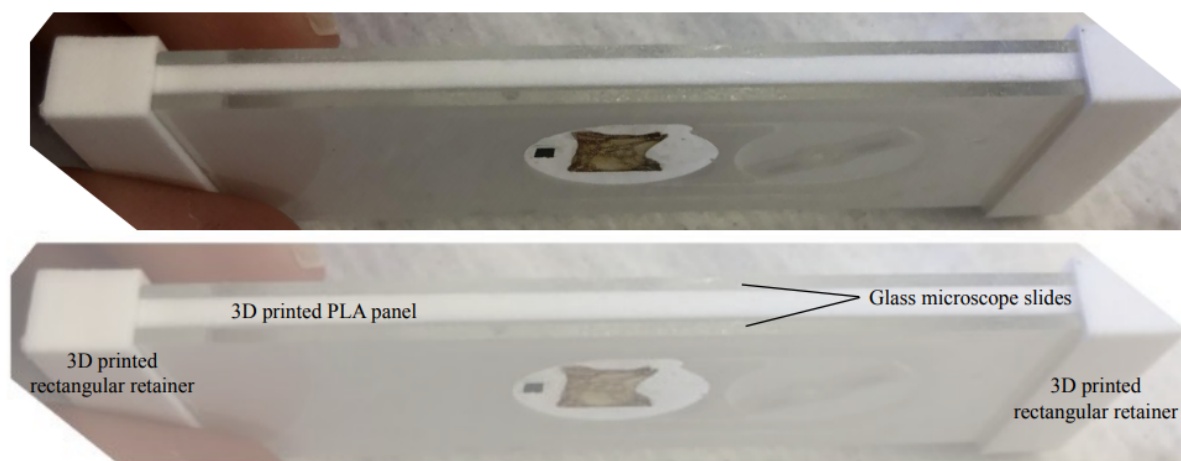


Figure 2. Assembled custom respirometer as viewed from the side. The top image is a picture of the respirometer and in the bottom image its parts have been labeled.

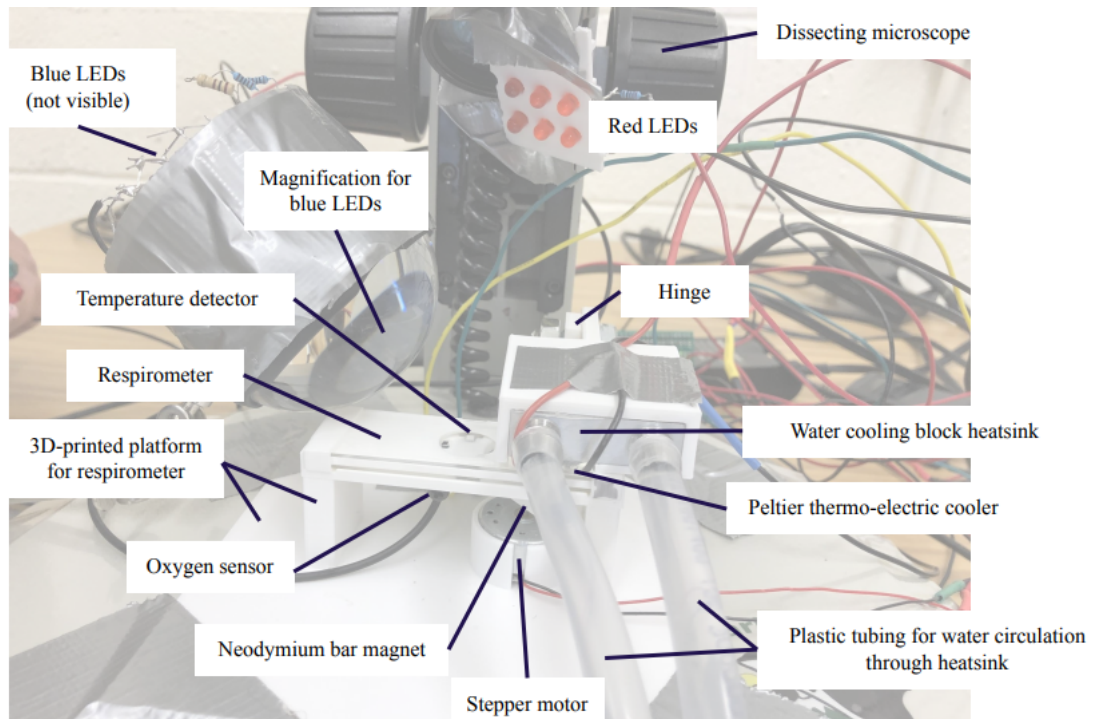
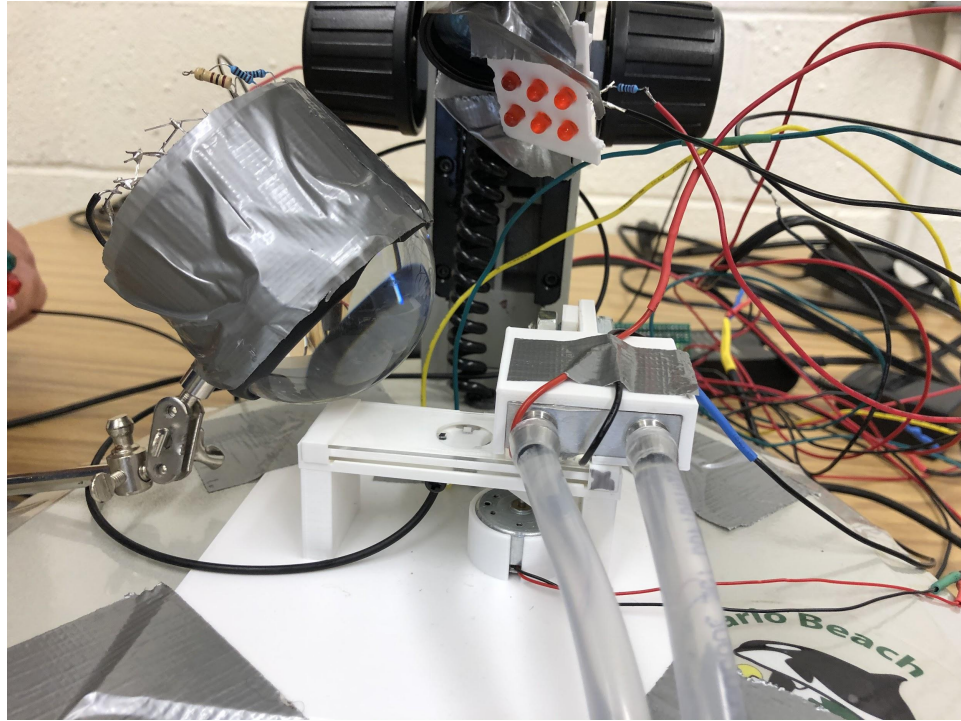


Figure 3. Placement of the respirometer during experimentation. The top image is a picture of the setup and in the bottom image its parts have been labeled.

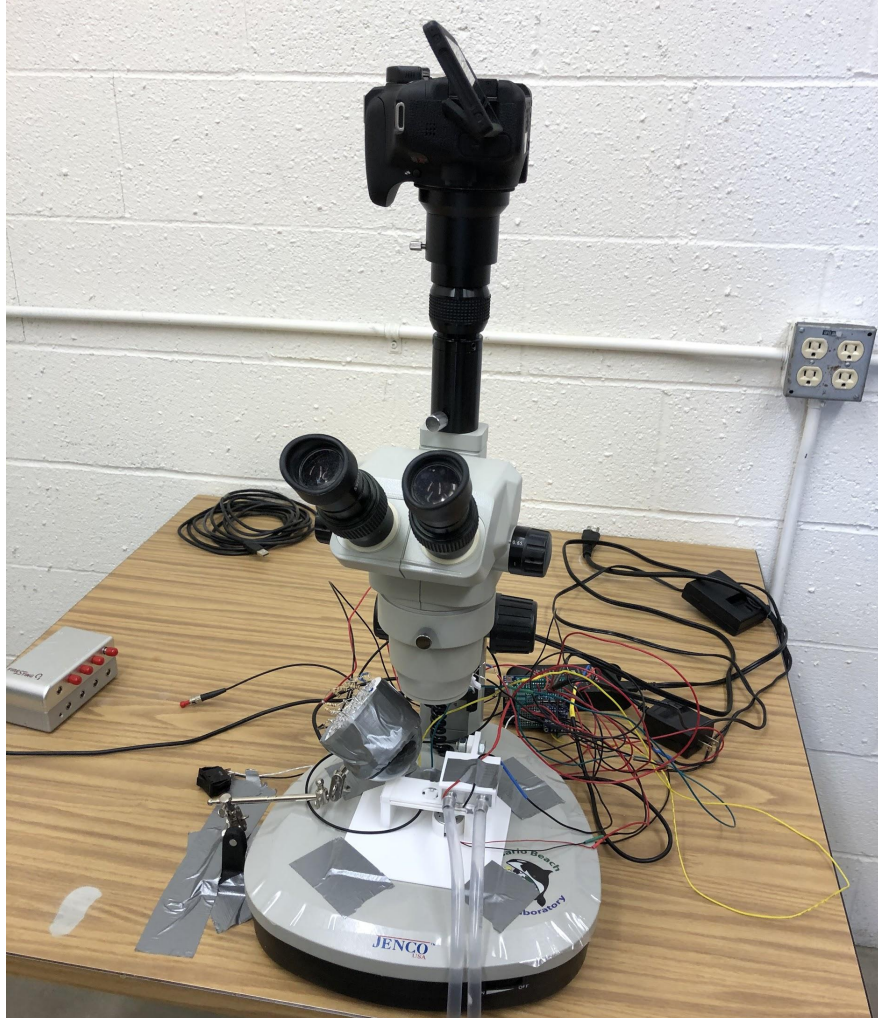


Figure 4. Main components of the experimental setup.

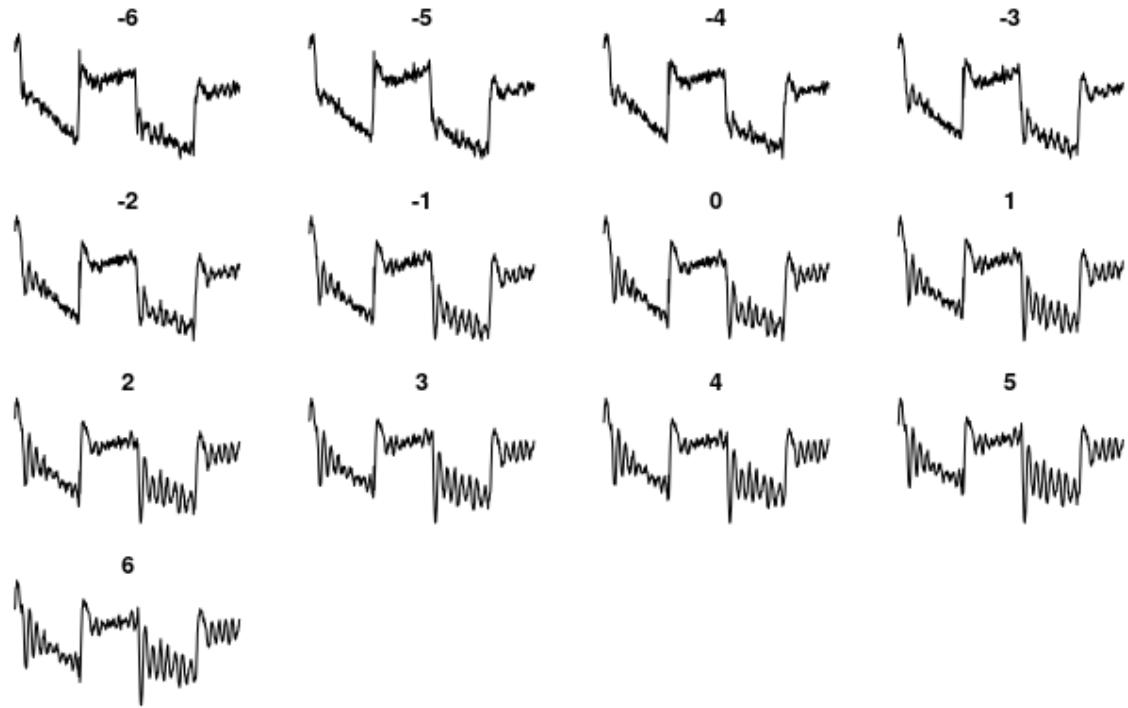


Figure 5. The difference between temperature and oxygen plotted for octopus 1r at various time offsets between the two datasets over the course of the trial. Better alignment is indicated by stable lines, and less accurate alignment is indicated by bigger waves. The correct time offset was determined by which second, from -6 to 6, showed the best alignment of temperature and apparent oxygen concentration. In this case, -4 was selected as the best time offset. This is an example of how the data was processed, and was used only for diagnostic purposes to identify correct alignment. As such, axis labels are not included; however, the difference between temperature and apparent oxygen concentration is plotted against time.

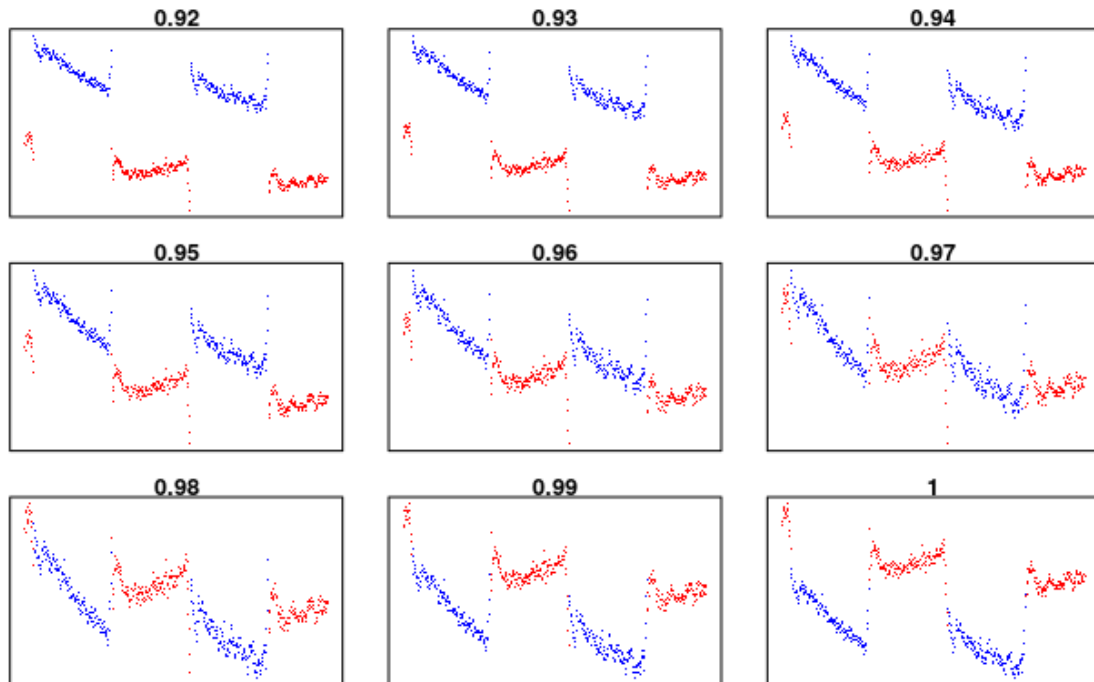


Figure 6. Oxygen concentration plotted versus time to determine the background light correction. Oxygen measurements during periods when background light exceeded 20 millivolts (blue light periods) for octopus 1r were divided by a percentage from 92% to 100% in 1% increments. 1 indicates the raw data with no background light correction. Blue points indicate data from the periods of blue light, and red points indicate data from the periods of red light. The alignment was selected in which the beginning and end of each light period were aligned with the previous and subsequent light period, respectively. In this case, the selected alignment was 0.97. This is an example of how the data was processed, and was used only for diagnostic purposes to identify correct alignment. As such, axis labels are not included; however, oxygen concentration is plotted against time.

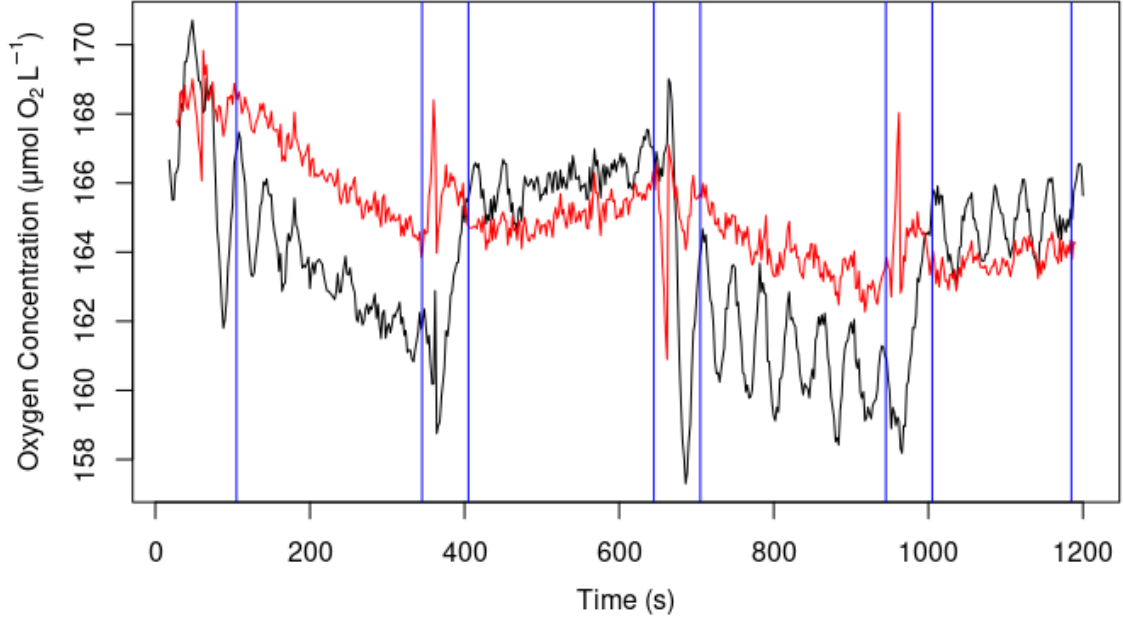


Figure 7. Data from octopus 1r. For each 5-minute period of light in each trial, 45 seconds was trimmed from the beginning and 15 seconds was trimmed from the end. The black line is uncorrected data while the red line represents the light and temperature corrected data. Vertical blue lines distinguish the trimmed time periods from the data used in analysis.

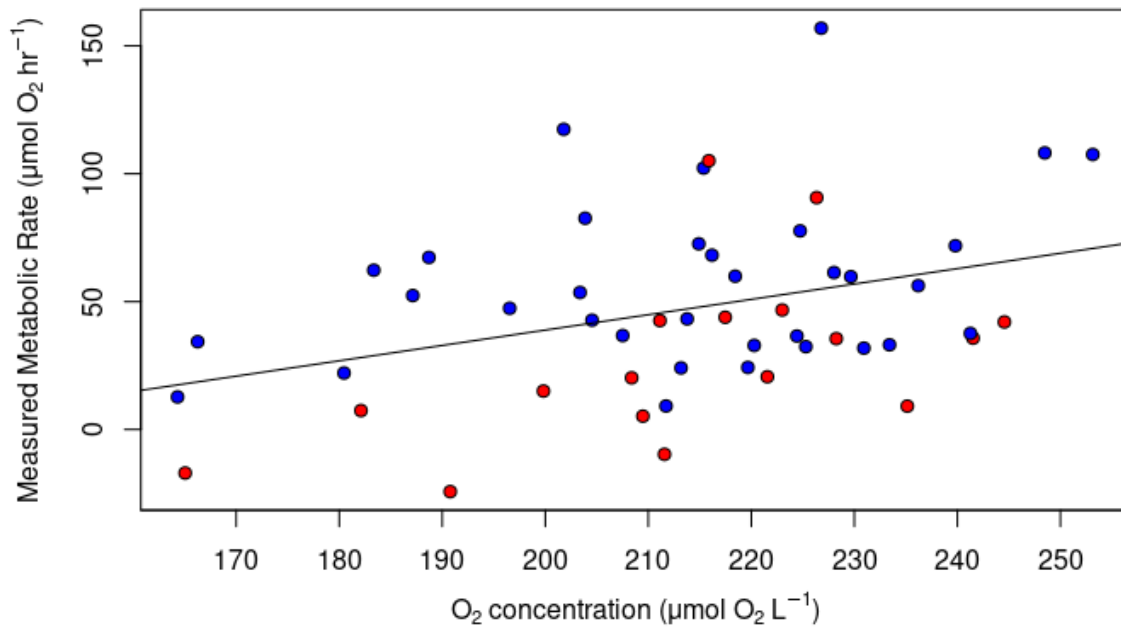


Figure 8. Relationship between metabolic rates and oxygen levels at which they were collected for both red and blue light periods of all skin sampled (linear regression, $p = 0.0106$). Red and blue data points represent data from red and blue light periods, respectively.

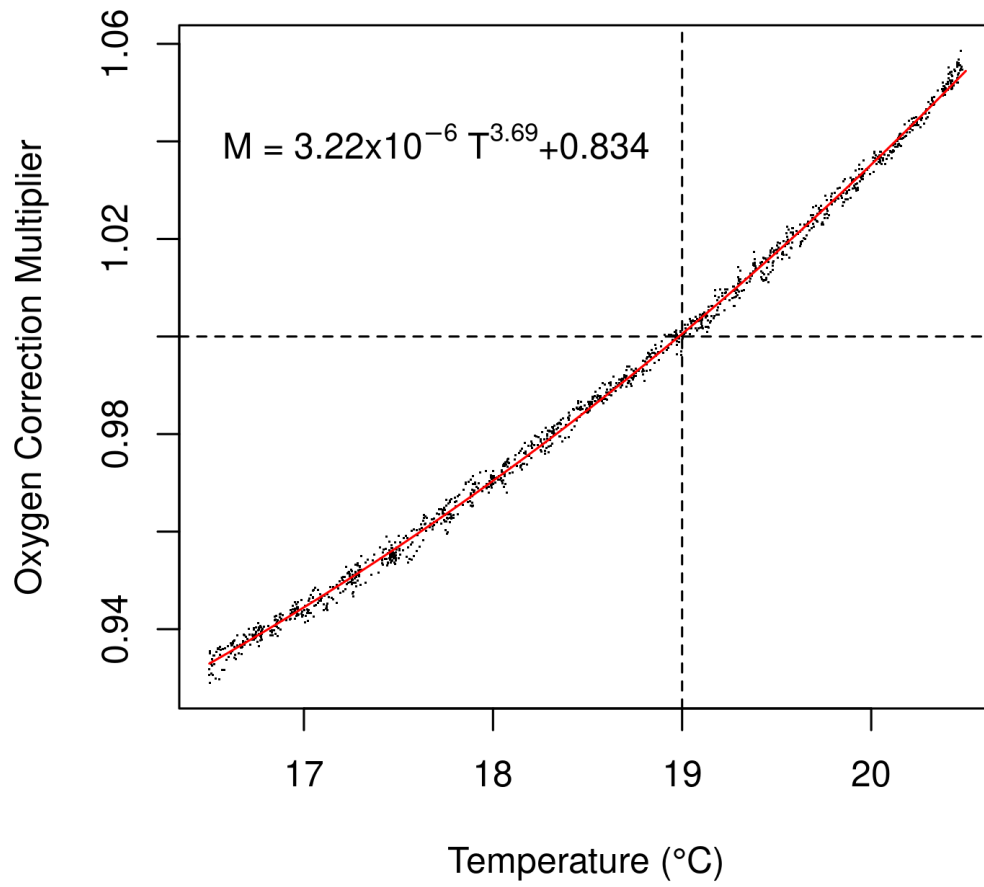


Figure 9. Relationship between temperature and oxygen correction multiplier to correct oxygen readings to 19°C.

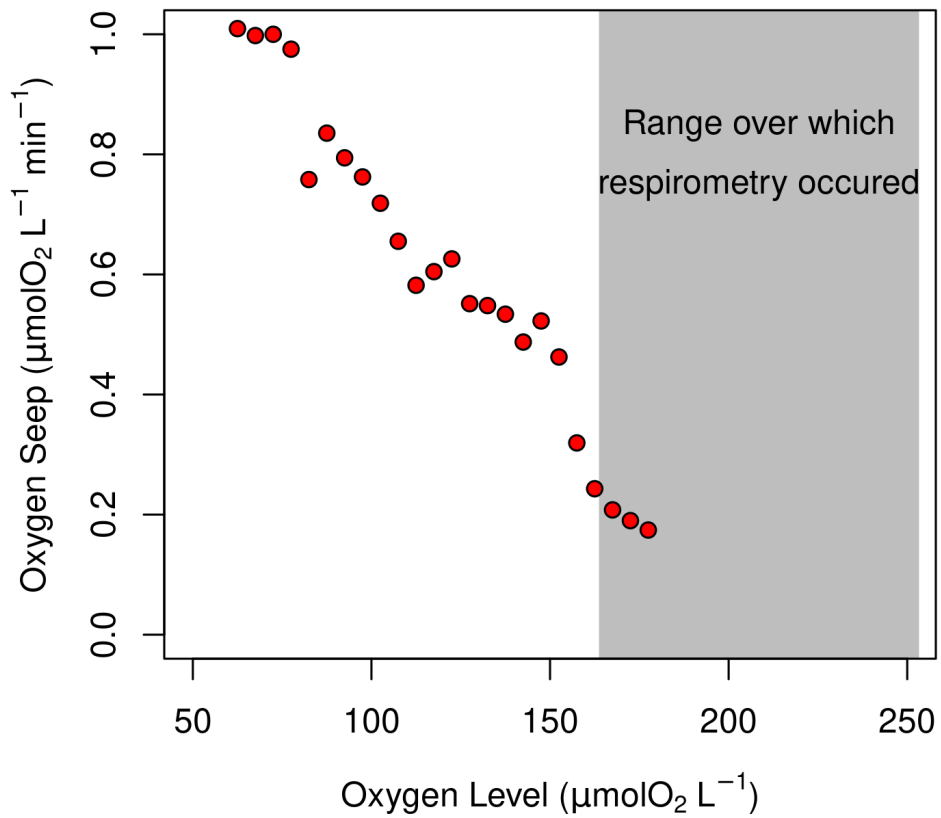


Figure 10. The oxygen seep for the custom respirometer used in experimentation. The range of oxygen levels at which respirometry occurred, from 164 to 253 $\mu\text{mol O}_2 \text{ L}^{-1}$, is shaded. Its oxygen seep was at a maximum of 0.2 $\mu\text{mol O}_2 \text{ L}^{-1} \text{ min}^{-1}$.

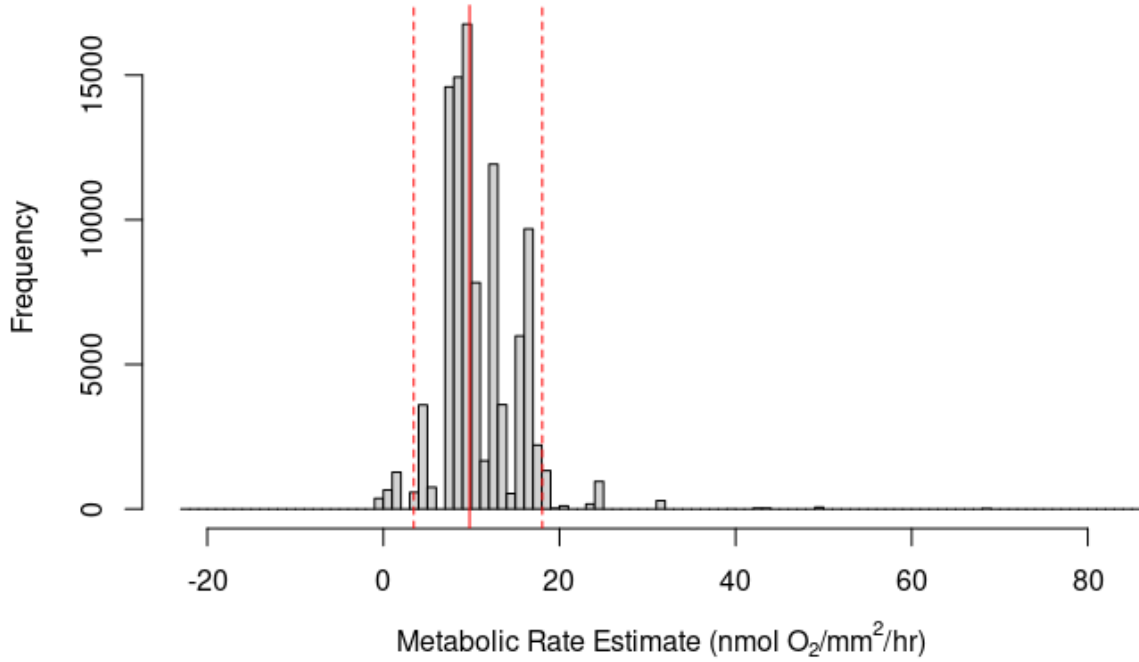


Figure 11. Histogram indicating the median metabolic rate estimate (9.9 nmol $O_2/mm^2/hr$) of chromatophores as a solid red line and its 95% confidence interval (3.45 – 18.05 nmol $O_2/mm^2/hr$) as the dotted red lines determined by using bootstrap resampling.

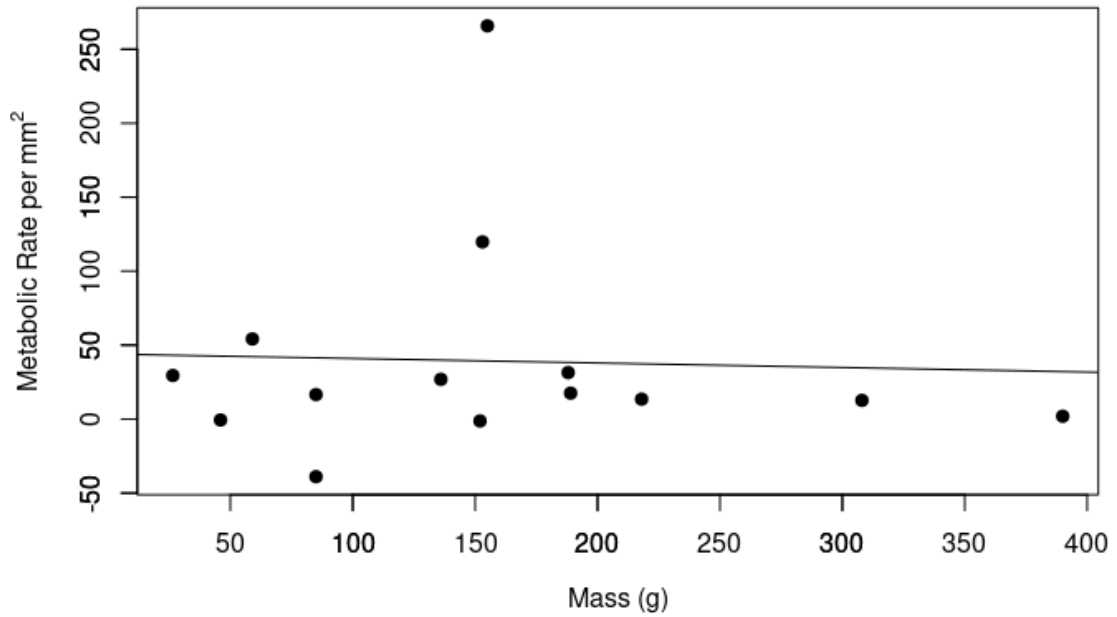


Figure 12. Relationship between octopus mass and metabolic rate per mm² of skin (linear regression, $p=0.8881$). Each point represents data from the excised skin of a different octopus.

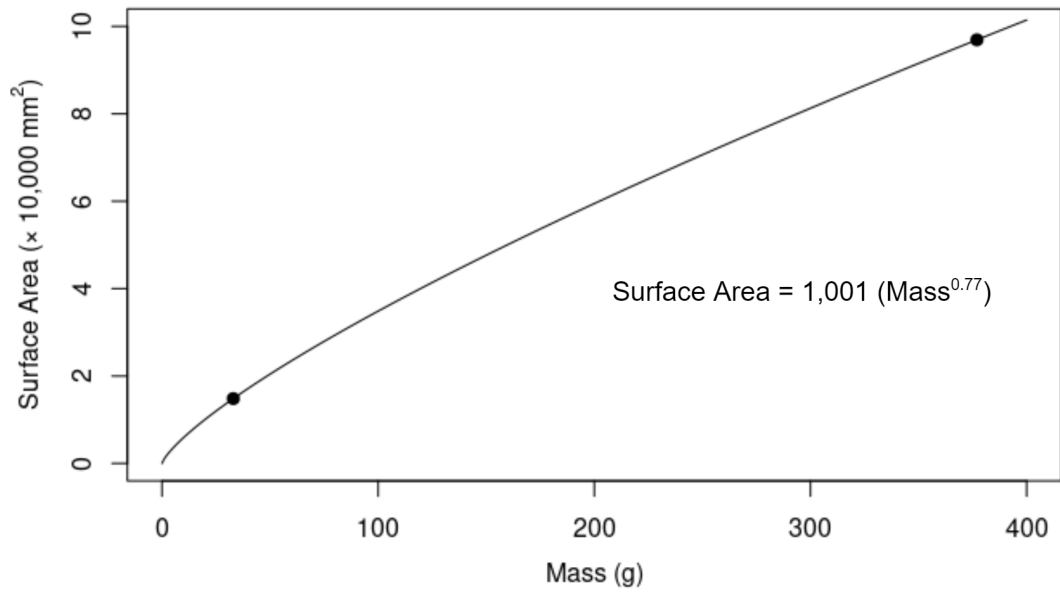


Figure 13. Relationship between *Octopus rubescens* surface area and mass as determined from preserved octopuses.

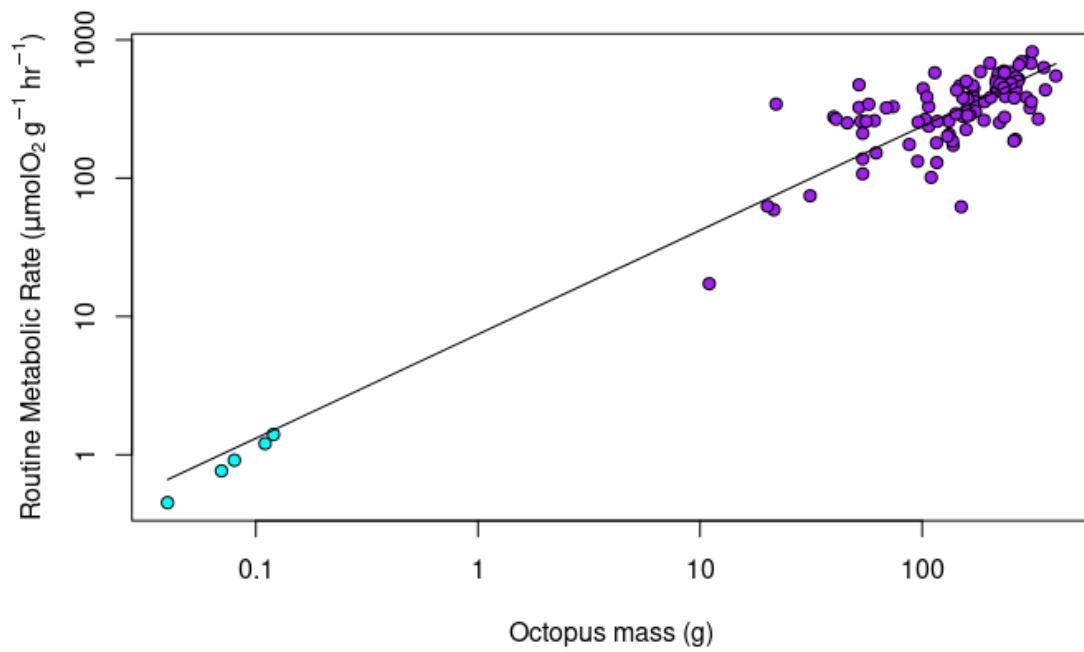


Figure 14. Relationship between routine metabolic rate and octopus mass. *Octopus rubescens* data from Kirt Onthank is shown in purple; *Octopus rubescens* paralarvae data from Seibel 1997 is shown in cyan.

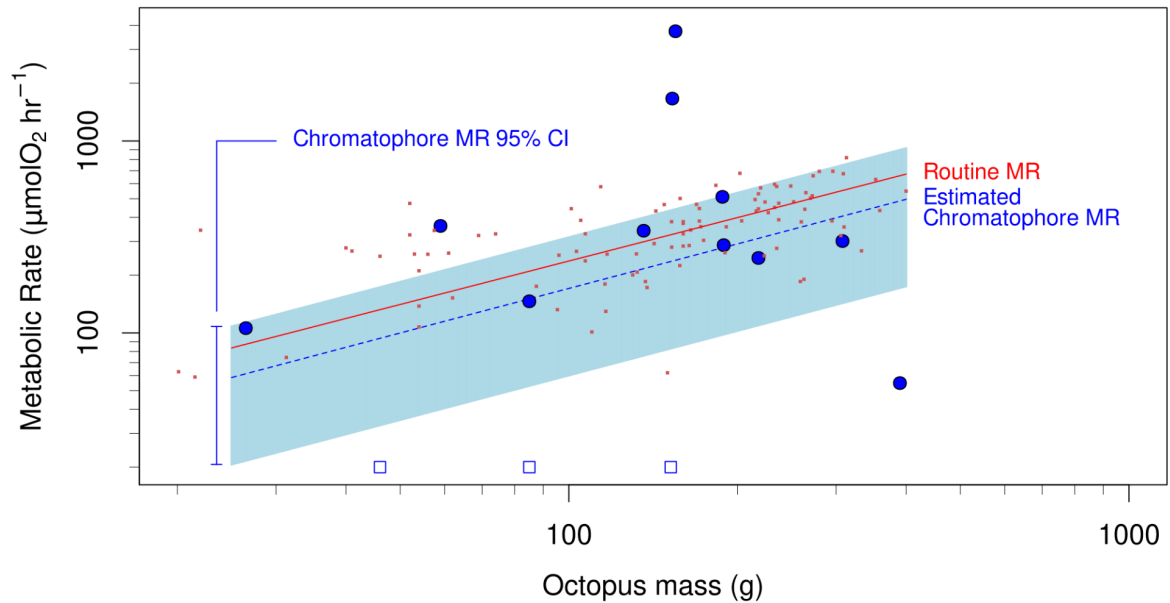
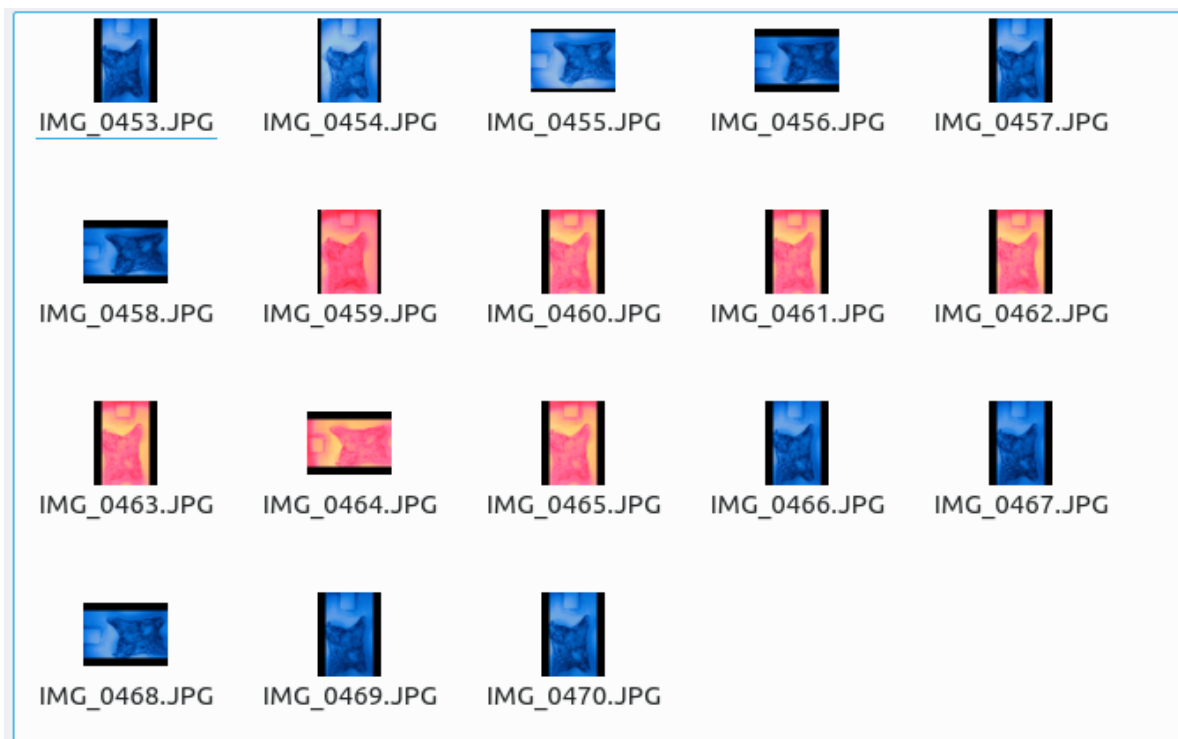


Figure 15. Relationship between octopus mass and metabolic rate. Red data points indicate previous routine metabolic rate measurements of *Octopus rubescens* by Kirt Onthank. The red line shows the trend among them. The blue dotted line depicts the estimated metabolic rate accounted for by chromatophores, and its 95% confidence interval is shaded in blue. The blue points are estimates of chromatophore metabolic rate for each octopus used in this study. Those that are squares have negative metabolic rate values, indicating a higher oxygen uptake during the periods of only red light.

APPENDIX A

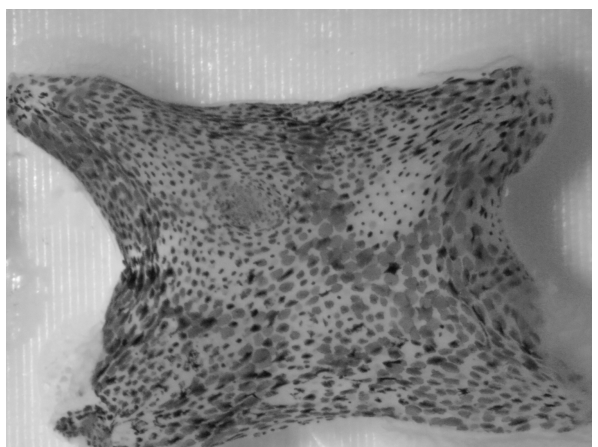
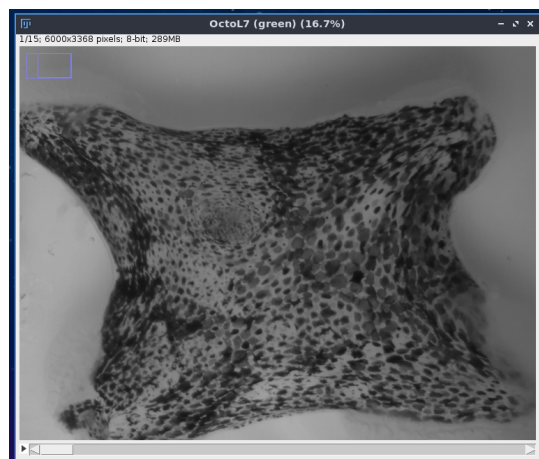
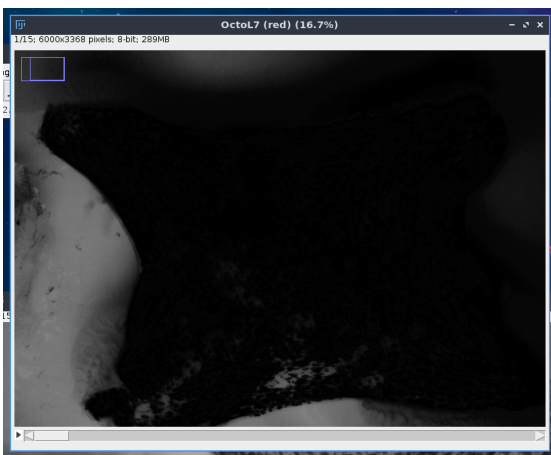
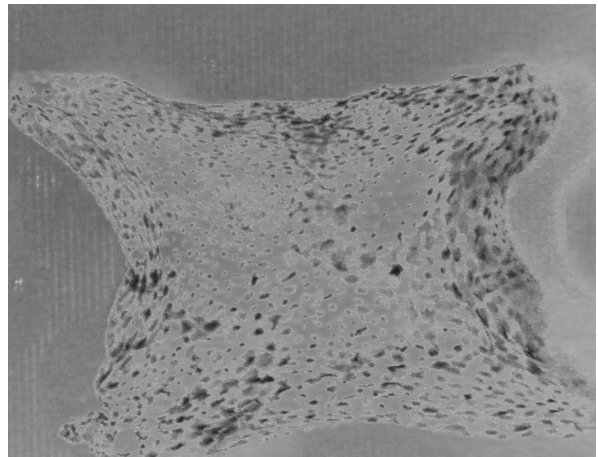
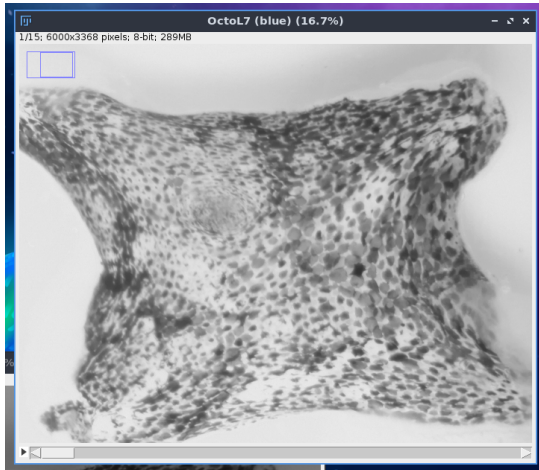
Excised Skin Photograph Analysis Steps Using FIJI

One folder was created for the excised skin photographs of each octopus used. Each folder was labeled with the identification number of each octopus (Table 1) and contained all of the pictures taken for that octopus's trial. Photos from the second red light period were removed from each folder so they contained photos of the skin in the following order of light periods: blue, red, blue. Blurry pictures, in which chromatophores were distorted, were removed from each series of pictures. The following series of screencaps are an example for octopus L7:

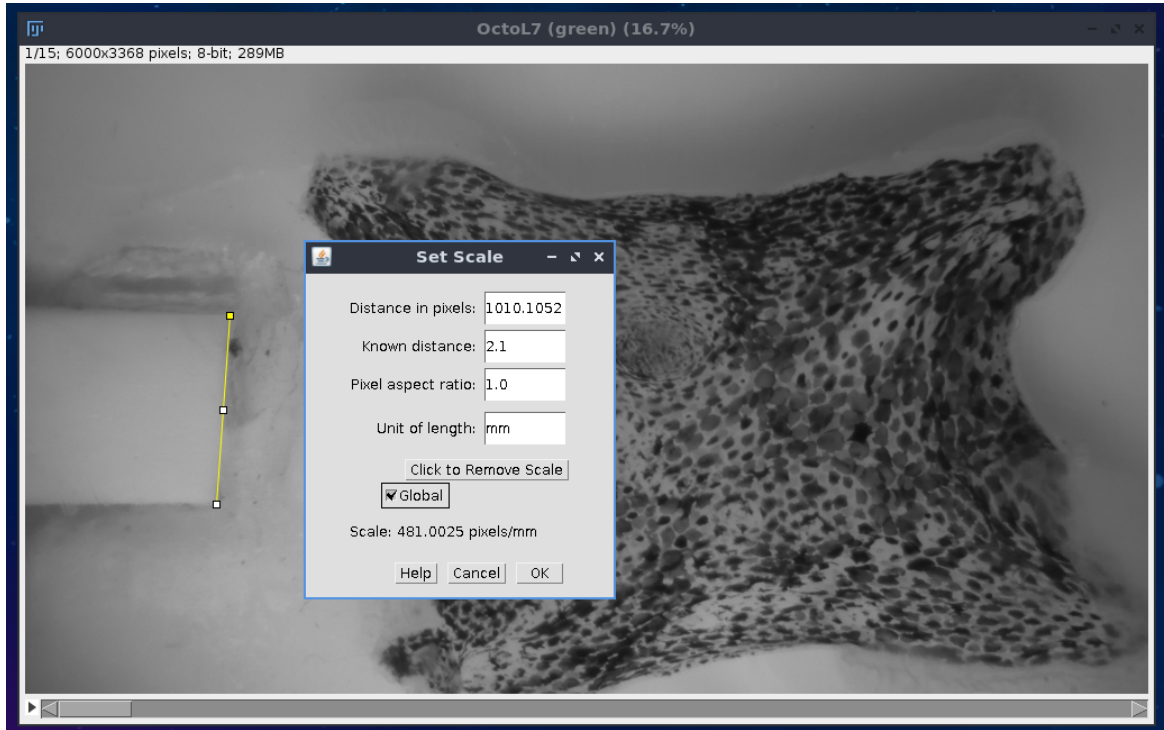


Once the folder contained all photos to be processed, they were imported into FIJI (FIJI Is Just ImageJ) using the following selections: File, Import, Image Sequence. The channels were split using the following sequence: Image, Color, Split Channels. Both the red and blue channels were deleted so that only the green channel remained for the series

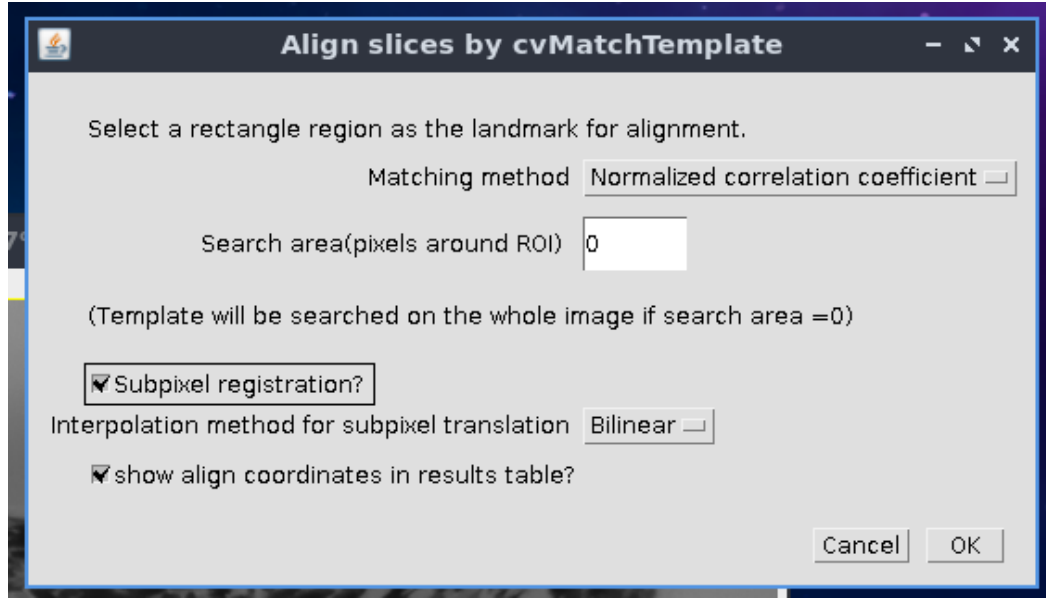
of images. This was found to show chromatophores most clearly and consistently for photographs in both red and blue light. The blue, red, and green color channels, respectively, are shown below for periods of blue light (left) and red light (right):



Next, the segment selection tool was used to select a line segment along the width of the temperature sensor. This had a known length of 2.1 millimeters. This scale was set using the following selections: Analyze, Set Scale.



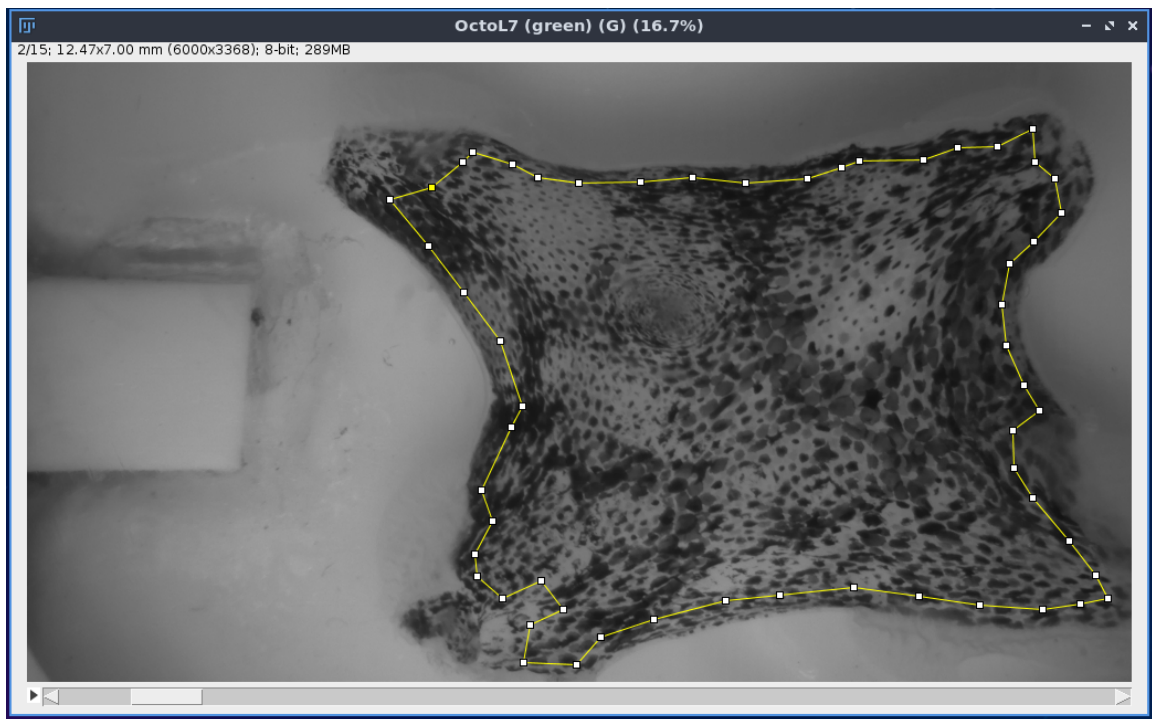
Next, the series of images required alignment so that the chromatophores appeared to be in the same place throughout the stack. Alignment was done using the following selections: Plugins, Template Matching, Align slices in stack. The template matching plugin (Template_Matching.jar) is not provided in FIJI by default and had to be installed beforehand. The following settings were used:



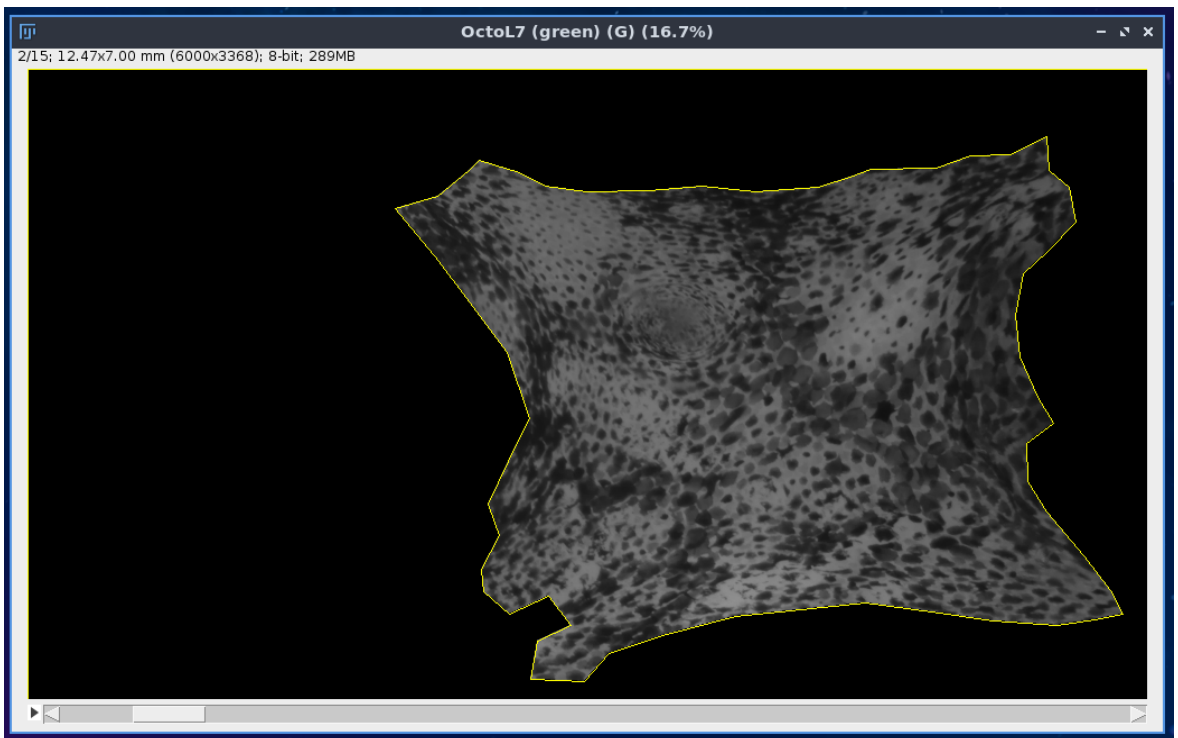
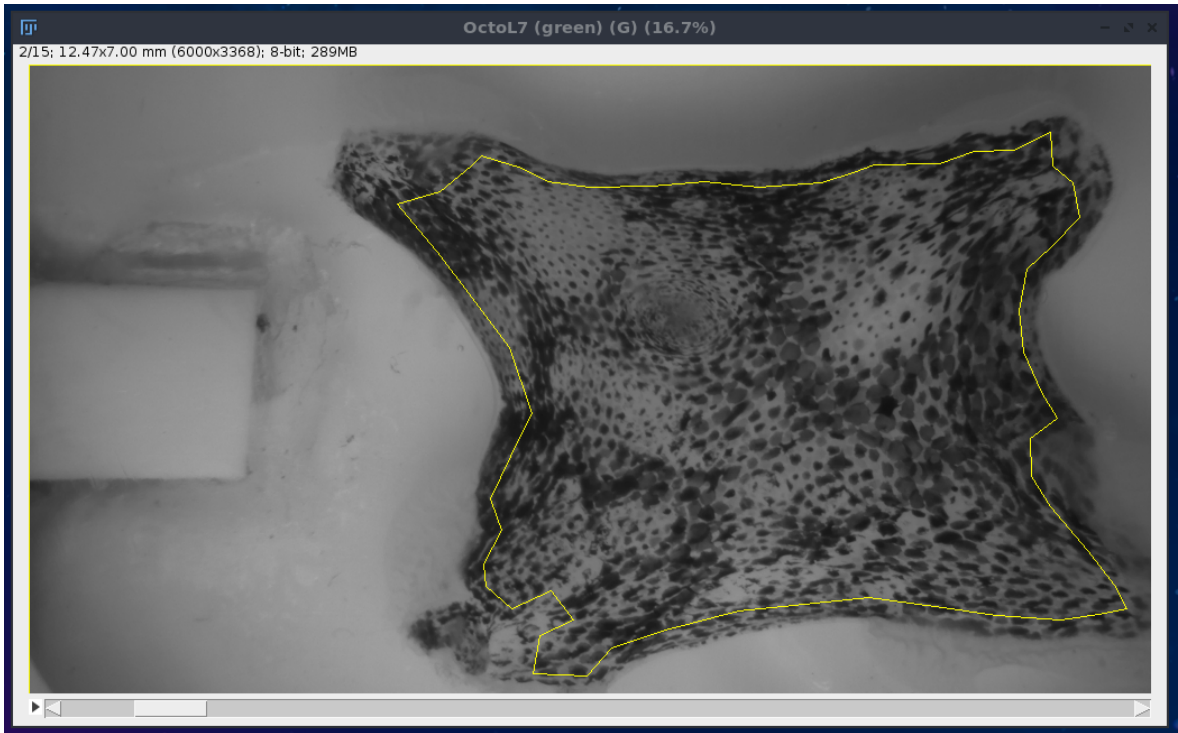
A rectangular region was selected as the landmark, which was used for reference, on the first image in the series. This was usually a mark that appeared prominently on all photos in the series and was not part of the excised skin.



Correct alignment was confirmed by scrolling through the series of images to see if any were offset. If they were, alignment steps were repeated using a different landmark in the selected rectangle region. If images were offset at an angle, rather than vertically or horizontally, the following were selected to adjust it: Image, Transform, Rotate. After all images were aligned, the polygon tool was used to select the area of the skin. The area was selected inside of the edges of the skin, leaving a small margin, so that only excised skin was included across the entirety of the image stack. Glue and ambiguous parts, which occurred frequently on the skin margins, were not included in the selected area of the skin.

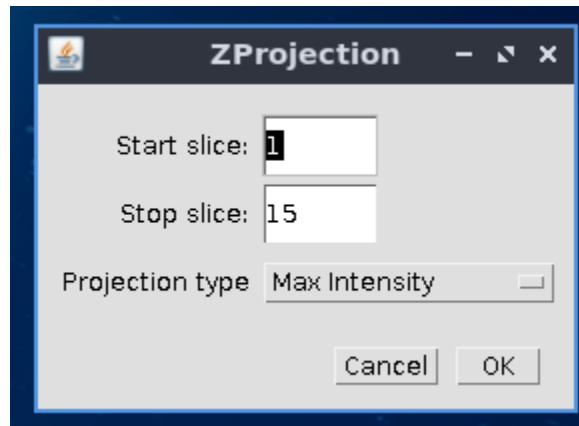


The area measurement of the skin was determined using the following selections: Analyze, Measure. The selection was made inverse (Edit, Selection, Make Inverse), so that all of the image besides the skin was selected, and was deleted so that only the excised skin remained.

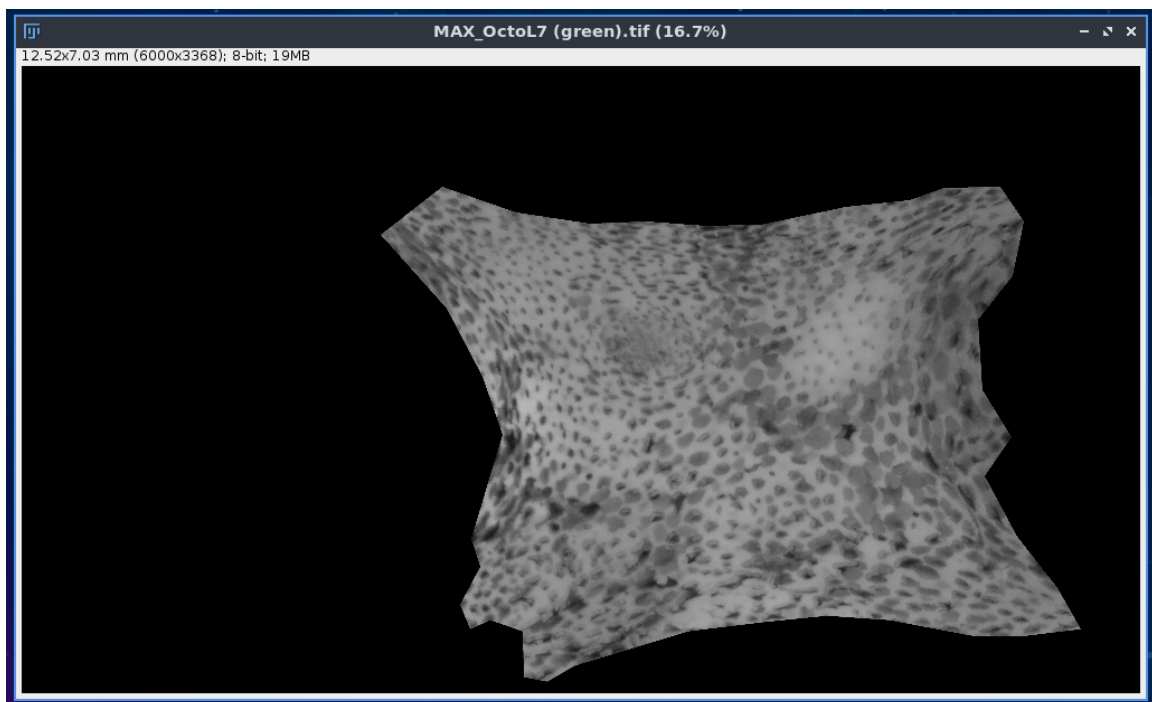


The following selections were made to create a new image with the maximum intensity of the aforementioned stack: Image, Stacks, Z Project. The following settings were used

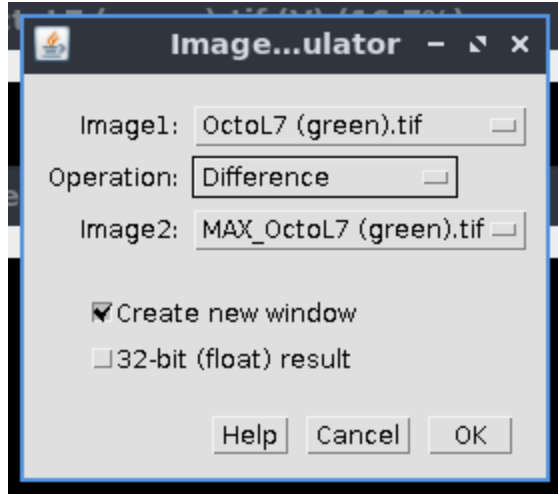
with the respective start and stop slices dependent on the number of images in the original stack:



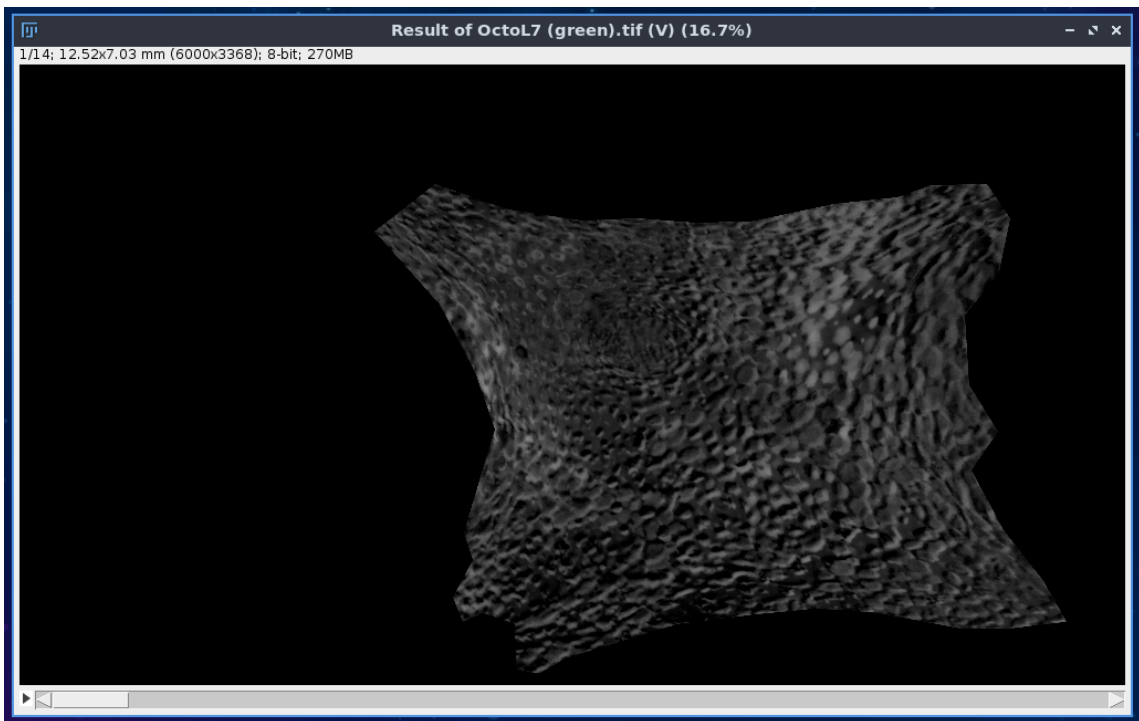
The resulting image appeared as follows:



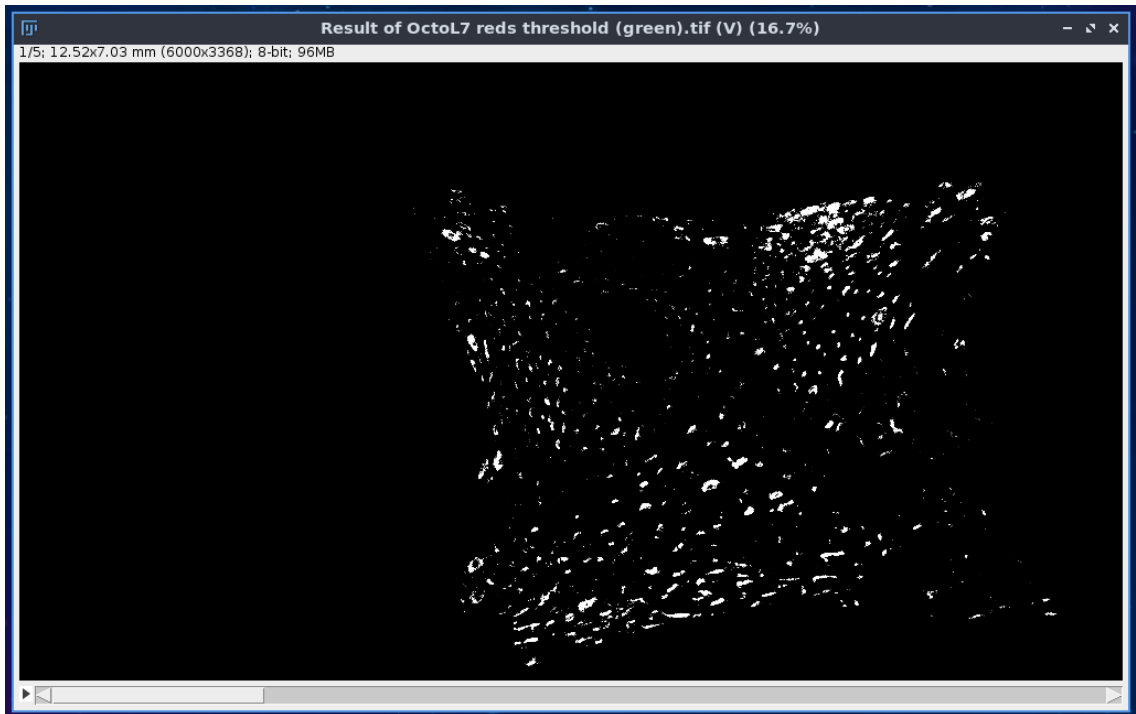
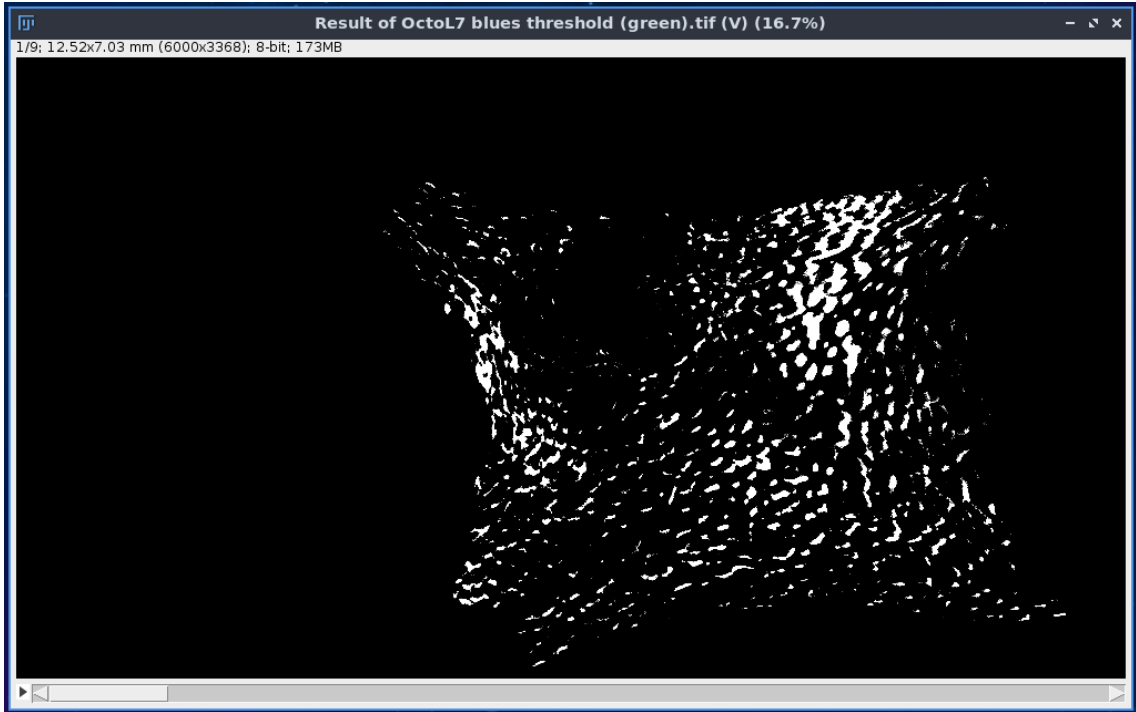
To create a new image stack that showed the difference between the original image stack and the maximum intensity image, the following selections were made: Process, Image Calculator. Image1 was the original image stack and Image2 was the image of maximum intensity. Settings were selected as follows:



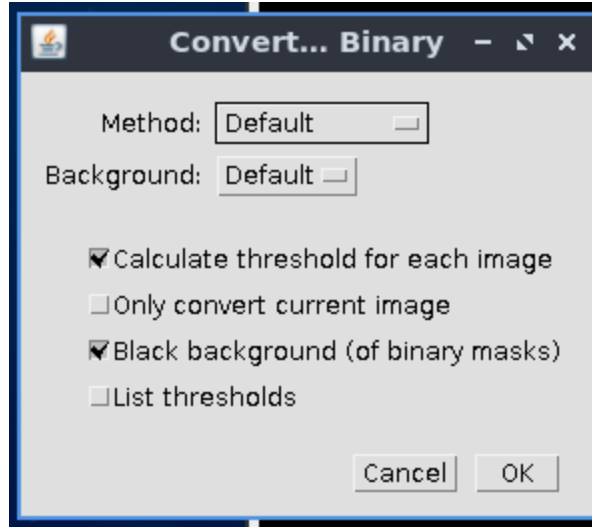
The resulting image stack appeared as follows:



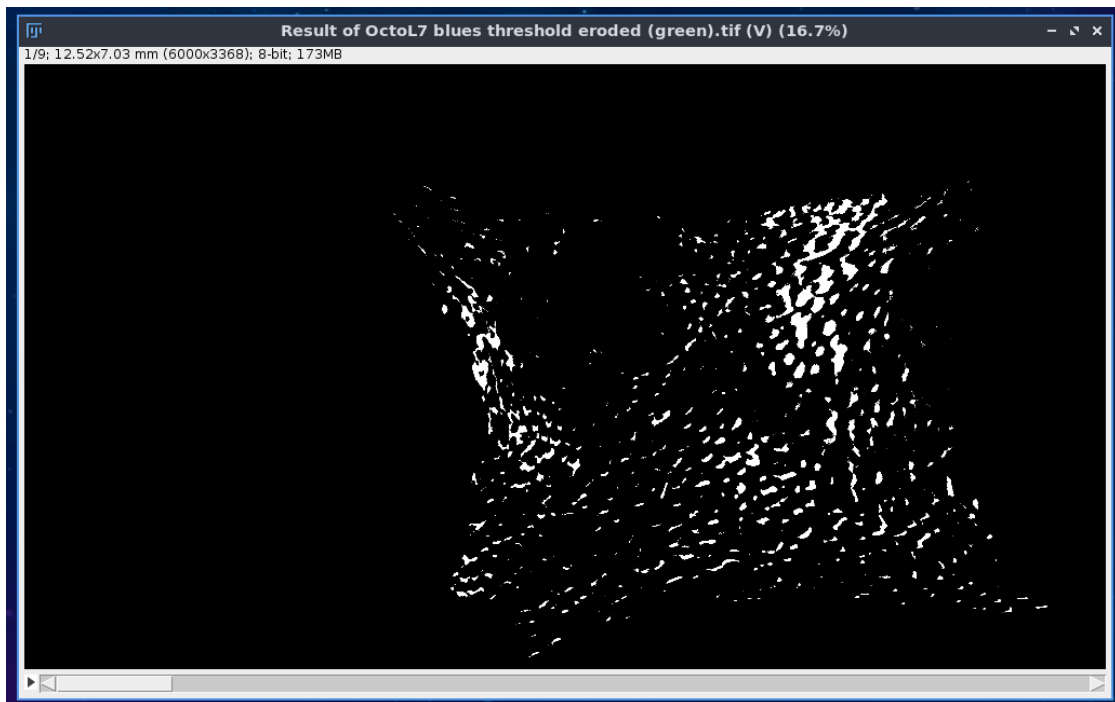
This image stack was separated into two image stacks- one that contained images from the blue light periods, and another that contained images from the red light period. For each of these stacks, a threshold was created to determine areas occupied by chromatophores using the following selections: Image, Adjust, Threshold. The resulting image stacks appeared as follows, for the blue and red light periods, respectively:



These stacks were converted to binary using the following selections: Process, Binary, Make Binary. The following settings were used:



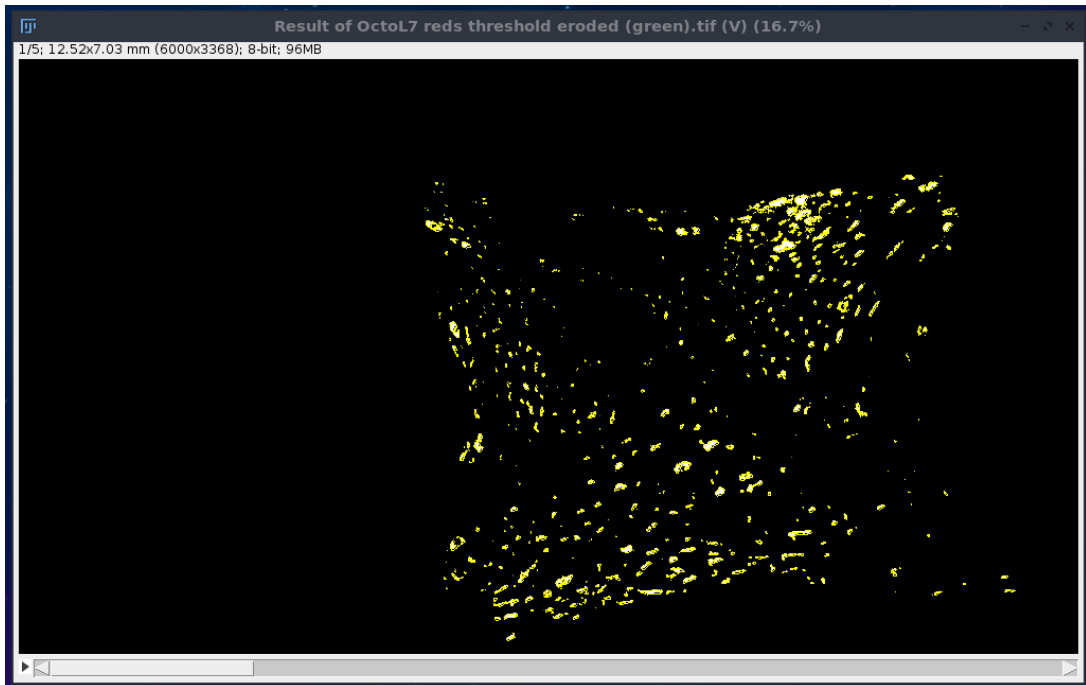
To remove inconsequential points that were too small to be counted as chromatophores, the following selections were used: Process, Binary, Erode. The resulting image stacks appeared as follows, for the blue and red light periods, respectively:





To select only the white parts of the image, that represented the area occupied by chromatophores, the following selections were made: Edit, Selection, Create Selection.

The selection appeared as follows:



By using Ctrl + M, the area of the selected region was shown. The area of this chromatophore-containing region was recorded and the selection repeated for each image in each stack.

APPENDIX B

3D Scanning Steps Using EinScan Desktop 3D Scanner

Once all parts of both preserved octopuses were scanned using an EinScan desktop 3D scanner, and each part was saved as a separate file, the following steps were taken to process each of the files in order to determine an accurate surface area:

- File, Import Mesh: To select the individual file to be processed
- Filters, Remeshing, Surface Reconstruction Screened Poisson: To smooth the surface and fill in gaps and holes
- Delete Current Mesh: To delete the raw original version of the scan from view
- Save: To save the smoothed version of the scan as a .alias file
- Selection Tool: Used to select and delete areas of the surface that would not be used in the determination of the total octopus surface area, such as the circular portion at the ends of the arms closest to the body where they were removed
 - Select Faces/Vertices Inside Polyline Area
 - Select Faces in a Rectangular Region
 - Selecting while pressing Shift erases the selection
 - Delete: To remove the selected area from the 3D model
- Record Surface Area: To record the surface area of the processed, finished file
 - Filters
 - Quality Measure and Computation
 - Compute Geometric Measures
 - Record “Mesh surface area is...” value

APPENDIX C

R Markdowns for Data Analysis

Respirometer Oxygen Fastness Test

Loading Libraries

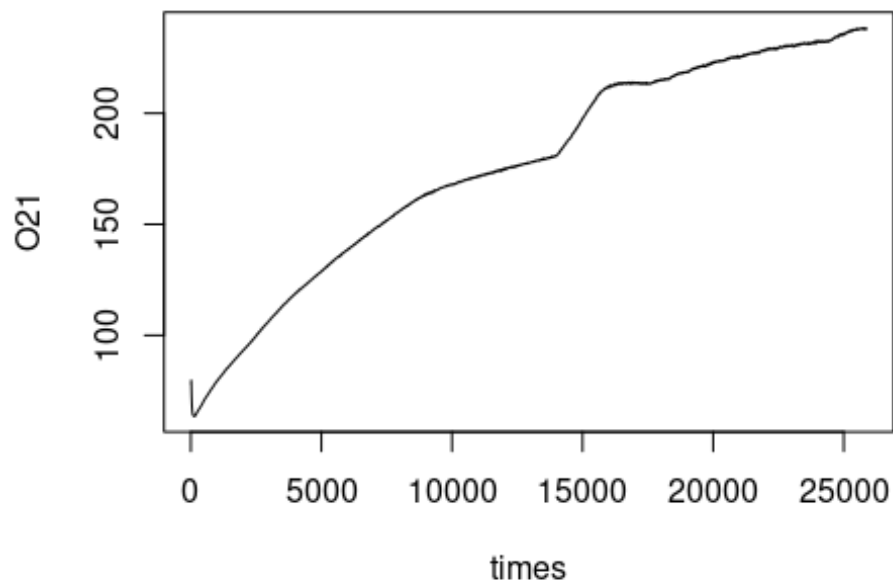
```
library(OTools)
```

Reading in the data

```
deoxy3=read.pyro("Blank_Deox3.txt")
```

Looking at the data

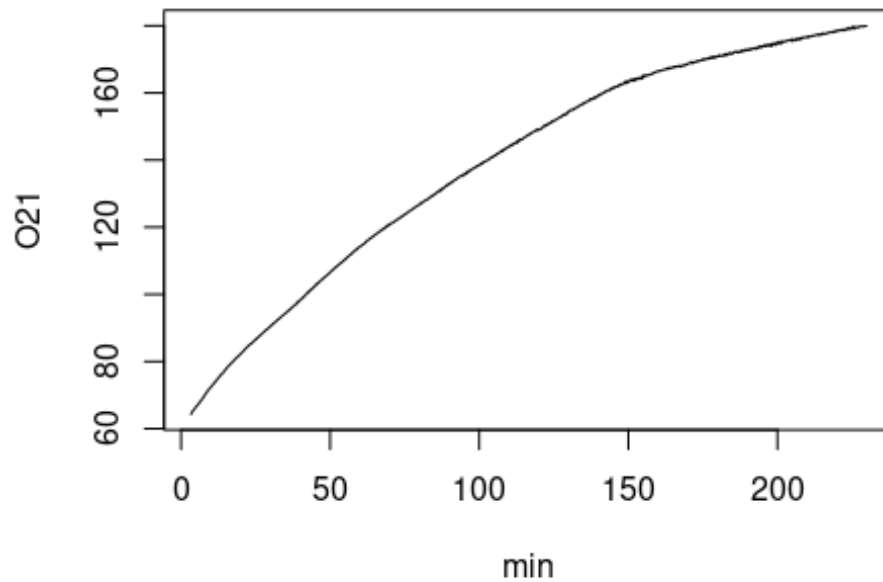
```
plot(O21~times,data=deoxy3,type="l")
```



There is some oxygen change at the beginning due to the beginning of mixing in the respirometer. We want to cut that off. Also, around 15000 s there is a sudden bump up in the oxygen level. That is likely a bubble that was introduced. We will also clear out that part.

Paring the data down

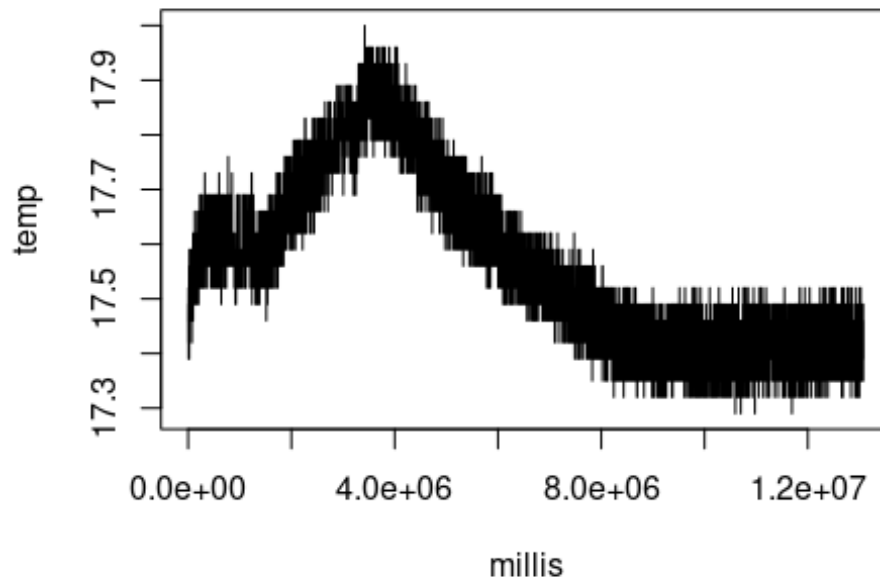
```
deoxy3.pared=deoxy3[deoxy3$times>=200&deoxy3$O21<=180,]  
deoxy3.pared$min=deoxy3.pared$times/60  
plot(O21~min,data=deoxy3.pared, type="l")
```



This is now the final pared-down data that we will use.

Checking temperature

```
deoxy3.temp=read.csv("Blank_Deox3_Temp.txt")  
plot(temp~millis,data=deoxy3.temp,type="l")
```



Temperature changes very little (by 0.6 C over 4 hours), so I am not going to use temperature correct.

Calculating oxygen seep over 5 $\mu\text{molO}_2/\text{L}$ intervals

First off I make the bins in which I will calculate the oxygen seep

```
bins=seq(from=60,to=180,by=5)
```

Then I am making a numeric vector to store the slope for those bins

```
slopes=numeric()
```

This loop will then calculate the slope (in $\mu\text{molO}_2/\text{L}/\text{min}$) for each of the 5 $\mu\text{molO}_2/\text{L}$ bin

```
for (i in 1:(length(bins)-1)){
```

```
  step.lm=lm(O21~min,data=deoxy3.pared[deoxy3$O21>=bins[i]&deoxy3$O21<=bins[i+1],])
```

```
  slopes[i]=step.lm$coefficients[2]
```

```
}
```

```
png("Oxygen_fastness.png",width=5,height=5,units = "in",res=300)
```

```
par(fig=c(0,1,0,1))
```

```
plot(slopes~seq(from=62.5,
```

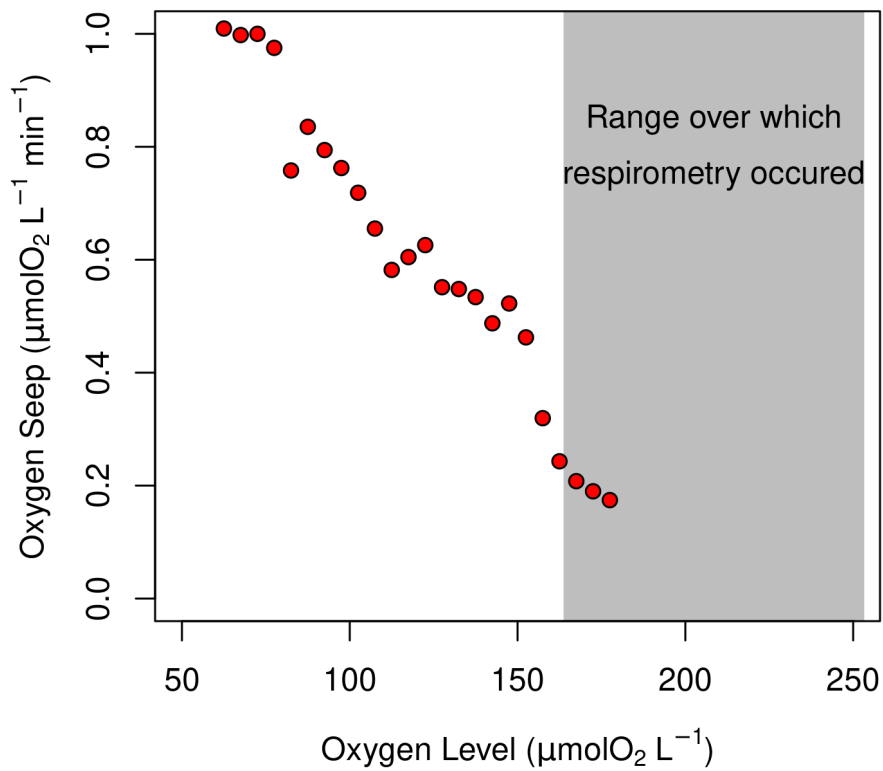
```
to=177.5,by=5),pch=21,bg="red",xlim=c(50,250),ylim=c(0,1),
```

```

    ylab="",xlab=expression("Oxygen Level ("*mu*"molO"[2]*"
L^-1*")"),type="n",axes=F)
mtext(expression("Oxygen Seep ("*mu*"molO"[2]*" L^-1*"
min^-1*")"),side=2,line=2.5)
abline(v=seq(from=164,to=253,by=0.1),col="grey")
box()
axis(1)
axis(2)
points(slopes~seq(from=62.5, to=177.5,by=5),pch=21,bg="red")
text(mean(c(164,253)),0.8,"Range over which",pos=3)
text(mean(c(164,253)),0.7,"respirometry occured",pos=3)
dev.off()

## png
## 2

```



Empirical Temperature Response Determination

Loading the OTools library
`library(OTools)`

Loading Data and manipulation

Only the first test worked well. The other two got bubbles.

```
mrback=read.pyro("temp_test1.txt")  
mrarduino=read.csv("temp_test1_arduino.txt")
```

Converting milliseconds in the arduino file to seconds. Also adding in a 4 second offset

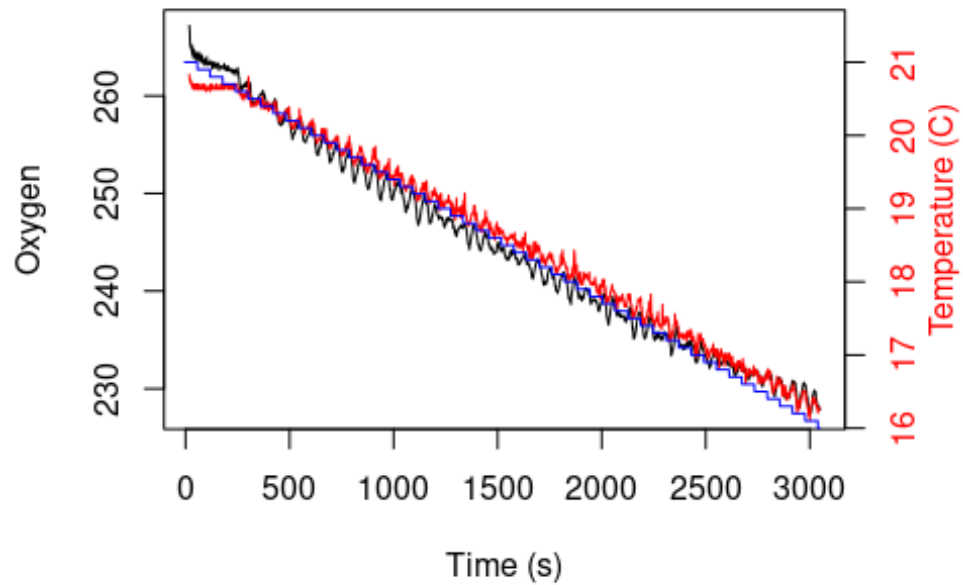
```
mrarduino$sec=round(mrarduino$millis/1000)-4
```

I am now taking the temperature data from the arduino file to fill data in the PyroScience file. I am doing this using a loop in which I find the nearest time(s) to each datapoint.

```
mrback$temp_ar=0  
for (i in 1:nrow(mrback)){  
  
mrback$temp_ar[i]=mean(mrarduino$temp[which.min(abs(mrarduino$sec-mrback$times[i]))])  
}  
}
```

Now I am plotting an overlay of temperature (in red) and oxygen (in black).

```
par(fig=c(0,0.95,0,1))  
plot(O21~times,data=mrback,type="l",ylab="Oxygen",xlab="Time (s)")  
par(new=T)  
plot(temp_ar~times,data=mrback,type="l",col="red",axes=F,ylab="",xlab="",ylim=c(16.2,21.5))  
lines(setpoint~sec,data=mrarduino,col="blue")  
axis(4,col.axis="red")  
mtext("Temperature (C)",side=4,line=2,col="red")
```

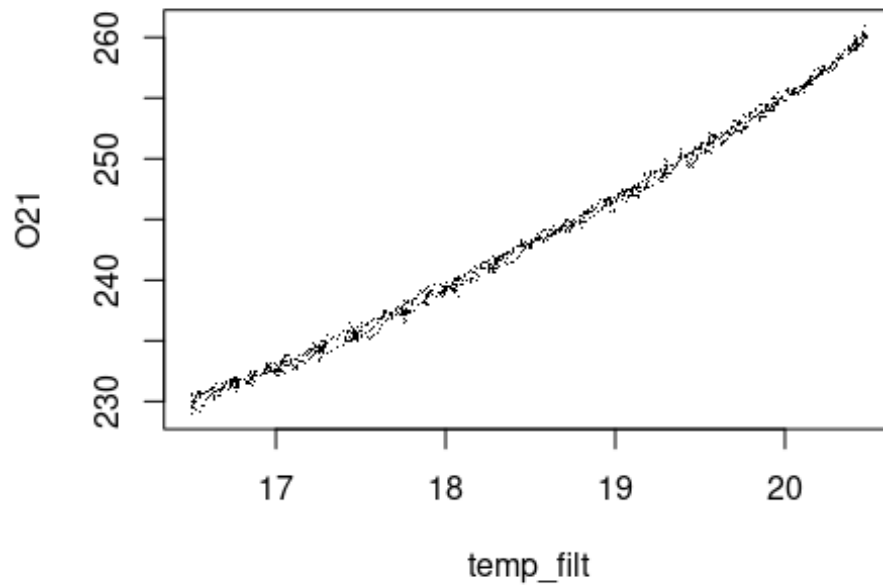


Now I am smoothing the temperature, because it is a bit noisier than the oxygen data.

```
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-3,to=3,length.out=smooth+1)))
mrback$temp_filt=filter(mrback$temp_ar,coeff)
```

Now that I have smoothed temperature, I am plotting oxygen against temperature to see the tightness of the relationship.

```
plot(O21~temp_filt,data=mrback[mrback$temp_filt>16.5&mrback$temp_filt<20.5,],pch=".")
```

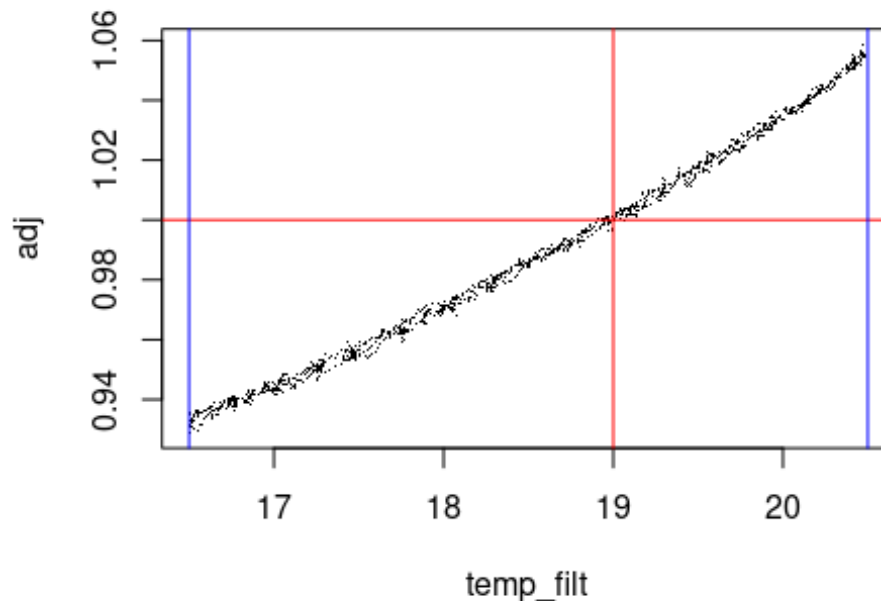


Here I am calculating an adjustment by dividing each oxygen value by the oxygen value at 19C. This gives me the multiplier I would use to adjust oxygen at each temperature.

```
mrback$adj=mrback$O21/mrback$O21[which.min(abs(mrback$temp_filt-19.00))
]
```

Now I am plotting that multiplier against temperature.

```
plot(adj~temp_filt,data=mrback[mrback$temp_filt>16.5&mrback$temp_filt<20.5,],pch=".")
abline(h=1,col="red")
abline(v=19,col="red")
abline(v=20.5,col="blue")
abline(v=16.5,col="blue")
```



Here I am using non-linear least square regression to find the relationship between temperature and the appropriate multiplier.

```
adj.nls=nls(adj~a*temp_filt^b+c,data=mrback[mrback$temp_filt>16.5&mrback
k$temp_filt<20.5,],
          start=list(a=1.212e-6,b=4,c=8.428e-1))
```

Next up, I see what the fitted coefficients are. These are the values I will actually use in the temperature correction.

```
coef(adj.nls)

##          a          b          c
## 3.223674e-06 3.686095e+00 8.338341e-01
```

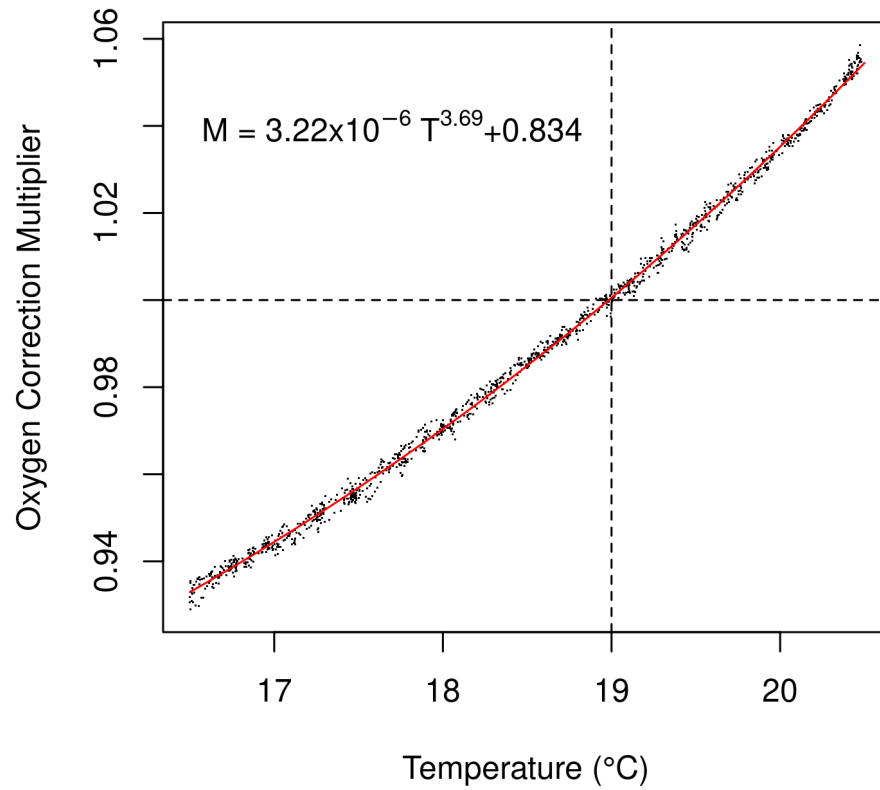
Finally, this is the plot of the fit between temperature and the multiplier.

```
png("Temperature_Correction.png",width=5,height=5,units="in",res=300)
plot(adj~temp_filt,data=mrback[mrback$temp_filt>16.5&mrback$temp_filt<2
0.5,],pch=".",
     ylab="Oxygen Correction Multiplier",
     xlab="Temperature (\u00B0C)")
at=seq(from=16.5,to=20.5,length.out=100)
at.pred=predict(adj.nls,newdata=data.frame(temp_filt=at))
lines(at,at.pred,col="red")
abline(v=19,lty=2)
abline(h=1,lty=2)
```



```
text(17.7,1.04,expression("M = 3.22x10-6" T3.69" +0.834"))  
dev.off()
```

```
## png  
## 2
```



Relationship between temperature and oxygen correction multiplier to correct oxygen readings to 19°C

Chromatophore Oxygen Consumption

```
library(remotes)

install_github("KirtOnthank/OTools")

library(OTools)

trim.begin=45
trim.end=15

resp=data.frame(matrix(nrow=17,ncol=13))
colnames(resp)=c("octo", "on1", "on1.o2", "on1.temp", "off1", "off1.o2", "off
1.temp", "on2", "on2.o2", "on2.temp", "off2", "off2.o2", "off2.temp")
```

Octopus 01

Loading data

```
octo01=read.pyro('Octo_01r_08-18-2020.txt')
octo01.temp=read.csv('Octo_01r_08-18-2020_temp.txt')
octo01.temp$times=octo01.temp$millis/1000
```

Smoothing temperature at different time offsets

```
octo01=cbind(octo01[,1:28],matrix(nrow=nrow(octo01),ncol=26))
colnames(octo01)=c(colnames(octo01)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",
"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",
"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")
```

```
for (j in -6:6){
  for (i in 1:nrow(octo01)){

octo01[i,35+j]=mean(octo01.temp$temp[which.min(abs(octo01.temp$times-oc
to01$times[i]+j))])
  }
}
```

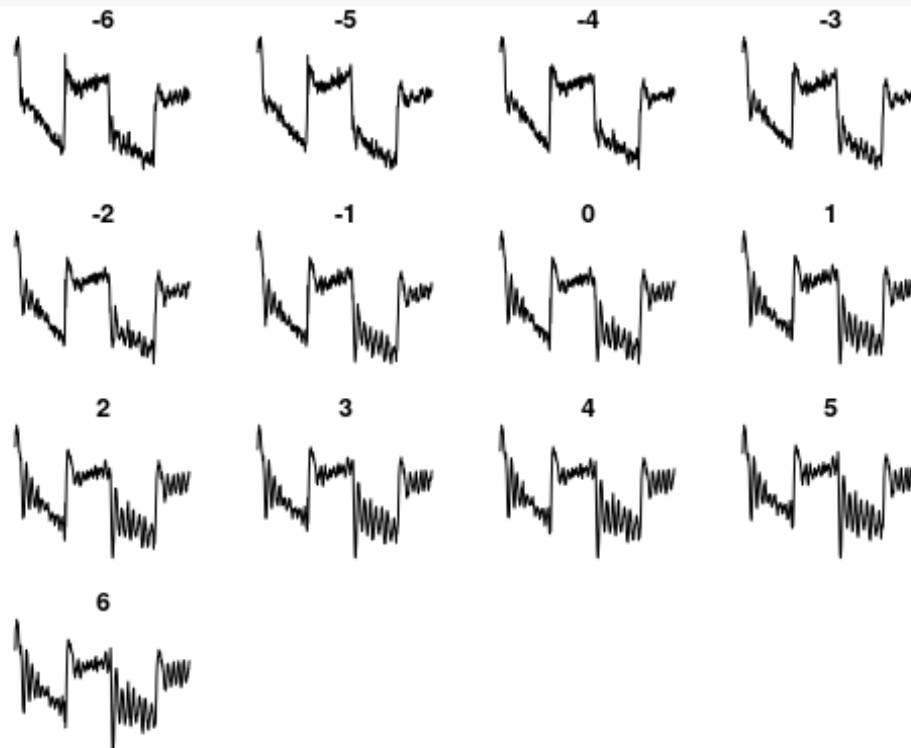
```
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))
```

```
for (i in -6:6){
  octo01[,48+i]=filter(octo01[,35+i],coeff)
}
```

Choosing correct time offset

```
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
```

```
plot(octo01$O21/((3.223674e-06*octo01[,48+i]^3.686095+8.338341e-01)^2)~
octo01$times,type="l",ylab="",xlab="",main=i,axes=F)
}
```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-5
```

```
octo01$O21adj=octo01$O21/((3.223674e-06*octo01[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

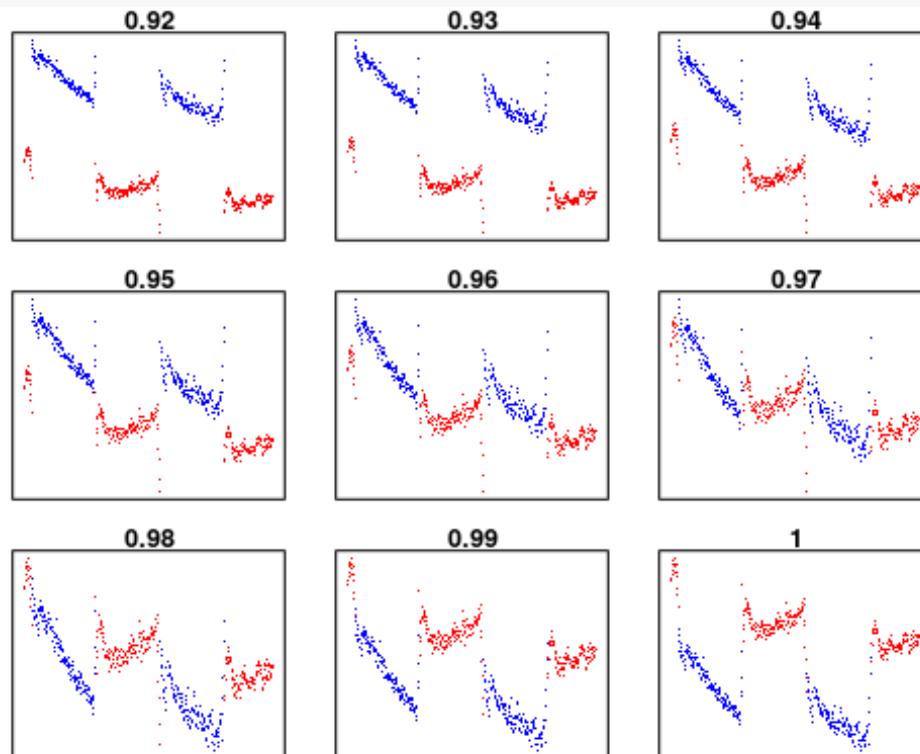
```
par(mfrow=c(3,3))
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
  temp=octo01$O21adj
```

```

temp[octo01$ambient1>20]=temp[octo01$ambient1>20]/i
plot(temp[octo01$ambient1<20]~octo01$times[octo01$ambient1<20],pch=".",
axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
points(temp[octo01$ambient1>20]~octo01$times[octo01$ambient1>20],pch=".",
,col="blue")
  box()
}

```

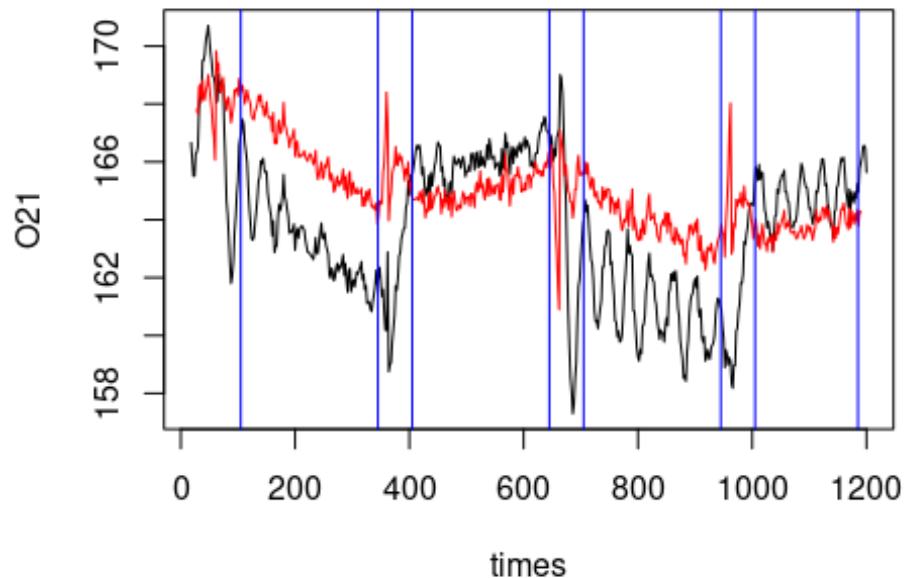


Choose the offset that best aligns the ends of the blue dots to the red dots.

```

Applying light correction
light.off=0.97
octo01$O21adj2=octo01$O21adj
octo01$O21adj2[octo01$ambient1>20]=octo01$O21adj2[octo01$ambient1>20]/l
ight.off
plot(O21~times,data=octo01,type="l")
lines(O21adj2~times,data=octo01,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo01$times>60+trim.begin)),
         min(which(octo01$times>360-trim.end)),
         min(which(octo01$times>360+trim.begin)),
         min(which(octo01$times>660-trim.end)),
         min(which(octo01$times>660-trim.begin)),
         min(which(octo01$times>960-trim.end)),
         min(which(octo01$times>960-trim.begin)),
         min(which(octo01$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[1]=1
```

```
octo01.1=
```

```
(octo01$O21adj2[breaks[1]:breaks[2]]-
 octo01$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[1]=mean(octo01.1)*1000
```

```
resp$on1.o2[1]=mean(octo01$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[1]=mean(octo01$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo01.2=
```

```
(octo01$O21adj2[breaks[3]:breaks[4]]-
 octo01$O21adj2[breaks[3]:breaks[4]+1])*
```

```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[1]=mean(octo01.2)*1000
resp$off1.o2[1]=mean(octo01$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[1]=mean(octo01$tempv5_filt[breaks[3]:breaks[4]])

octo01.3=
(octo01$O21adj2[breaks[5]:breaks[6]]-
 octo01$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[1]=mean(octo01.3)*1000
resp$on2.o2[1]=mean(octo01$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[1]=mean(octo01$tempv5_filt[breaks[5]:breaks[6]])

octo01.4=
(octo01$O21adj2[breaks[7]:breaks[8]]-
 octo01$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[1]=mean(octo01.4)*1000
resp$off2.o2[1]=mean(octo01$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[1]=mean(octo01$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 02

Loading data

```

octo02=read.pyro('Octo_02r_08-18-2020.txt')
octo02.temp=read.csv('Octo_02r_08-18-2020_temp.txt')
octo02.temp$times=octo02.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo02=cbind(octo02[,1:28],matrix(nrow=nrow(octo02),ncol=26))
colnames(octo02)=c(colnames(octo02)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",

"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",

"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo02)){
    octo02[i,35+j]=mean(octo02.temp$temp[which.min(abs(octo02.temp$times-oc
to02$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo02[,48+i]=filter(octo02[,35+i],coeff)
}

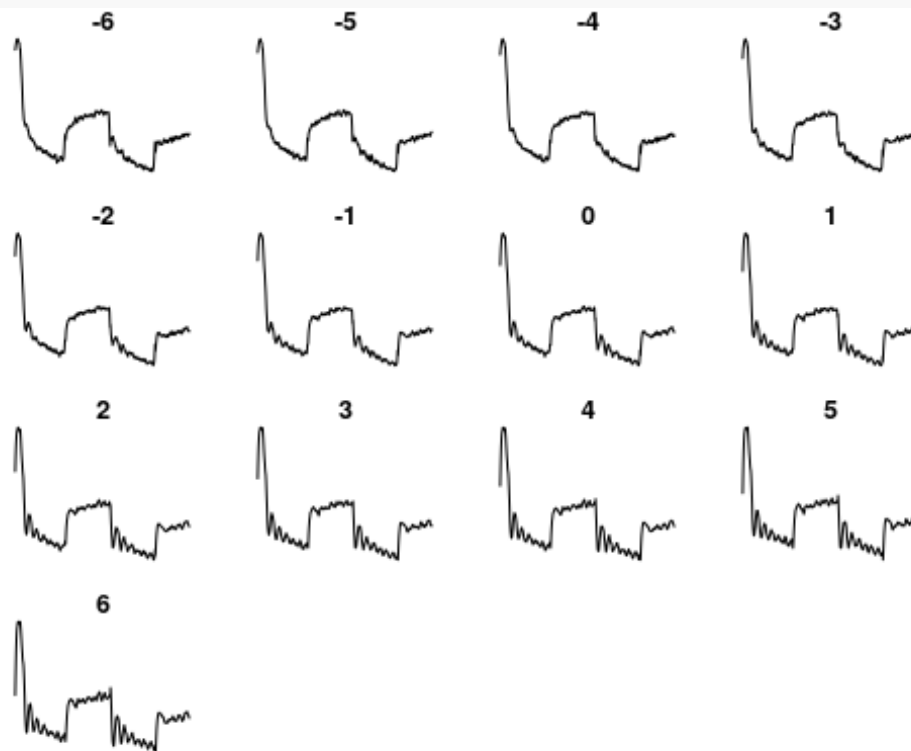
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo02$O21/((3.223674e-06*octo02[,48+i]^3.686095+8.338341e-01)^2)~
octo02$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
```

```
octo02$O21adj=octo02$O21/((3.223674e-06*octo02[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo02$O21adj
```

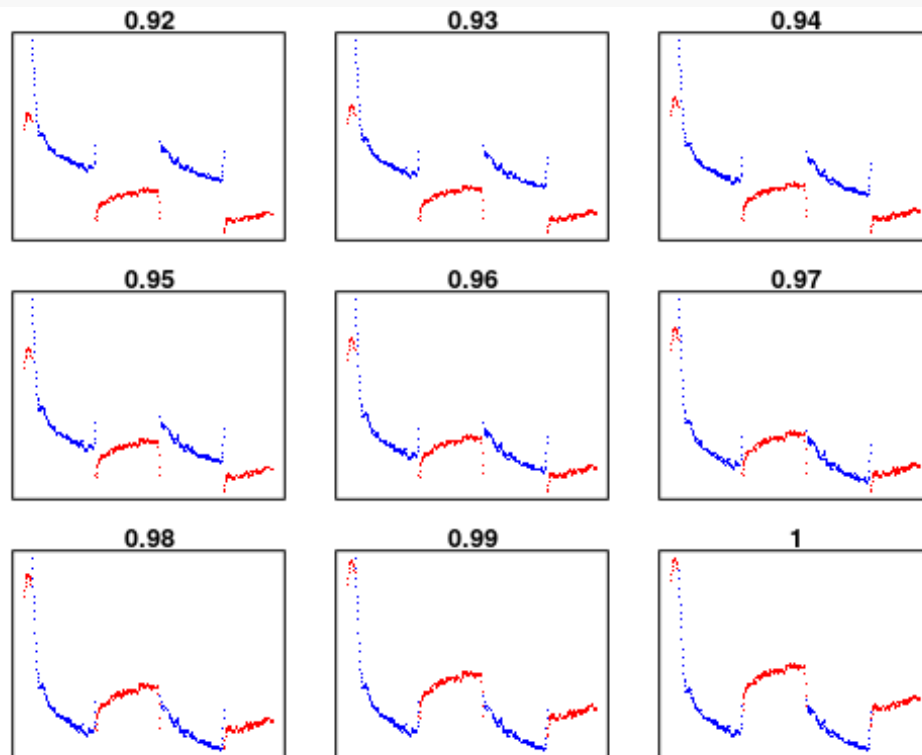
```
  temp[octo02$ambient1>20]=temp[octo02$ambient1>20]/i
```

```
  plot(temp[octo02$ambient1<20]~octo02$times[octo02$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo02$ambient1>20]~octo02$times[octo02$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



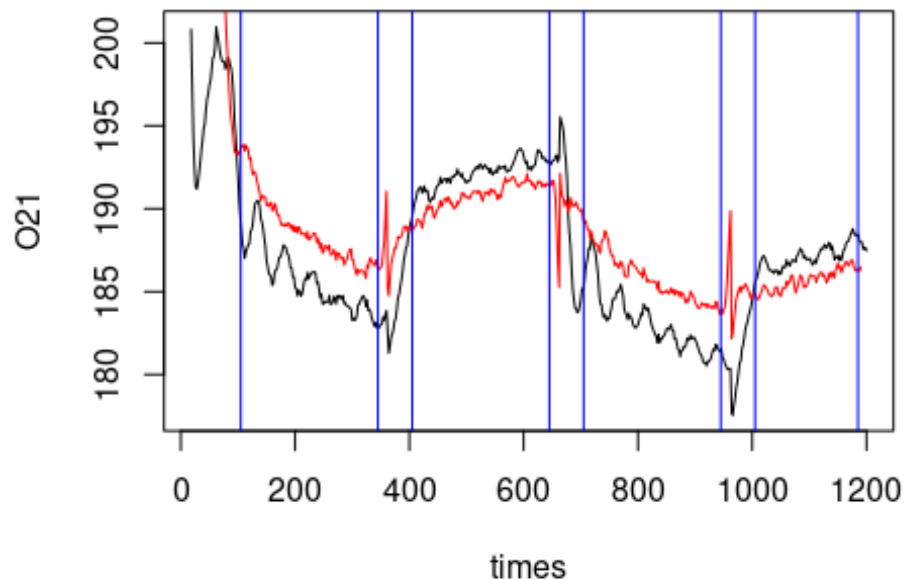
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction


```

light.off=0.97
octo02$O21adj2=octo02$O21adj
octo02$O21adj2[octo02$ambient1>20]=octo02$O21adj2[octo02$ambient1>20]/light.off
plot(O21~times,data=octo02,type="l")
lines(O21adj2~times,data=octo02,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo02$times>60+trim.begin)),
min(which(octo02$times>360-trim.end)),
min(which(octo02$times>360+trim.begin)),
min(which(octo02$times>660-trim.end)),
min(which(octo02$times>660-trim.begin)),
min(which(octo02$times>960-trim.end)),
min(which(octo02$times>960-trim.begin)),
min(which(octo02$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[2]=2

```

```

octo02.1=
(octo02$O21adj2[breaks[1]:breaks[2]]-
octo02$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[2]=mean(octo02.1)*1000
resp$on1.o2[2]=mean(octo02$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[2]=mean(octo02$tempv5_filt[breaks[1]:breaks[2]])

octo02.2=
(octo02$O21adj2[breaks[3]:breaks[4]]-
 octo02$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[2]=mean(octo02.2)*1000
resp$off1.o2[2]=mean(octo02$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[2]=mean(octo02$tempv5_filt[breaks[3]:breaks[4]])

octo02.3=
(octo02$O21adj2[breaks[5]:breaks[6]]-
 octo02$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[2]=mean(octo02.3)*1000
resp$on2.o2[2]=mean(octo02$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[2]=mean(octo02$tempv5_filt[breaks[5]:breaks[6]])

octo02.4=
(octo02$O21adj2[breaks[7]:breaks[8]]-
 octo02$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[2]=mean(octo02.4)*1000
resp$off2.o2[2]=mean(octo02$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[2]=mean(octo02$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 03

Loading data

```

octo03=read.pyro('Octo_04r_08-18-2020.txt')
octo03.temp=read.csv('Octo_04r_08-18-2020_temp.txt')
octo03.temp$times=octo03.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo03=cbind(octo03[,1:28],matrix(nrow=nrow(octo03),ncol=26))
colnames(octo03)=c(colnames(octo03)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo03)){

octo03[i, 35+j]=mean(octo03.temp$temp[which.min(abs(octo03.temp$times-oc
to03$times[i]+j))])
  }
}

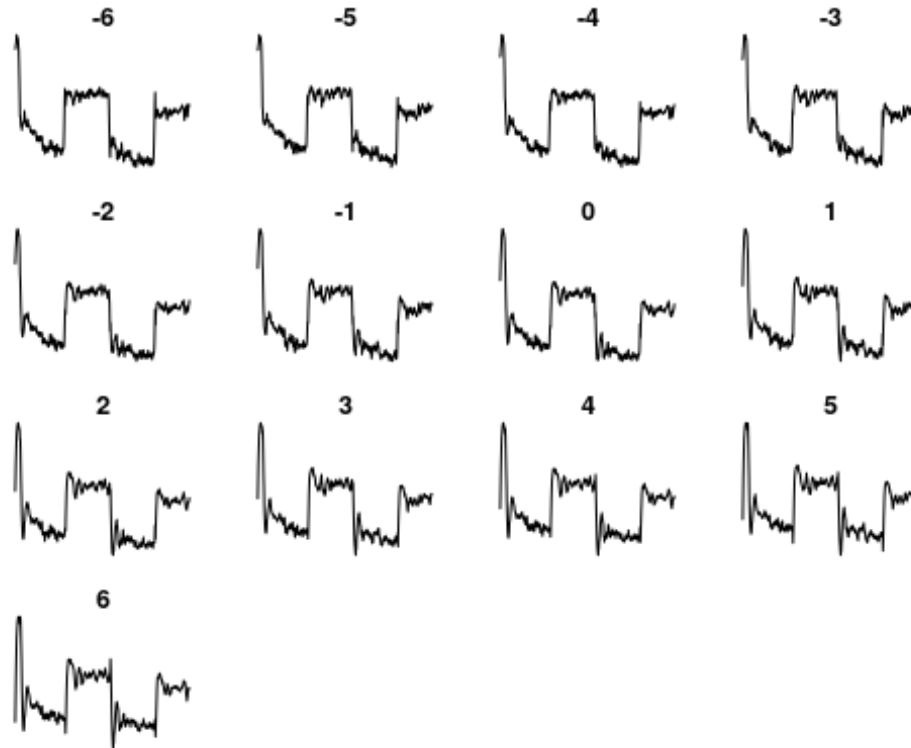
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo03[, 48+i]=filter(octo03[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo03$O21/((3.223674e-06*octo03[, 48+i]^3.686095+8.338341e-01)^2)~
octo03$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-5
```

```
octo03$O21adj=octo03$O21/((3.223674e-06*octo03[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo03$O21adj
```

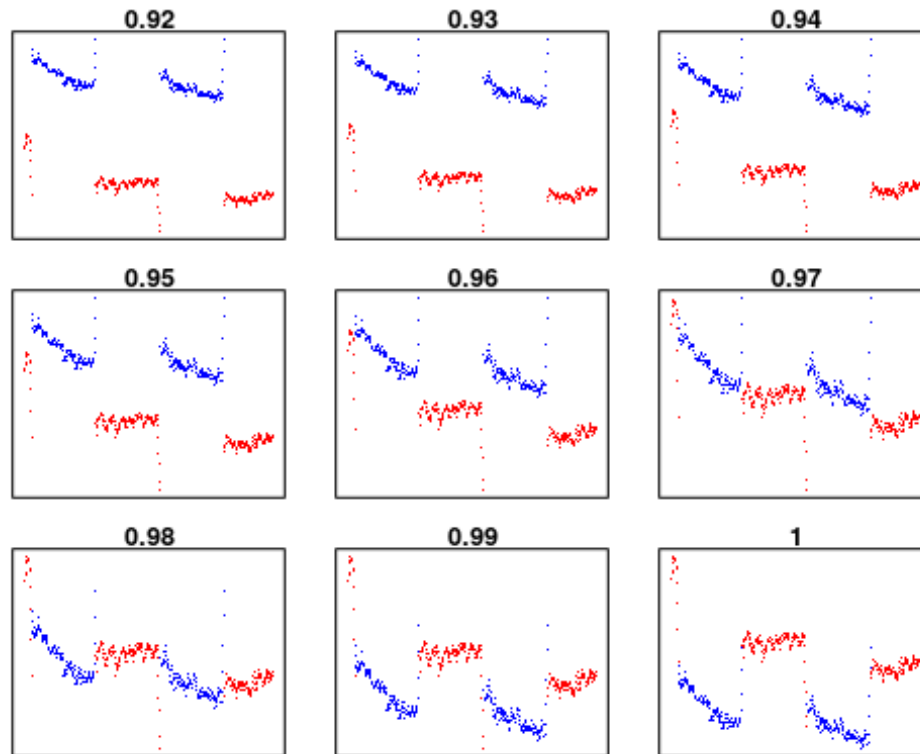
```
  temp[octo03$ambient1>20]=temp[octo03$ambient1>20]/i
```

```
  plot(temp[octo03$ambient1<20]~octo03$times[octo03$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo03$ambient1>20]~octo03$times[octo03$ambient1>20],pch=".",
  ",col="blue")
```

```
  box()
```

```
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.97
```

```
octo03$O21adj2=octo03$O21adj
```

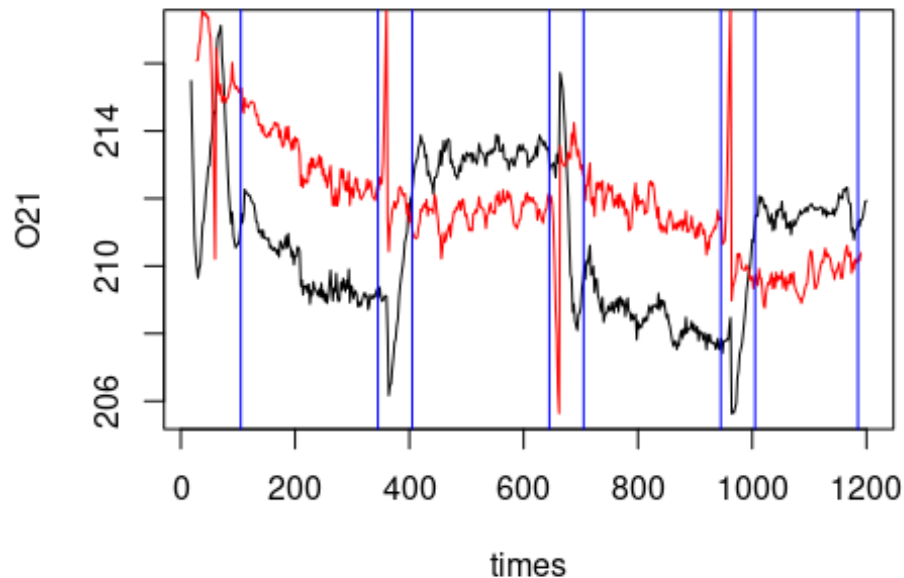
```
octo03$O21adj2[octo03$ambient1>20]=octo03$O21adj2[octo03$ambient1>20]/light.off
```

```
plot(O21~times,data=octo03,type="l")
```

```
lines(O21adj2~times,data=octo03,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo03$times>60+trim.begin)),
         min(which(octo03$times>360-trim.end)),
         min(which(octo03$times>360+trim.begin)),
         min(which(octo03$times>660-trim.end)),
         min(which(octo03$times>660-trim.begin)),
         min(which(octo03$times>960-trim.end)),
         min(which(octo03$times>960-trim.begin)),
         min(which(octo03$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[3]=3
```

```
octo03.1=
```

```
(octo03$O21adj2[breaks[1]:breaks[2]]-
 octo03$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[3]=mean(octo03.1)*1000
```

```
resp$on1.o2[3]=mean(octo03$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[3]=mean(octo03$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo03.2=
```

```
(octo03$O21adj2[breaks[3]:breaks[4]]-
 octo03$O21adj2[breaks[3]:breaks[4]+1])*
```

```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[3]=mean(octo03.2)*1000
resp$off1.o2[3]=mean(octo03$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[3]=mean(octo03$tempv5_filt[breaks[3]:breaks[4]])

octo03.3=
(octo03$O21adj2[breaks[5]:breaks[6]]-
 octo03$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[3]=mean(octo03.3)*1000
resp$on2.o2[3]=mean(octo03$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[3]=mean(octo03$tempv5_filt[breaks[5]:breaks[6]])

octo03.4=
(octo03$O21adj2[breaks[7]:breaks[8]]-
 octo03$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[3]=mean(octo03.4)*1000
resp$off2.o2[3]=mean(octo03$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[3]=mean(octo03$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 04

Loading data

```

octo04=read.pyro('Octo05_08-14-2020.txt')
octo04.temp=read.csv('Octo05_08-14-2020_temp.txt')
octo04.temp$times=octo04.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo04=cbind(octo04[,1:28],matrix(nrow=nrow(octo04),ncol=26))
colnames(octo04)=c(colnames(octo04)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",

"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",

"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo04)){
    octo04[i,35+j]=mean(octo04.temp$temp[which.min(abs(octo04.temp$times-oc
to04$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo04[,48+i]=filter(octo04[,35+i],coeff)
}

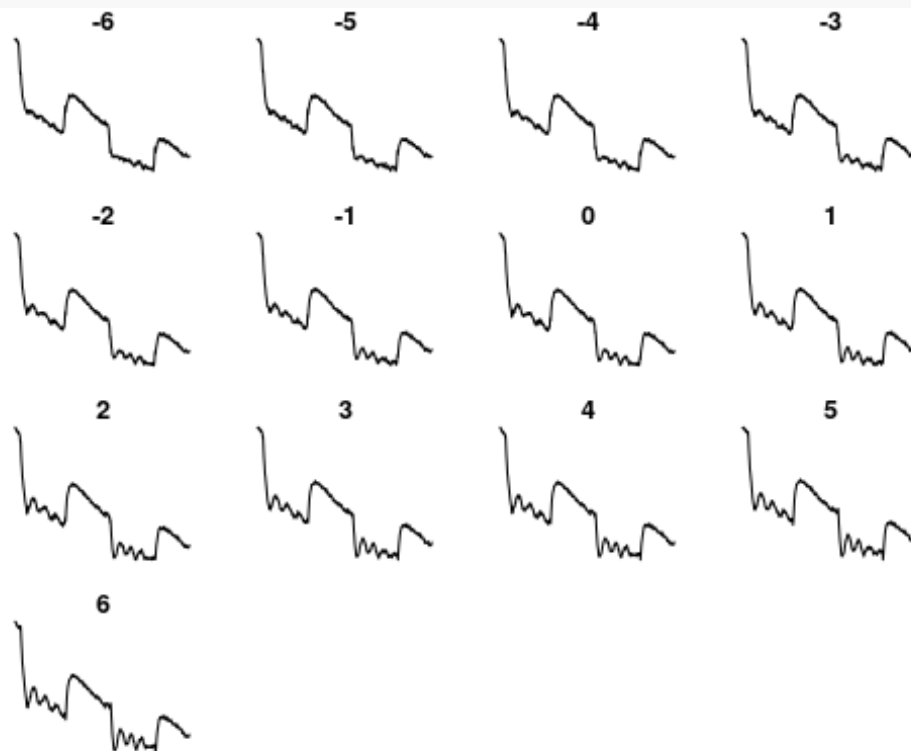
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo04$O21/((3.223674e-06*octo04[,48+i]^3.686095+8.338341e-01)^2)~
octo04$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-5
```

```
octo04$O21adj=octo04$O21/((3.223674e-06*octo04[,48+time.off]^3.686095+.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo04$O21adj
```

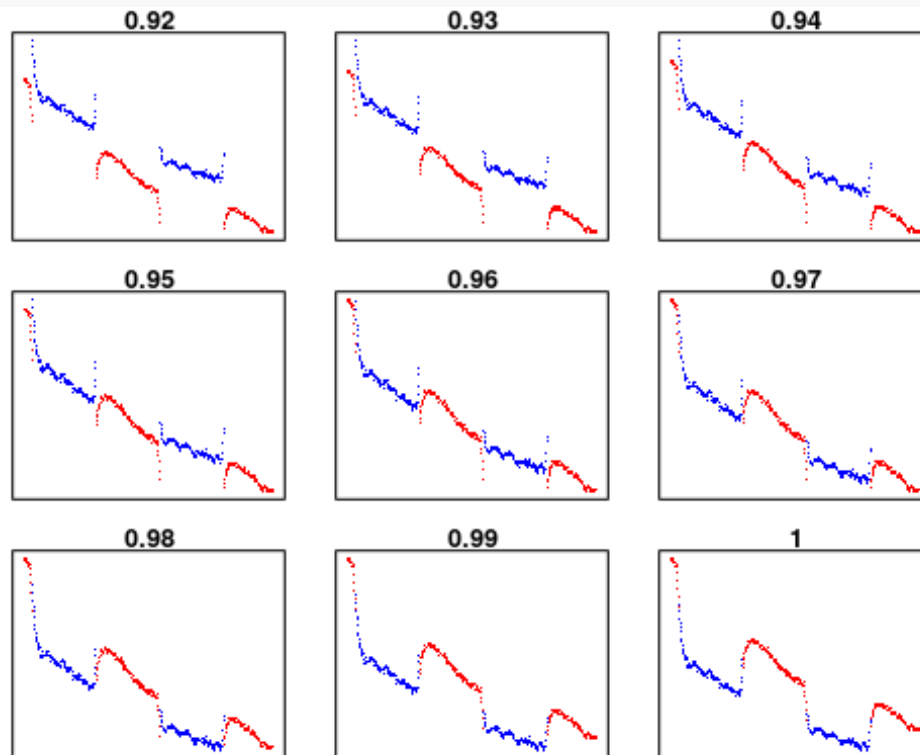
```
  temp[octo04$ambient1>20]=temp[octo04$ambient1>20]/i
```

```
  plot(temp[octo04$ambient1<20]~octo04$times[octo04$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo04$ambient1>20]~octo04$times[octo04$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



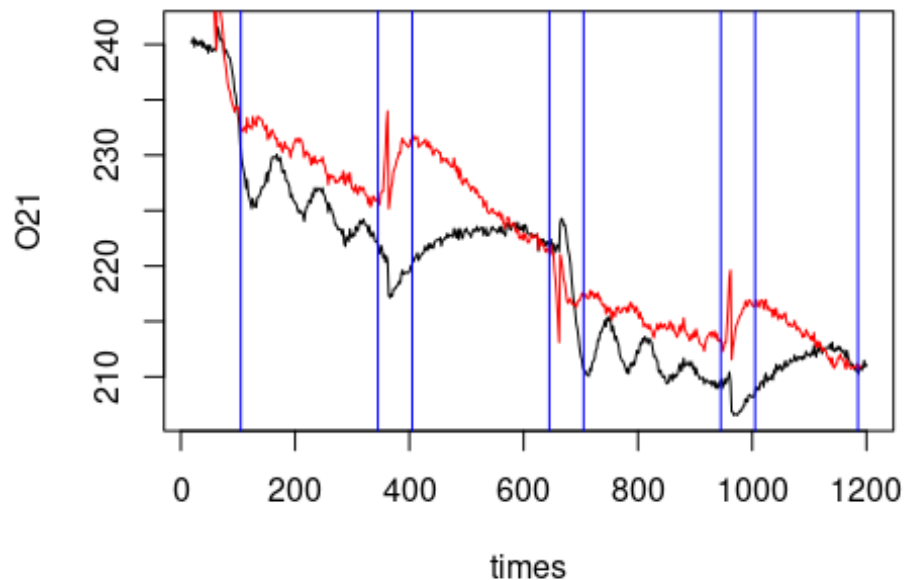
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.97
octo04$O21adj2=octo04$O21adj
octo04$O21adj2[octo04$ambient1>20]=octo04$O21adj2[octo04$ambient1>20]/light.off
plot(O21~times,data=octo04,type="l")
lines(O21adj2~times,data=octo04,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo04$times>60+trim.begin)),
min(which(octo04$times>360-trim.end)),
min(which(octo04$times>360+trim.begin)),
min(which(octo04$times>660-trim.end)),
min(which(octo04$times>660-trim.begin)),
min(which(octo04$times>960-trim.end)),
min(which(octo04$times>960-trim.begin)),
min(which(octo04$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[4]=4

```

```

octo04.1=
(octo04$O21adj2[breaks[1]:breaks[2]]-
octo04$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[4]=mean(octo04.1)*1000
resp$on1.o2[4]=mean(octo04$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[4]=mean(octo04$tempv5_filt[breaks[1]:breaks[2]])

octo04.2=
(octo04$O21adj2[breaks[3]:breaks[4]]-
 octo04$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[4]=mean(octo04.2)*1000
resp$off1.o2[4]=mean(octo04$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[4]=mean(octo04$tempv5_filt[breaks[3]:breaks[4]])

octo04.3=
(octo04$O21adj2[breaks[5]:breaks[6]]-
 octo04$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[4]=mean(octo04.3)*1000
resp$on2.o2[4]=mean(octo04$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[4]=mean(octo04$tempv5_filt[breaks[5]:breaks[6]])

octo04.4=
(octo04$O21adj2[breaks[7]:breaks[8]]-
 octo04$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[4]=mean(octo04.4)*1000
resp$off2.o2[4]=mean(octo04$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[4]=mean(octo04$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 05

Loading data

```

octo05=read.pyro('Octo06_08-14-2020.txt')
octo05.temp=read.csv('Octo06_08-14-2020_temp.txt')
octo05.temp$times=octo05.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo05=cbind(octo05[,1:28],matrix(nrow=nrow(octo05),ncol=26))
colnames(octo05)=c(colnames(octo05)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo05)){

octo05[i, 35+j]=mean(octo05.temp$temp[which.min(abs(octo05.temp$times-oc
to05$times[i]+j))])
  }
}

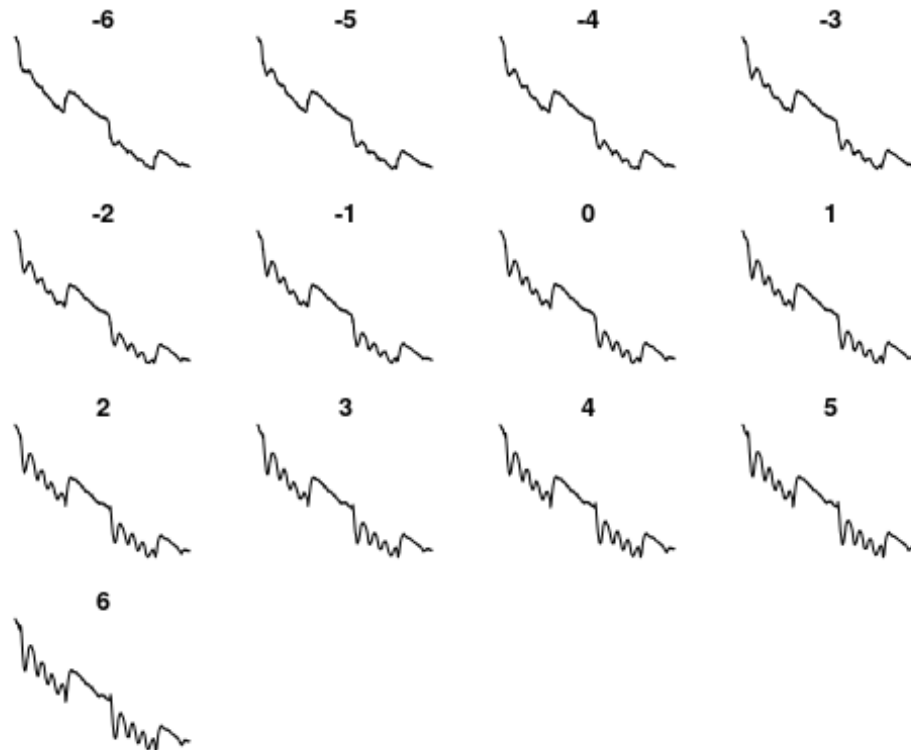
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo05[, 48+i]=filter(octo05[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo05$O21/((3.223674e-06*octo05[, 48+i]^3.686095+8.338341e-01)^2)~
octo05$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-6
octo05$O21adj=octo05$O21/((3.223674e-06*octo05[,48+time.off]^3.686095+8
.338341e-01)^2)
```

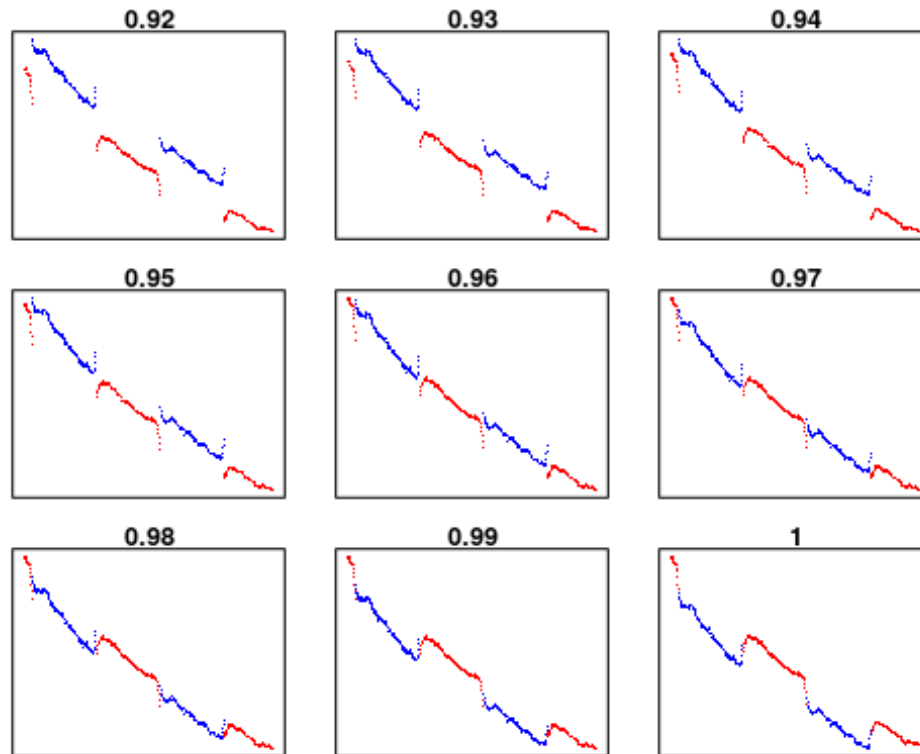
Correcting for light

```
par(mfrow=c(3,3))
par(mar=c(1,1,1,1))

for (i in seq(from=0.92,to=1,by=.01)){
  temp=octo05$O21adj
  temp[octo05$ambient1>20]=temp[octo05$ambient1>20]/i

  plot(temp[octo05$ambient1<20]~octo05$times[octo05$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))

  points(temp[octo05$ambient1>20]~octo05$times[octo05$ambient1>20],pch=".",
  col="blue")
  box()
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.98
```

```
octo05$O21adj2=octo05$O21adj
```

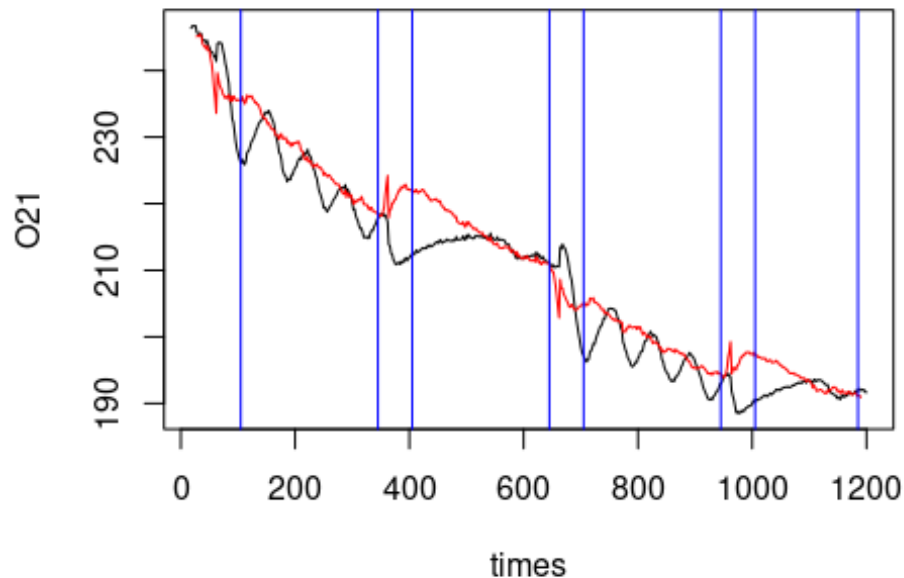
```
octo05$O21adj2[octo05$ambient1>20]=octo05$O21adj2[octo05$ambient1>20]/light.off
```

```
plot(O21~times,data=octo05,type="l")
```

```
lines(O21adj2~times,data=octo05,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo05$times>60+trim.begin)),
         min(which(octo05$times>360-trim.end)),
         min(which(octo05$times>360+trim.begin)),
         min(which(octo05$times>660-trim.end)),
         min(which(octo05$times>660-trim.begin)),
         min(which(octo05$times>960-trim.end)),
         min(which(octo05$times>960-trim.begin)),
         min(which(octo05$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[5]=5
```

```
octo05.1=
```

```
(octo05$O21adj2[breaks[1]:breaks[2]]-
 octo05$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[5]=mean(octo05.1)*1000
```

```
resp$on1.o2[5]=mean(octo05$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[5]=mean(octo05$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo05.2=
```

```
(octo05$O21adj2[breaks[3]:breaks[4]]-
 octo05$O21adj2[breaks[3]:breaks[4]+1])*
```

```

      (3600/2)*
      0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[5]=mean(octo05.2)*1000
resp$off1.o2[5]=mean(octo05$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[5]=mean(octo05$tempv5_filt[breaks[3]:breaks[4]])

octo05.3=
(octo05$O21adj2[breaks[5]:breaks[6]]-
 octo05$O21adj2[breaks[5]:breaks[6]+1])*
  (3600/2)*
  0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[5]=mean(octo05.3)*1000
resp$on2.o2[5]=mean(octo05$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[5]=mean(octo05$tempv5_filt[breaks[5]:breaks[6]])

octo05.4=
(octo05$O21adj2[breaks[7]:breaks[8]]-
 octo05$O21adj2[breaks[7]:breaks[8]+1])*
  (3600/2)*
  0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[5]=mean(octo05.4)*1000
resp$off2.o2[5]=mean(octo05$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[5]=mean(octo05$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 06

Loading data

```

octo06=read.pyro('Octo07_08-14-2020.txt')
octo06.temp=read.csv('Octo07_08-14-2020_temp.txt')
octo06.temp$times=octo06.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo06=cbind(octo06[,1:28],matrix(nrow=nrow(octo06),ncol=26))
colnames(octo06)=c(colnames(octo06)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",

"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",

"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```



```

for (j in -6:6){
  for (i in 1:nrow(octo06)){
    octo06[i,35+j]=mean(octo06.temp$temp[which.min(abs(octo06.temp$times-oc
to06$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo06[,48+i]=filter(octo06[,35+i],coeff)
}

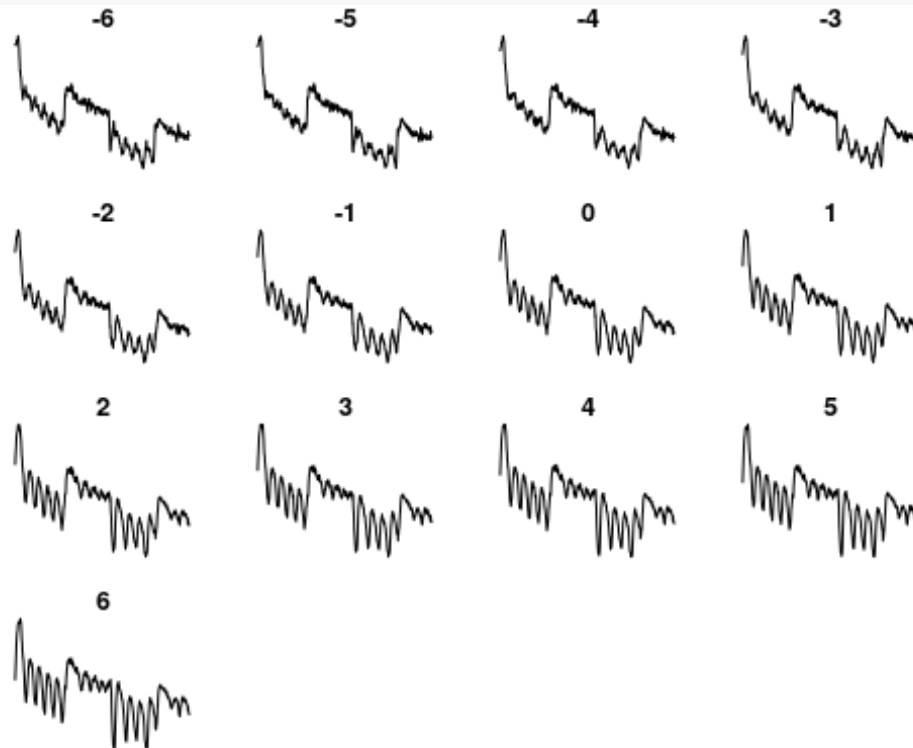
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo06$O21/((3.223674e-06*octo06[,48+i]^3.686095+8.338341e-01)^2)~
octo06$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the "temp.off" parameter below.

Correcting for temperature using correct time offset

```
time.off=-5
```

```
octo06$O21adj=octo06$O21/((3.223674e-06*octo06[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo06$O21adj
```

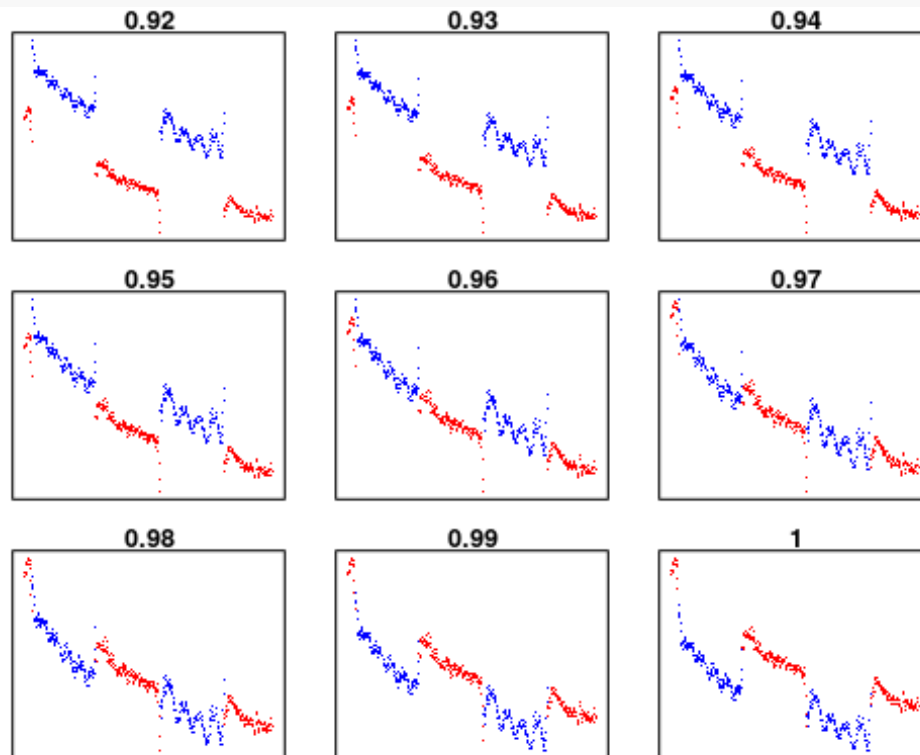
```
  temp[octo06$ambient1>20]=temp[octo06$ambient1>20]/i
```

```
  plot(temp[octo06$ambient1<20]~octo06$times[octo06$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo06$ambient1>20]~octo06$times[octo06$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



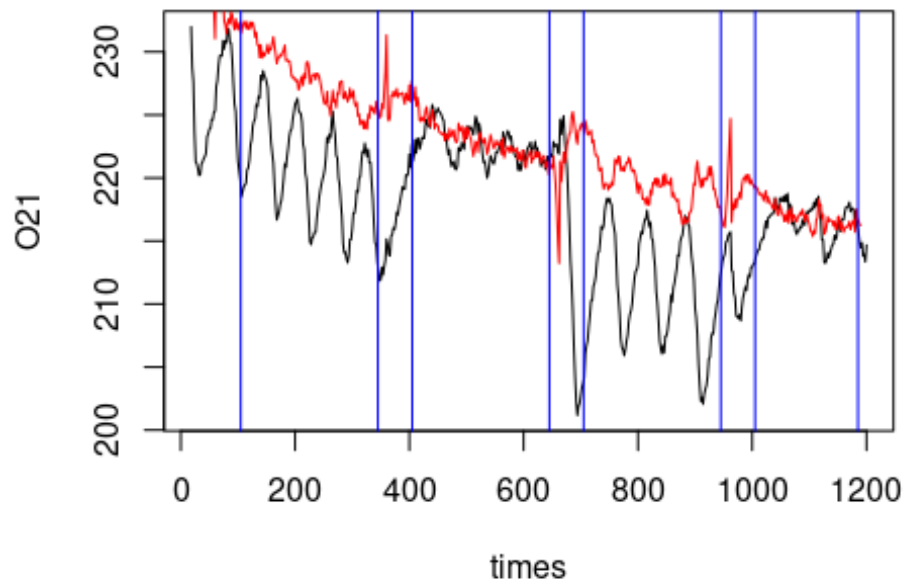
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.97
octo06$O21adj2=octo06$O21adj
octo06$O21adj2[octo06$ambient1>20]=octo06$O21adj2[octo06$ambient1>20]/light.off
plot(O21~times,data=octo06,type="l")
lines(O21adj2~times,data=octo06,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo06$times>60+trim.begin)),
min(which(octo06$times>360-trim.end)),
min(which(octo06$times>360+trim.begin)),
min(which(octo06$times>660-trim.end)),
min(which(octo06$times>660-trim.begin)),
min(which(octo06$times>960-trim.end)),
min(which(octo06$times>960-trim.begin)),
min(which(octo06$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[6]=6

```

```

octo06.1=
(octo06$O21adj2[breaks[1]:breaks[2]]-
octo06$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[6]=mean(octo06.1)*1000
resp$on1.o2[6]=mean(octo06$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[6]=mean(octo06$tempv5_filt[breaks[1]:breaks[2]])

octo06.2=
(octo06$O21adj2[breaks[3]:breaks[4]]-
 octo06$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[6]=mean(octo06.2)*1000
resp$off1.o2[6]=mean(octo06$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[6]=mean(octo06$tempv5_filt[breaks[3]:breaks[4]])

octo06.3=
(octo06$O21adj2[breaks[5]:breaks[6]]-
 octo06$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[6]=mean(octo06.3)*1000
resp$on2.o2[6]=mean(octo06$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[6]=mean(octo06$tempv5_filt[breaks[5]:breaks[6]])

octo06.4=
(octo06$O21adj2[breaks[7]:breaks[8]]-
 octo06$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[6]=mean(octo06.4)*1000
resp$off2.o2[6]=mean(octo06$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[6]=mean(octo06$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 07

Loading data

```

octo07=read.pyro('Octo_09r_08-19-2020.txt')
octo07.temp=read.csv('Octo_09r_08-19-2020_temp.txt')
octo07.temp$times=octo07.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo07=cbind(octo07[,1:28],matrix(nrow=nrow(octo07),ncol=26))
colnames(octo07)=c(colnames(octo07)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo07)){

octo07[i, 35+j]=mean(octo07.temp$temp[which.min(abs(octo07.temp$times-oc
to07$times[i]+j))])
  }
}

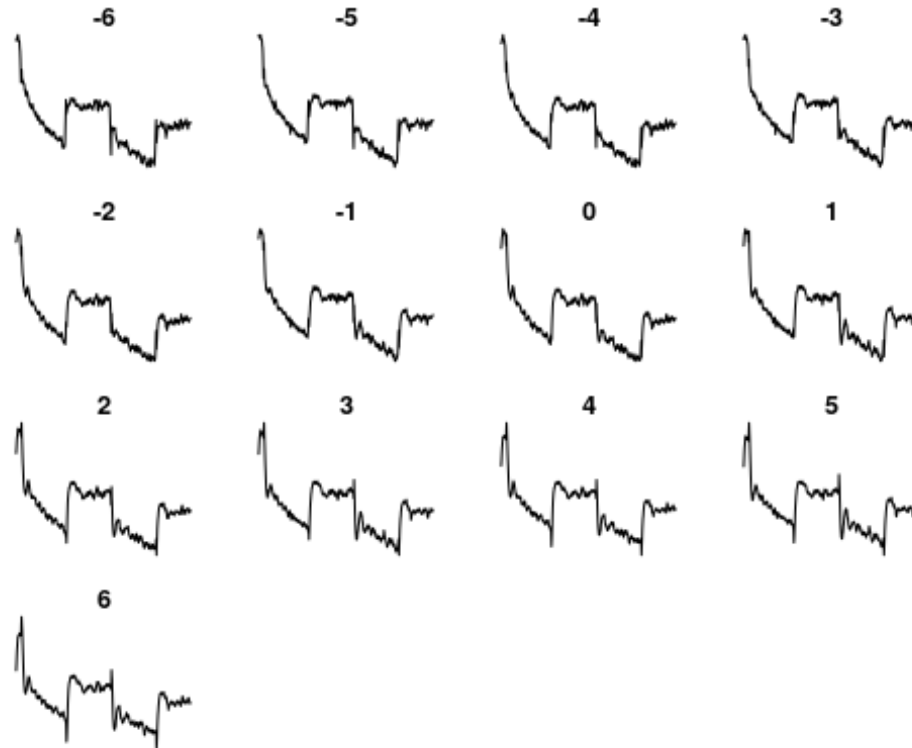
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo07[, 48+i]=filter(octo07[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo07$O21/((3.223674e-06*octo07[, 48+i]^3.686095+8.338341e-01)^2)~
octo07$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-2
```

```
octo07$O21adj=octo07$O21/((3.223674e-06*octo07[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo07$O21adj
```

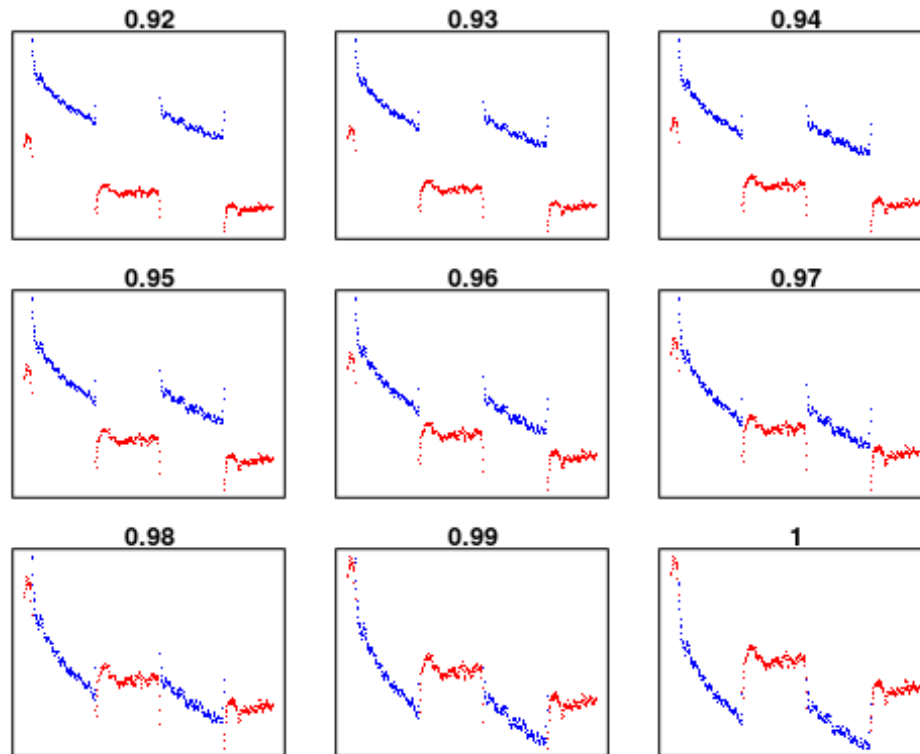
```
  temp[octo07$ambient1>20]=temp[octo07$ambient1>20]/i
```

```
  plot(temp[octo07$ambient1<20]~octo07$times[octo07$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo07$ambient1>20]~octo07$times[octo07$ambient1>20],pch=".",
  col="blue")
```

```
  box()
```

```
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.98
```

```
octo07$O21adj2=octo07$O21adj
```

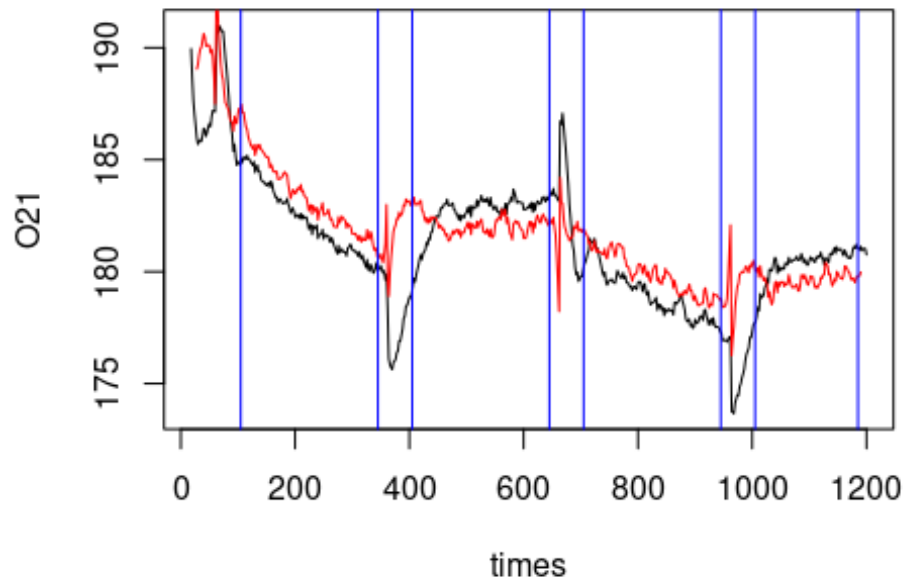
```
octo07$O21adj2[octo07$ambient1>20]=octo07$O21adj2[octo07$ambient1>20]/light.off
```

```
plot(O21~times,data=octo07,type="l")
```

```
lines(O21adj2~times,data=octo07,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo07$times>60+trim.begin)),
          min(which(octo07$times>360-trim.end)),
          min(which(octo07$times>360+trim.begin)),
          min(which(octo07$times>660-trim.end)),
          min(which(octo07$times>660-trim.begin)),
          min(which(octo07$times>960-trim.end)),
          min(which(octo07$times>960-trim.begin)),
          min(which(octo07$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[7]=7
```

```
octo07.1=
```

```
(octo07$O21adj2[breaks[1]:breaks[2]]-
  octo07$O21adj2[breaks[1]:breaks[2]+1])*
  (3600/2)*
  0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[7]=mean(octo07.1)*1000
```

```
resp$on1.o2[7]=mean(octo07$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[7]=mean(octo07$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo07.2=
```

```
(octo07$O21adj2[breaks[3]:breaks[4]]-
  octo07$O21adj2[breaks[3]:breaks[4]+1])*
```



```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[7]=mean(octo07.2)*1000
resp$off1.o2[7]=mean(octo07$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[7]=mean(octo07$tempv5_filt[breaks[3]:breaks[4]])

octo07.3=
(octo07$O21adj2[breaks[5]:breaks[6]]-
 octo07$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[7]=mean(octo07.3)*1000
resp$on2.o2[7]=mean(octo07$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[7]=mean(octo07$tempv5_filt[breaks[5]:breaks[6]])

octo07.4=
(octo07$O21adj2[breaks[7]:breaks[8]]-
 octo07$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[7]=mean(octo07.4)*1000
resp$off2.o2[7]=mean(octo07$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[7]=mean(octo07$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 08

Loading data

```

octo08=read.pyro('Octo_10r_08-19-2020.txt')
octo08.temp=read.csv('Octo_10r_08-19-2020_temp.txt')
octo08.temp$times=octo08.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo08=cbind(octo08[,1:28],matrix(nrow=nrow(octo08),ncol=26))
colnames(octo08)=c(colnames(octo08)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",

"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",

"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo08)){
    octo08[i,35+j]=mean(octo08.temp$temp[which.min(abs(octo08.temp$times-oc
to08$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo08[,48+i]=filter(octo08[,35+i],coeff)
}

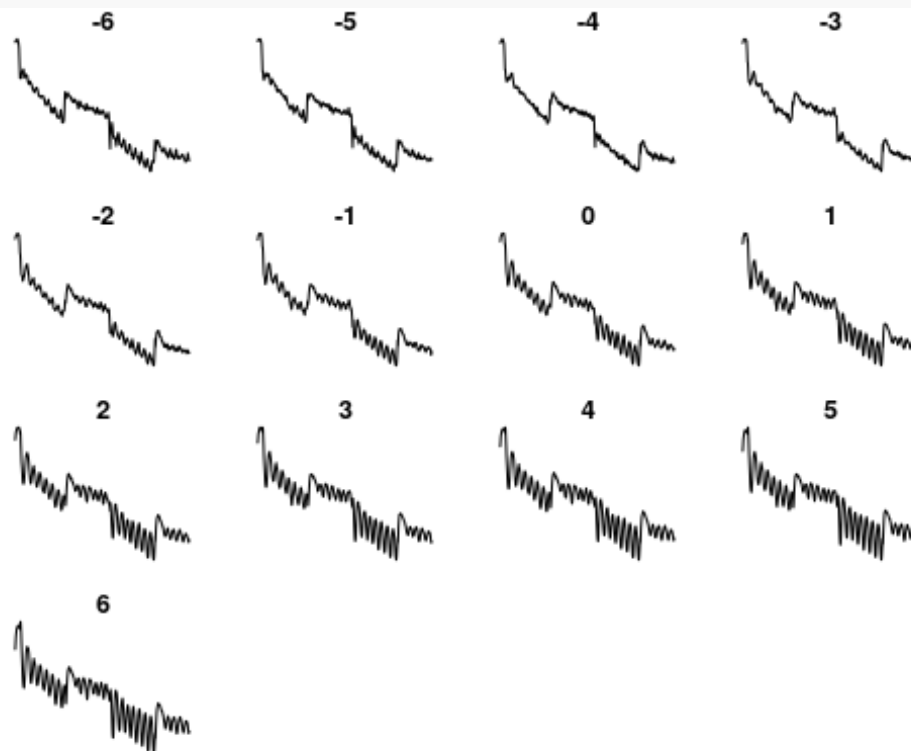
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo08$O21/((3.223674e-06*octo08[,48+i]^3.686095+8.338341e-01)^2)~
octo08$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
```

```
octo08$O21adj=octo08$O21/((3.223674e-06*octo08[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo08$O21adj
```

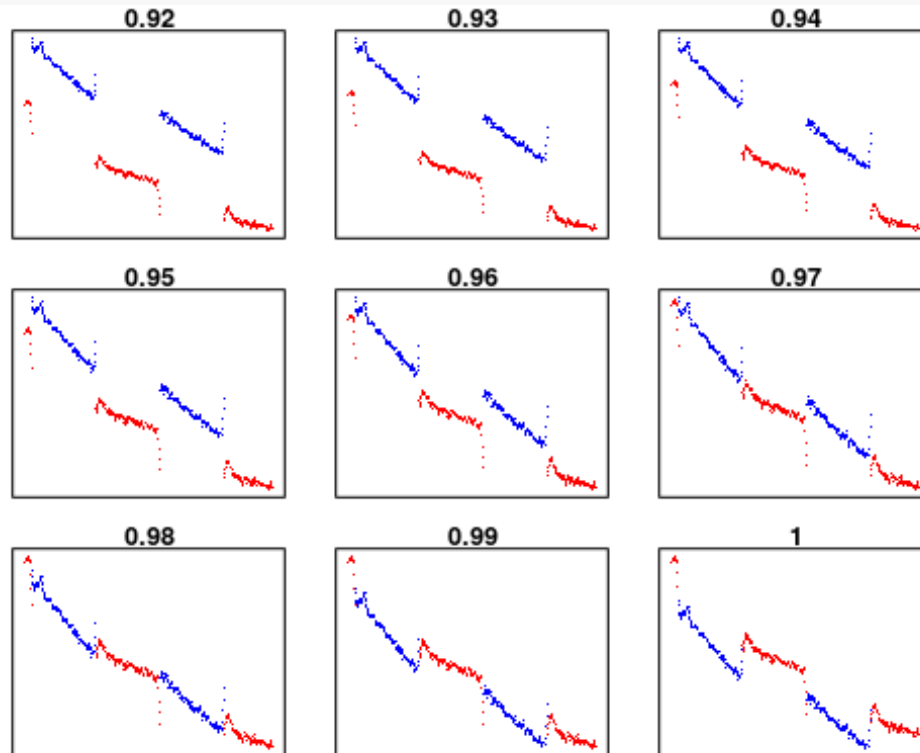
```
  temp[octo08$ambient1>20]=temp[octo08$ambient1>20]/i
```

```
  plot(temp[octo08$ambient1<20]~octo08$times[octo08$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo08$ambient1>20]~octo08$times[octo08$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



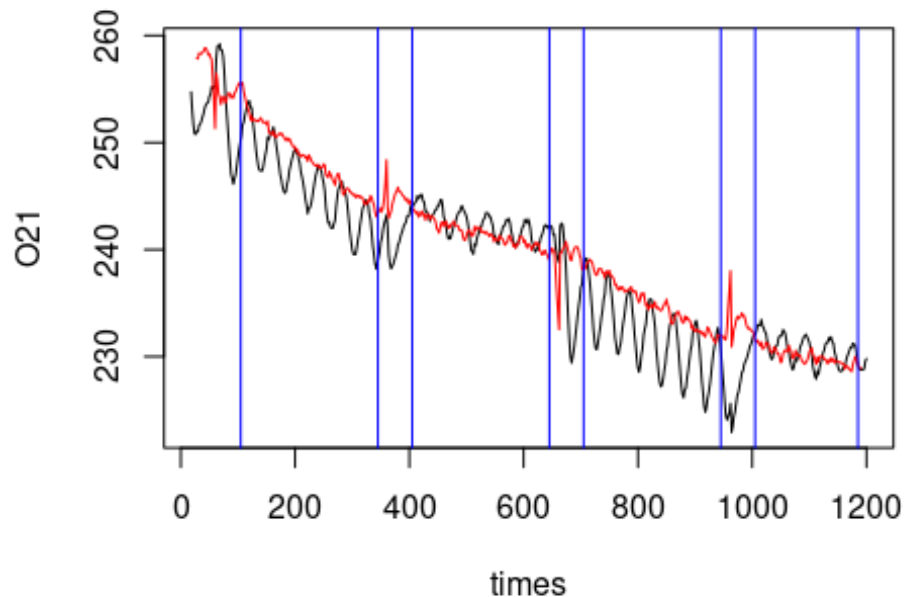
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.98
octo08$O21adj2=octo08$O21adj
octo08$O21adj2[octo08$ambient1>20]=octo08$O21adj2[octo08$ambient1>20]/light.off
plot(O21~times,data=octo08,type="l")
lines(O21adj2~times,data=octo08,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo08$times>60+trim.begin)),
min(which(octo08$times>360-trim.end)),
min(which(octo08$times>360+trim.begin)),
min(which(octo08$times>660-trim.end)),
min(which(octo08$times>660-trim.begin)),
min(which(octo08$times>960-trim.end)),
min(which(octo08$times>960-trim.begin)),
min(which(octo08$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[8]=8

```

```

octo08.1=
(octo08$O21adj2[breaks[1]:breaks[2]]-
octo08$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[8]=mean(octo08.1)*1000
resp$on1.o2[8]=mean(octo08$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[8]=mean(octo08$tempv5_filt[breaks[1]:breaks[2]])

octo08.2=
(octo08$O21adj2[breaks[3]:breaks[4]]-
 octo08$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[8]=mean(octo08.2)*1000
resp$off1.o2[8]=mean(octo08$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[8]=mean(octo08$tempv5_filt[breaks[3]:breaks[4]])

octo08.3=
(octo08$O21adj2[breaks[5]:breaks[6]]-
 octo08$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[8]=mean(octo08.3)*1000
resp$on2.o2[8]=mean(octo08$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[8]=mean(octo08$tempv5_filt[breaks[5]:breaks[6]])

octo08.4=
(octo08$O21adj2[breaks[7]:breaks[8]]-
 octo08$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[8]=mean(octo08.4)*1000
resp$off2.o2[8]=mean(octo08$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[8]=mean(octo08$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 09

Loading data

```

octo09=read.pyro('Octo11_08-16-2020.txt')
octo09.temp=read.csv('Octo11_08-16-2020_temp.txt')
octo09.temp$times=octo09.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo09=cbind(octo09[,1:28],matrix(nrow=nrow(octo09),ncol=26))
colnames(octo09)=c(colnames(octo09)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo09)){

octo09[i, 35+j]=mean(octo09.temp$temp[which.min(abs(octo09.temp$times-oc
to09$times[i]+j))])
  }
}

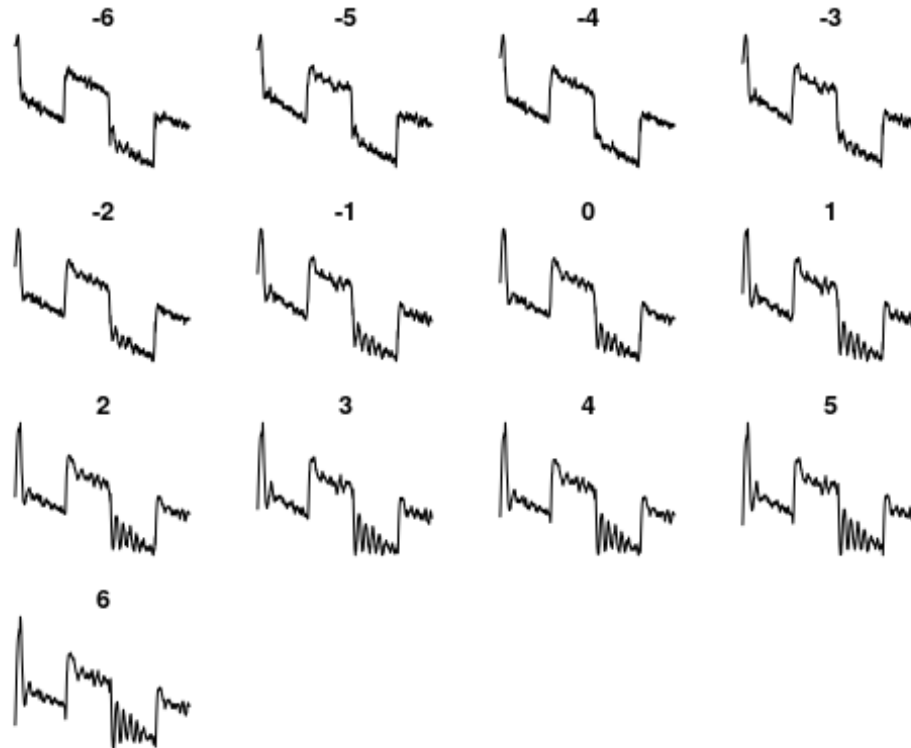
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo09[, 48+i]=filter(octo09[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo09$O21/((3.223674e-06*octo09[, 48+i]^3.686095+8.338341e-01)^2)~
octo09$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
```

```
octo09$O21adj=octo09$O21/((3.223674e-06*octo09[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo09$O21adj
```

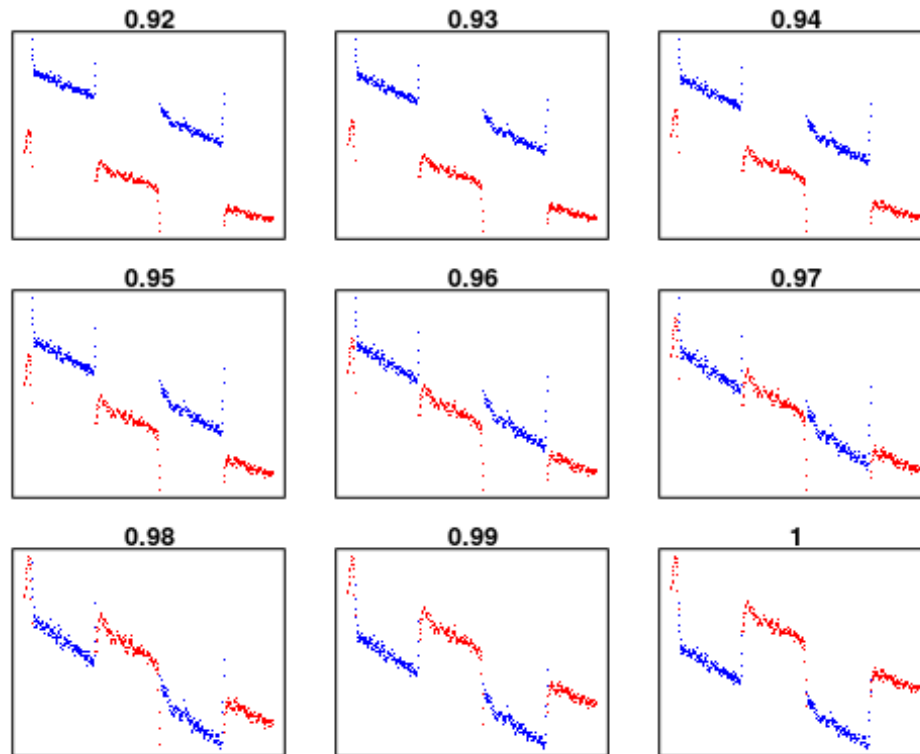
```
  temp[octo09$ambient1>20]=temp[octo09$ambient1>20]/i
```

```
  plot(temp[octo09$ambient1<20]~octo09$times[octo09$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo09$ambient1>20]~octo09$times[octo09$ambient1>20],pch=".",
  ",col="blue")
```

```
    box()
```

```
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.97
```

```
octo09$O21adj2=octo09$O21adj
```

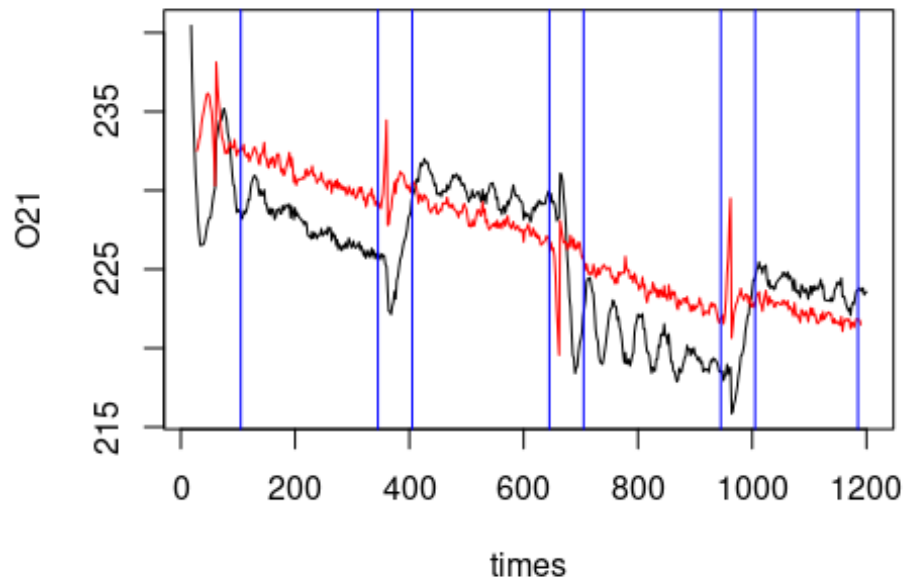
```
octo09$O21adj2[octo09$ambient1>20]=octo09$O21adj2[octo09$ambient1>20]/light.off
```

```
plot(O21~times,data=octo09,type="l")
```

```
lines(O21adj2~times,data=octo09,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```

Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo09$times>60+trim.begin)),
         min(which(octo09$times>360-trim.end)),
         min(which(octo09$times>360+trim.begin)),
         min(which(octo09$times>660-trim.end)),
         min(which(octo09$times>660-trim.begin)),
         min(which(octo09$times>960-trim.end)),
         min(which(octo09$times>960-trim.begin)),
         min(which(octo09$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[9]=9
```

```
octo09.1=
```

```
(octo09$O21adj2[breaks[1]:breaks[2]]-
 octo09$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[9]=mean(octo09.1)*1000
```

```
resp$on1.o2[9]=mean(octo09$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[9]=mean(octo09$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo09.2=
```

```
(octo09$O21adj2[breaks[3]:breaks[4]]-
 octo09$O21adj2[breaks[3]:breaks[4]+1])*
```

```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[9]=mean(octo09.2)*1000
resp$off1.o2[9]=mean(octo09$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[9]=mean(octo09$tempv5_filt[breaks[3]:breaks[4]])

octo09.3=
(octo09$O21adj2[breaks[5]:breaks[6]]-
 octo09$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[9]=mean(octo09.3)*1000
resp$on2.o2[9]=mean(octo09$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[9]=mean(octo09$tempv5_filt[breaks[5]:breaks[6]])

octo09.4=
(octo09$O21adj2[breaks[7]:breaks[8]]-
 octo09$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[9]=mean(octo09.4)*1000
resp$off2.o2[9]=mean(octo09$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[9]=mean(octo09$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 10

Loading data

```

octo10=read.pyro('Octo_13r_08-19-2020.txt')
octo10.temp=read.csv('Octo_13r_08-19-2020_temp.txt')
octo10.temp$times=octo10.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo10=cbind(octo10[,1:28],matrix(nrow=nrow(octo10),ncol=26))
colnames(octo10)=c(colnames(octo10)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",
"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",
"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo10)){
    octo10[i,35+j]=mean(octo10.temp$temp[which.min(abs(octo10.temp$times-oc
to10$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo10[,48+i]=filter(octo10[,35+i],coeff)
}

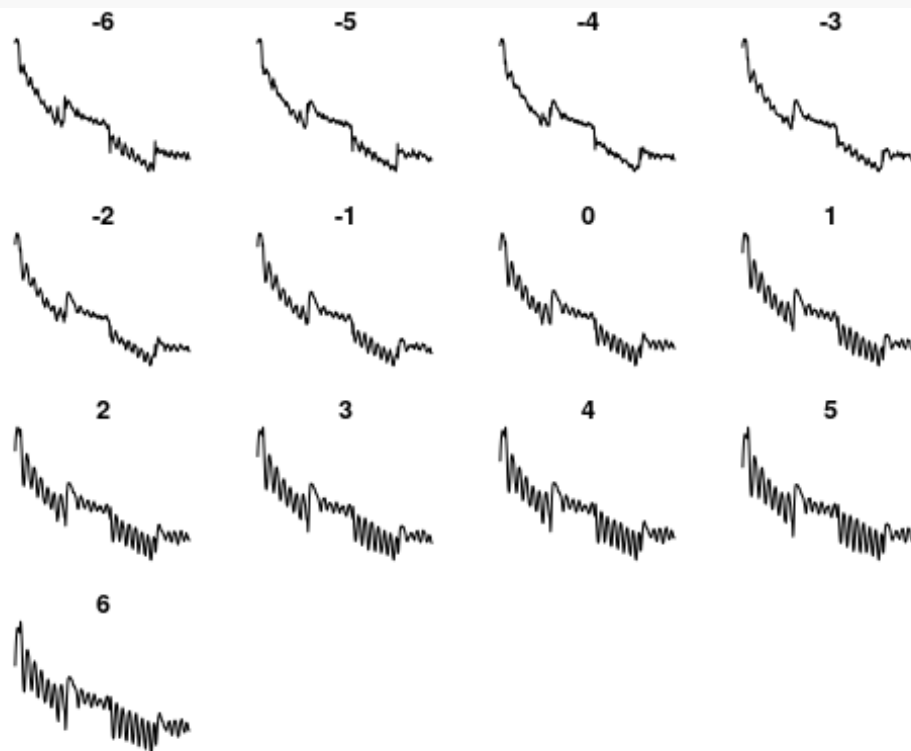
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo10$O21/((3.223674e-06*octo10[,48+i]^3.686095+8.338341e-01)^2)~
octo10$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
```

```
octo10$O21adj=octo10$O21/((3.223674e-06*octo10[,48+time.off]^3.686095+.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo10$O21adj
```

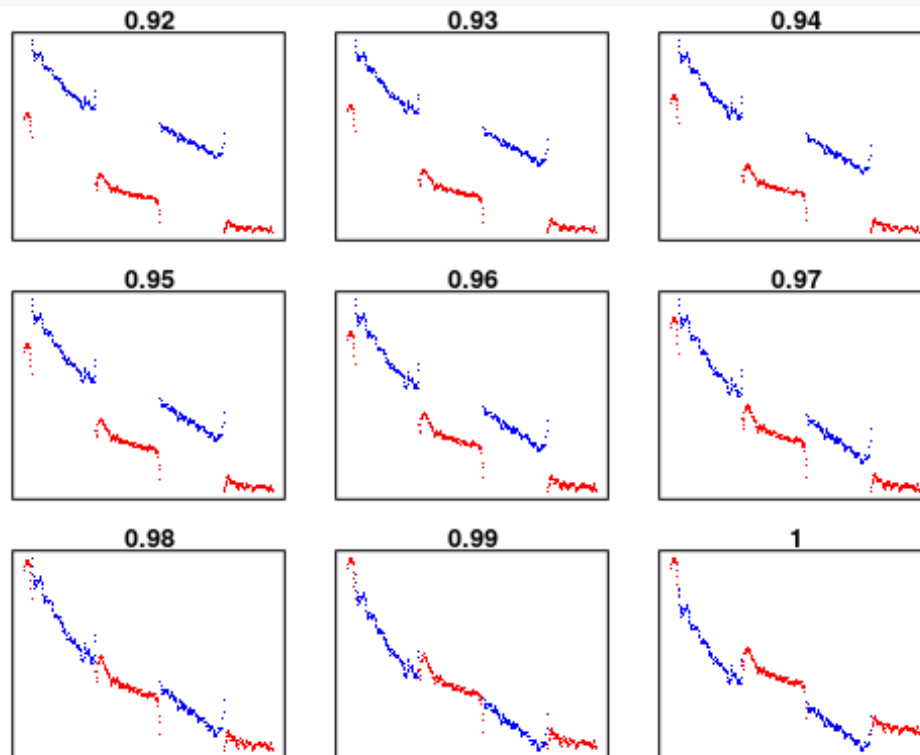
```
  temp[octo10$ambient1>20]=temp[octo10$ambient1>20]/i
```

```
  plot(temp[octo10$ambient1<20]~octo10$times[octo10$ambient1<20],pch=".",
        axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo10$ambient1>20]~octo10$times[octo10$ambient1>20],pch=".",
         col="blue")
```

```
  box()
```

```
}
```



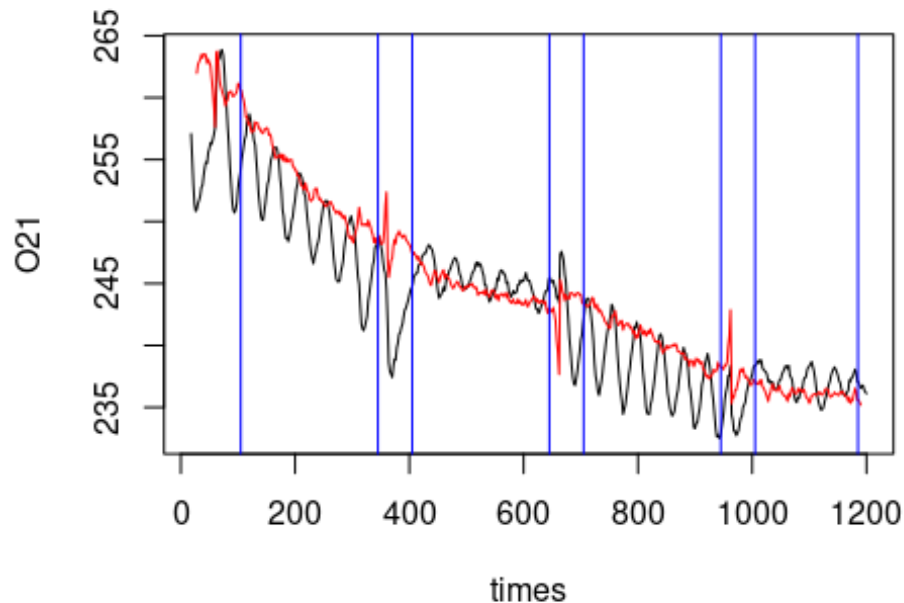
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.98
octo10$O21adj2=octo10$O21adj
octo10$O21adj2[octo10$ambient1>20]=octo10$O21adj2[octo10$ambient1>20]/light.off
plot(O21~times,data=octo10,type="l")
lines(O21adj2~times,data=octo10,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo10$times>60+trim.begin)),
min(which(octo10$times>360-trim.end)),
min(which(octo10$times>360+trim.begin)),
min(which(octo10$times>660-trim.end)),
min(which(octo10$times>660-trim.begin)),
min(which(octo10$times>960-trim.end)),
min(which(octo10$times>960-trim.begin)),
min(which(octo10$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[10]=10

```

```

octo10.1=
(octo10$O21adj2[breaks[1]:breaks[2]]-
octo10$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[10]=mean(octo10.1)*1000
resp$on1.o2[10]=mean(octo10$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[10]=mean(octo10$tempv5_filt[breaks[1]:breaks[2]])

octo10.2=
(octo10$O21adj2[breaks[3]:breaks[4]]-
 octo10$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[10]=mean(octo10.2)*1000
resp$off1.o2[10]=mean(octo10$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[10]=mean(octo10$tempv5_filt[breaks[3]:breaks[4]])

octo10.3=
(octo10$O21adj2[breaks[5]:breaks[6]]-
 octo10$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[10]=mean(octo10.3)*1000
resp$on2.o2[10]=mean(octo10$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[10]=mean(octo10$tempv5_filt[breaks[5]:breaks[6]])

octo10.4=
(octo10$O21adj2[breaks[7]:breaks[8]]-
 octo10$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[10]=mean(octo10.4)*1000
resp$off2.o2[10]=mean(octo10$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[10]=mean(octo10$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 11

Loading data

```

octo11=read.pyro('Octo_14r_08-19-2020.txt')
octo11.temp=read.csv('Octo_14r_08-19-2020_temp.txt')
octo11.temp$times=octo11.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo11=cbind(octo11[,1:28],matrix(nrow=nrow(octo11),ncol=26))
colnames(octo11)=c(colnames(octo11)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo11)){

octo11[i, 35+j]=mean(octo11.temp$temp[which.min(abs(octo11.temp$times-oc
to11$times[i]+j))])
  }
}

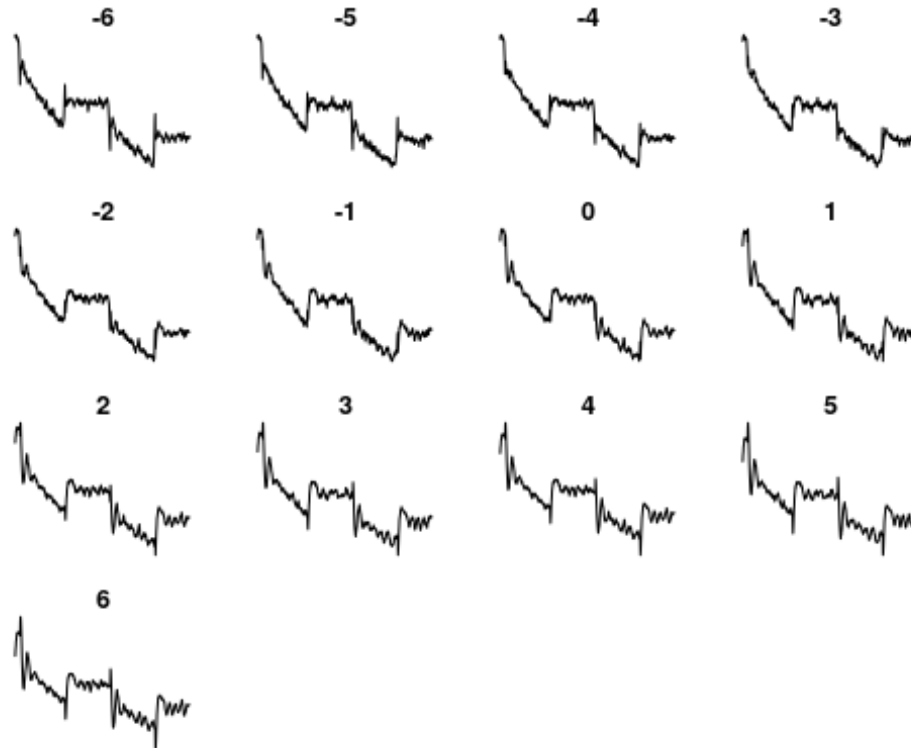
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo11[, 48+i]=filter(octo11[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo11$O21/((3.223674e-06*octo11[, 48+i]^3.686095+8.338341e-01)^2)~
octo11$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
octo11$O21adj=octo11$O21/((3.223674e-06*octo11[,48+time.off]^3.686095+8
.338341e-01)^2)
```

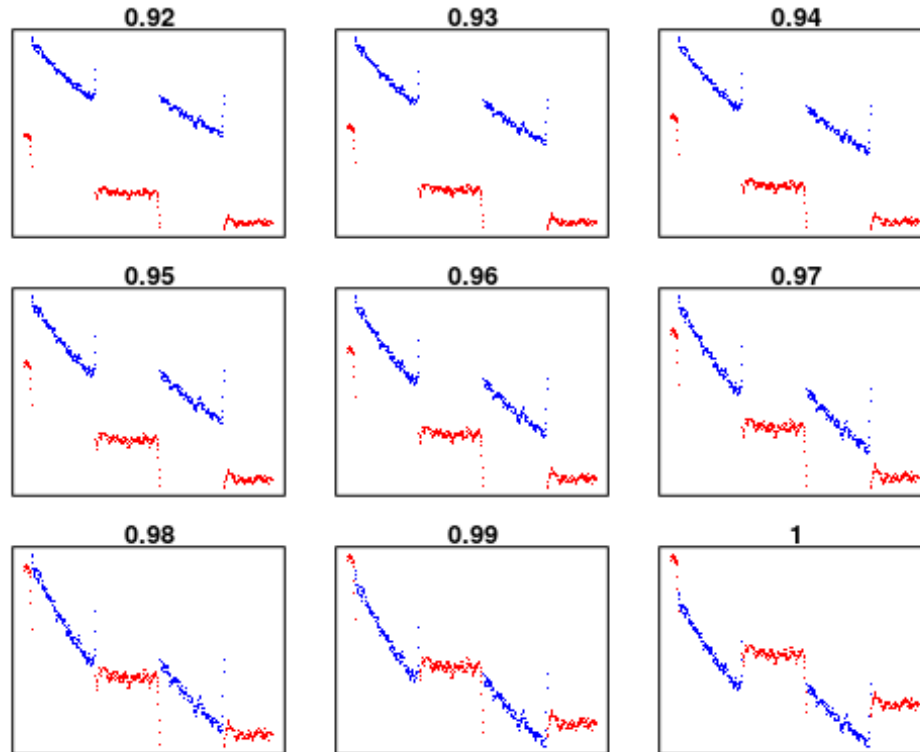
Correcting for light

```
par(mfrow=c(3,3))
par(mar=c(1,1,1,1))

for (i in seq(from=0.92,to=1,by=.01)){
  temp=octo11$O21adj
  temp[octo11$ambient1>20]=temp[octo11$ambient1>20]/i

  plot(temp[octo11$ambient1<20]~octo11$times[octo11$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))

  points(temp[octo11$ambient1>20]~octo11$times[octo11$ambient1>20],pch=".",
  col="blue")
  box()
}
```

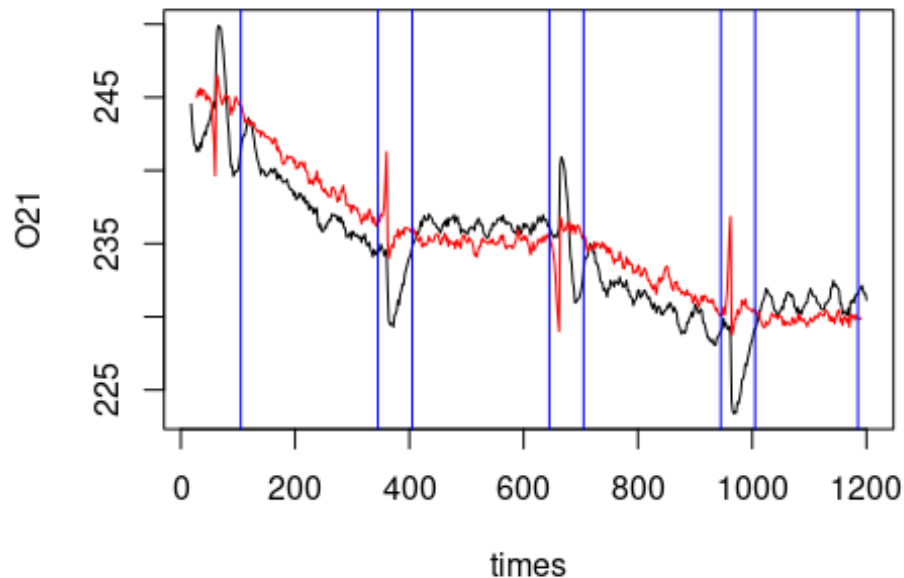
Choose the offset that best aligns the ends of the blue dots to the red dots.

Making an example figure

```
png("example02.png",width=1000,height=1000,res=200)
plot(temp[octo11$ambient1<20]~octo11$times[octo11$ambient1<20],pch=".",
col="red",cex=3,xlab="Time (s)",
ylab="")
mtext(expression("Oxygen Concentration ("*mu*"molO"[2]*"
L^-1*")"),side=2,line=2.5)
points(temp[octo11$ambient1>20]/0.98~octo11$times[octo11$ambient1>20],p
ch=".",col="blue",cex=3)
dev.off()
```

Applying light correction

```
light.off=0.98
octo11$O21adj2=octo11$O21adj
octo11$O21adj2[octo11$ambient1>20]=octo11$O21adj2[octo11$ambient1>20]/l
ight.off
plot(O21~times,data=octo11,type="l")
lines(O21adj2~times,data=octo11,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo11$times>60+trim.begin)),
          min(which(octo11$times>360-trim.end)),
          min(which(octo11$times>360+trim.begin)),
          min(which(octo11$times>660-trim.end)),
          min(which(octo11$times>660-trim.begin)),
          min(which(octo11$times>960-trim.end)),
          min(which(octo11$times>960-trim.begin)),
          min(which(octo11$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[11]=11
```

```
octo11.1=
```

```
(octo11$O21adj2[breaks[1]:breaks[2]]-
 octo11$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[11]=mean(octo11.1)*1000
```

```
resp$on1.o2[11]=mean(octo11$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[11]=mean(octo11$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo11.2=
```

```
(octo11$O21adj2[breaks[3]:breaks[4]]-
 octo11$O21adj2[breaks[3]:breaks[4]+1])*
```

```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[11]=mean(octo11.2)*1000
resp$off1.o2[11]=mean(octo11$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[11]=mean(octo11$tempv5_filt[breaks[3]:breaks[4]])

octo11.3=
(octo11$O21adj2[breaks[5]:breaks[6]]-
 octo11$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[11]=mean(octo11.3)*1000
resp$on2.o2[11]=mean(octo11$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[11]=mean(octo11$tempv5_filt[breaks[5]:breaks[6]])

octo11.4=
(octo11$O21adj2[breaks[7]:breaks[8]]-
 octo11$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[11]=mean(octo11.4)*1000
resp$off2.o2[11]=mean(octo11$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[11]=mean(octo11$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 12

Loading data

```

octo12=read.pyro('Octo15_08-17-2020.txt')
octo12.temp=read.csv('Octo15_08-17-2020_temp.txt')
octo12.temp$times=octo12.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo12=cbind(octo12[,1:28],matrix(nrow=nrow(octo12),ncol=26))
colnames(octo12)=c(colnames(octo12)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",
"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",
"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo12)){
    octo12[i,35+j]=mean(octo12.temp$temp[which.min(abs(octo12.temp$times-oc
to12$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo12[,48+i]=filter(octo12[,35+i],coeff)
}

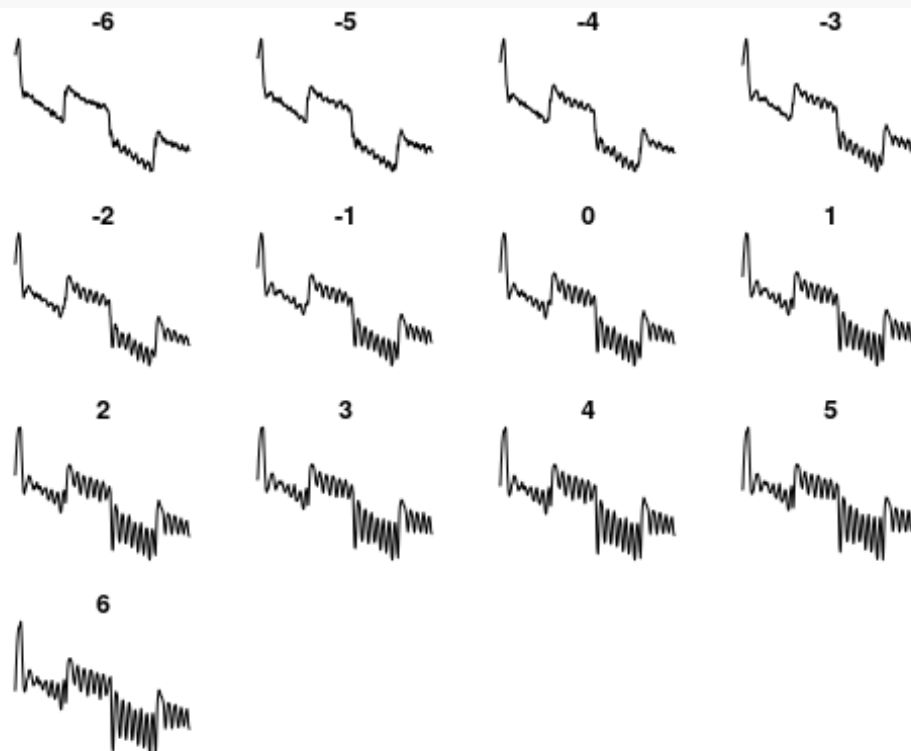
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo12$O21/((3.223674e-06*octo12[,48+i]^3.686095+8.338341e-01)^2)~
octo12$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-6
```

```
octo12$O21adj=octo12$O21/((3.223674e-06*octo12[,48+time.off]^3.686095+.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo12$O21adj
```

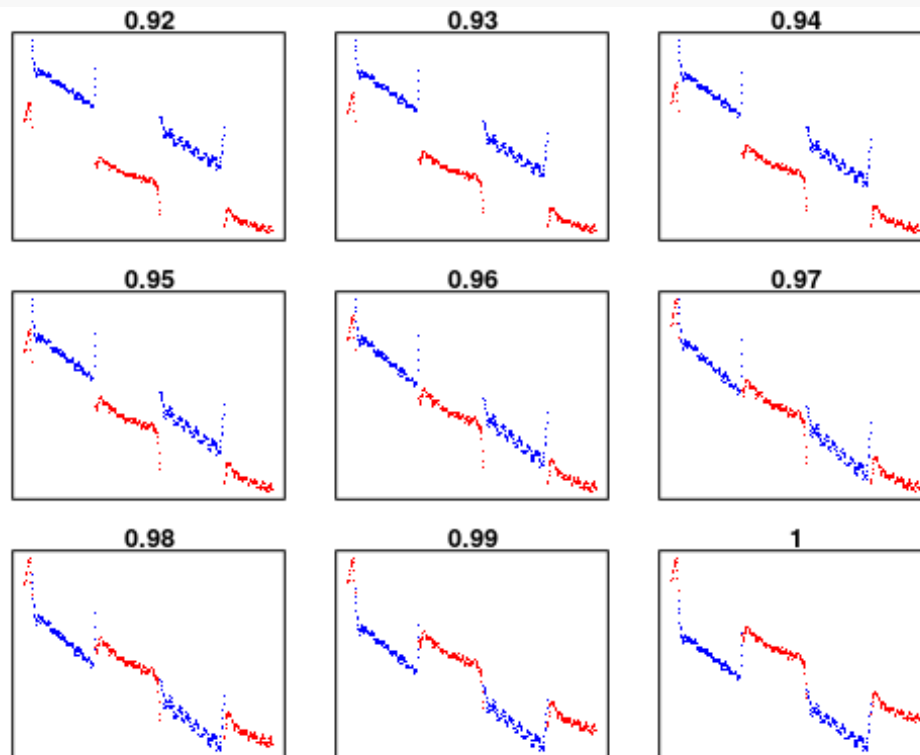
```
  temp[octo12$ambient1>20]=temp[octo12$ambient1>20]/i
```

```
  plot(temp[octo12$ambient1<20]~octo12$times[octo12$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo12$ambient1>20]~octo12$times[octo12$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



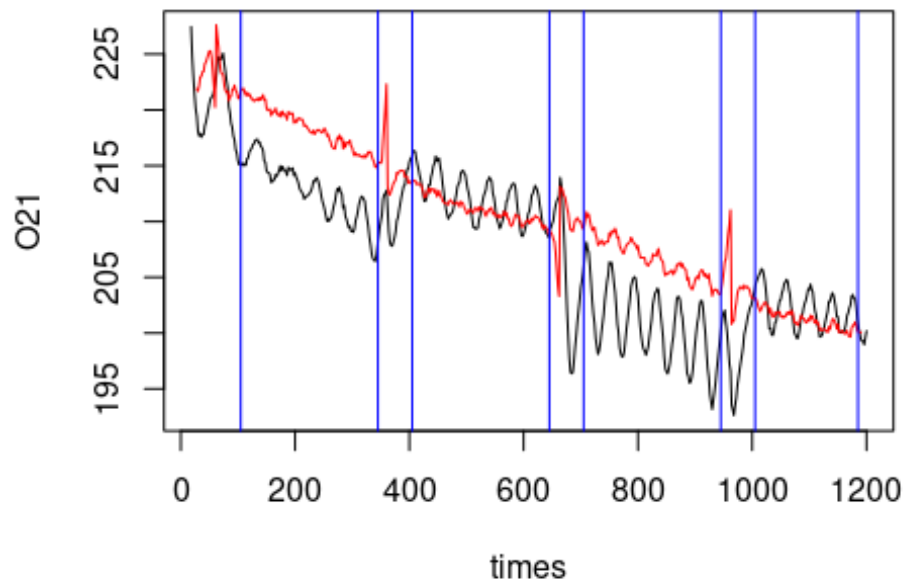
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.96
octo12$O21adj2=octo12$O21adj
octo12$O21adj2[octo12$ambient1>20]=octo12$O21adj2[octo12$ambient1>20]/light.off
plot(O21~times,data=octo12,type="l")
lines(O21adj2~times,data=octo12,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo12$times>60+trim.begin)),
min(which(octo12$times>360-trim.end)),
min(which(octo12$times>360+trim.begin)),
min(which(octo12$times>660-trim.end)),
min(which(octo12$times>660-trim.begin)),
min(which(octo12$times>960-trim.end)),
min(which(octo12$times>960-trim.begin)),
min(which(octo12$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[12]=12

```

```

octo12.1=
(octo12$O21adj2[breaks[1]:breaks[2]]-
octo12$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[12]=mean(octo12.1)*1000
resp$on1.o2[12]=mean(octo12$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[12]=mean(octo12$tempv5_filt[breaks[1]:breaks[2]])

octo12.2=
(octo12$O21adj2[breaks[3]:breaks[4]]-
 octo12$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[12]=mean(octo12.2)*1000
resp$off1.o2[12]=mean(octo12$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[12]=mean(octo12$tempv5_filt[breaks[3]:breaks[4]])

octo12.3=
(octo12$O21adj2[breaks[5]:breaks[6]]-
 octo12$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[12]=mean(octo12.3)*1000
resp$on2.o2[12]=mean(octo12$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[12]=mean(octo12$tempv5_filt[breaks[5]:breaks[6]])

octo12.4=
(octo12$O21adj2[breaks[7]:breaks[8]]-
 octo12$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[12]=mean(octo12.4)*1000
resp$off2.o2[12]=mean(octo12$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[12]=mean(octo12$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 13

Loading data

```

octo13=read.pyro('Octo16_08-17-2020.txt')
octo13.temp=read.csv('Octo16_08-17-2020_temp.txt')
octo13.temp$times=octo13.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo13=cbind(octo13[,1:28],matrix(nrow=nrow(octo13),ncol=26))
colnames(octo13)=c(colnames(octo13)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo13)){

octo13[i, 35+j]=mean(octo13.temp$temp[which.min(abs(octo13.temp$times-oc
to13$times[i]+j))])
  }
}

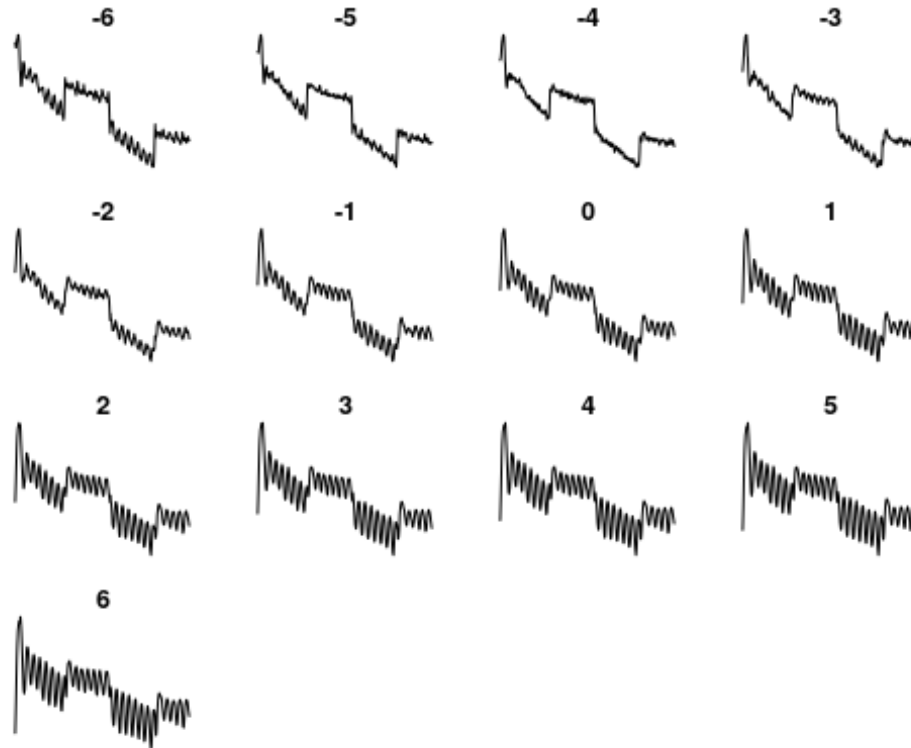
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo13[, 48+i]=filter(octo13[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo13$O21/((3.223674e-06*octo13[, 48+i]^3.686095+8.338341e-01)^2)~
octo13$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```

Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
```

```
octo13$O21adj=octo13$O21/((3.223674e-06*octo13[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo13$O21adj
```

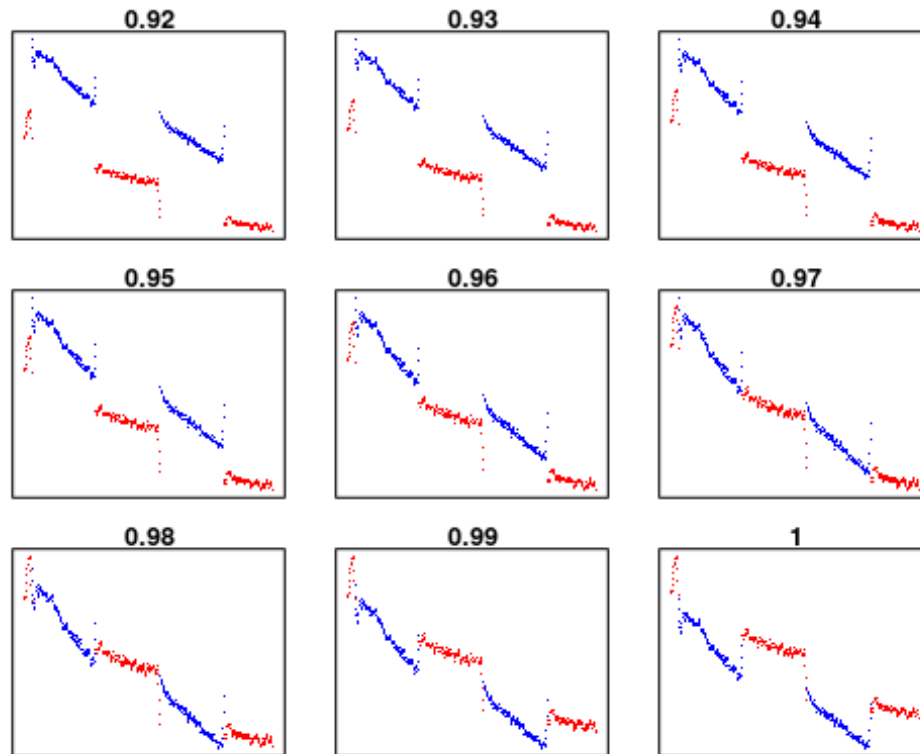
```
  temp[octo13$ambient1>20]=temp[octo13$ambient1>20]/i
```

```
  plot(temp[octo13$ambient1<20]~octo13$times[octo13$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo13$ambient1>20]~octo13$times[octo13$ambient1>20],pch=".",
  col="blue")
```

```
  box()
```

```
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.97
```

```
octo13$O21adj2=octo13$O21adj
```

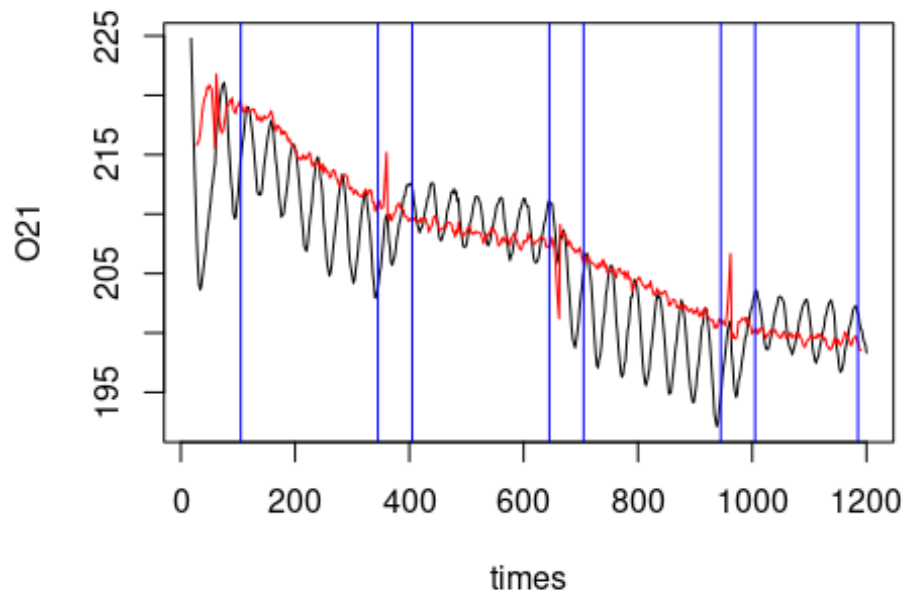
```
octo13$O21adj2[octo13$ambient1>20]=octo13$O21adj2[octo13$ambient1>20]/light.off
```

```
plot(O21~times,data=octo13,type="l")
```

```
lines(O21adj2~times,data=octo13,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo13$times>60+trim.begin)),
         min(which(octo13$times>360-trim.end)),
         min(which(octo13$times>360+trim.begin)),
         min(which(octo13$times>660-trim.end)),
         min(which(octo13$times>660-trim.begin)),
         min(which(octo13$times>960-trim.end)),
         min(which(octo13$times>960-trim.begin)),
         min(which(octo13$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[13]=13
```

```
octo13.1=
(octo13$O21adj2[breaks[1]:breaks[2]]-
 octo13$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[13]=mean(octo13.1)*1000
resp$on1.o2[13]=mean(octo13$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[13]=mean(octo13$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo13.2=
(octo13$O21adj2[breaks[3]:breaks[4]]-
 octo13$O21adj2[breaks[3]:breaks[4]+1])*
```

```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[13]=mean(octo13.2)*1000
resp$off1.o2[13]=mean(octo13$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[13]=mean(octo13$tempv5_filt[breaks[3]:breaks[4]])

octo13.3=
(octo13$O21adj2[breaks[5]:breaks[6]]-
 octo13$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[13]=mean(octo13.3)*1000
resp$on2.o2[13]=mean(octo13$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[13]=mean(octo13$tempv5_filt[breaks[5]:breaks[6]])

octo13.4=
(octo13$O21adj2[breaks[7]:breaks[8]]-
 octo13$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[13]=mean(octo13.4)*1000
resp$off2.o2[13]=mean(octo13$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[13]=mean(octo13$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 14

Loading data

```

octo14=read.pyro('Octo23_08-18-2020.txt')
octo14.temp=read.csv('Octo23_08-18-2020_temp.txt')
octo14.temp$times=octo14.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo14=cbind(octo14[,1:28],matrix(nrow=nrow(octo14),ncol=26))
colnames(octo14)=c(colnames(octo14)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",

"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",

"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo14)){
    octo14[i,35+j]=mean(octo14.temp$temp[which.min(abs(octo14.temp$times-oc
to14$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo14[,48+i]=filter(octo14[,35+i],coeff)
}

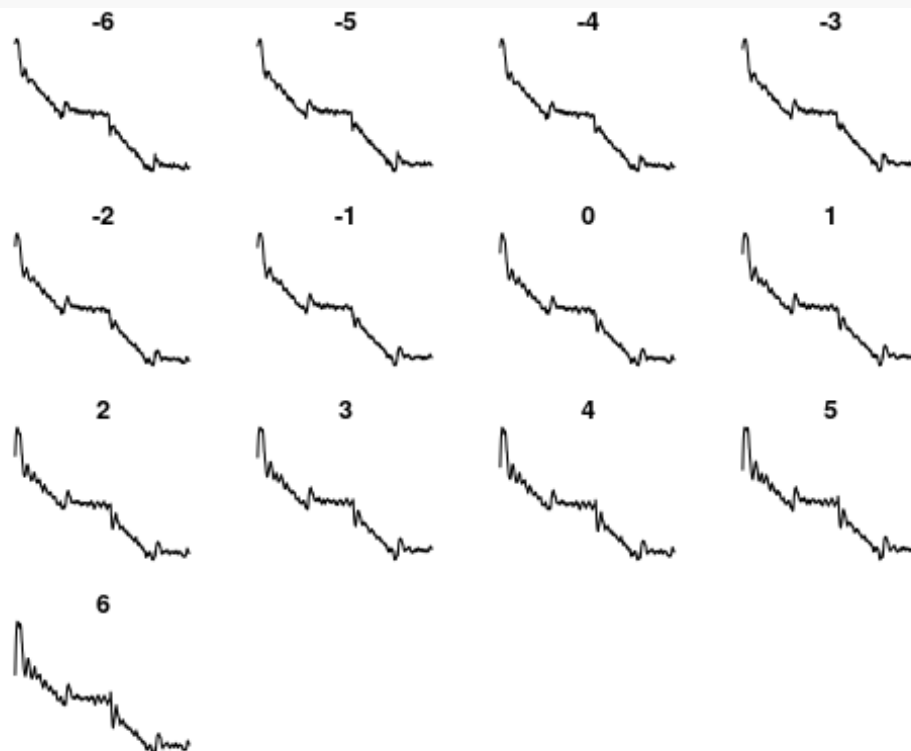
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo14$O21/((3.223674e-06*octo14[,48+i]^3.686095+8.338341e-01)^2)~
octo14$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the "temp.off" parameter below.

Correcting for temperature using correct time offset

```
time.off=-3
```

```
octo14$O21adj=octo14$O21/((3.223674e-06*octo14[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo14$O21adj
```

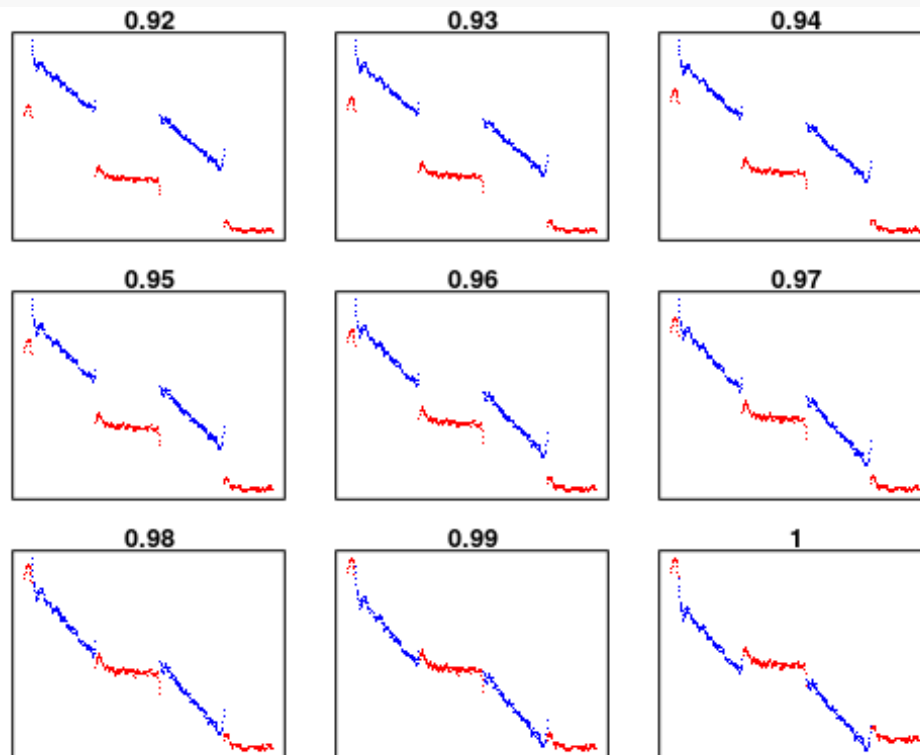
```
  temp[octo14$ambient1>20]=temp[octo14$ambient1>20]/i
```

```
  plot(temp[octo14$ambient1<20]~octo14$times[octo14$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo14$ambient1>20]~octo14$times[octo14$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



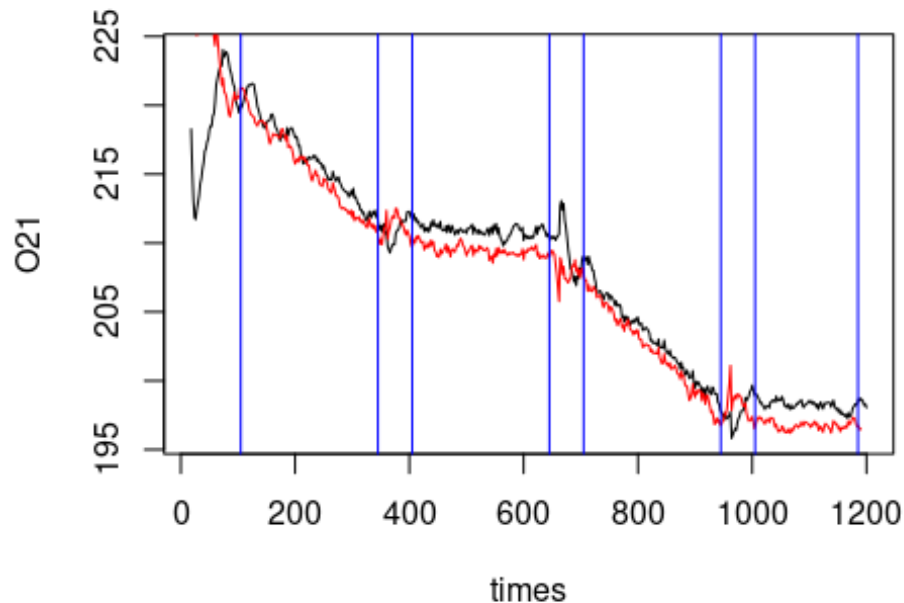
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.99
octo14$O21adj2=octo14$O21adj
octo14$O21adj2[octo14$ambient1>20]=octo14$O21adj2[octo14$ambient1>20]/light.off
plot(O21~times,data=octo14,type="l")
lines(O21adj2~times,data=octo14,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo14$times>60+trim.begin)),
min(which(octo14$times>360-trim.end)),
min(which(octo14$times>360+trim.begin)),
min(which(octo14$times>660-trim.end)),
min(which(octo14$times>660-trim.begin)),
min(which(octo14$times>960-trim.end)),
min(which(octo14$times>960-trim.begin)),
min(which(octo14$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[14]=14

```

```

octo14.1=
(octo14$O21adj2[breaks[1]:breaks[2]]-
octo14$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```

```

resp$on1[14]=mean(octo14.1)*1000
resp$on1.o2[14]=mean(octo14$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[14]=mean(octo14$tempv5_filt[breaks[1]:breaks[2]])

octo14.2=
(octo14$O21adj2[breaks[3]:breaks[4]]-
 octo14$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[14]=mean(octo14.2)*1000
resp$off1.o2[14]=mean(octo14$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[14]=mean(octo14$tempv5_filt[breaks[3]:breaks[4]])

octo14.3=
(octo14$O21adj2[breaks[5]:breaks[6]]-
 octo14$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[14]=mean(octo14.3)*1000
resp$on2.o2[14]=mean(octo14$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[14]=mean(octo14$tempv5_filt[breaks[5]:breaks[6]])

octo14.4=
(octo14$O21adj2[breaks[7]:breaks[8]]-
 octo14$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[14]=mean(octo14.4)*1000
resp$off2.o2[14]=mean(octo14$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[14]=mean(octo14$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 15

Loading data

```

octo15=read.pyro('Octo24_08-18-2020.txt')
octo15.temp=read.csv('Octo24_08-18-2020_temp.txt')
octo15.temp$times=octo15.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo15=cbind(octo15[,1:28],matrix(nrow=nrow(octo15),ncol=26))
colnames(octo15)=c(colnames(octo15)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```



```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo15)){

octo15[i, 35+j]=mean(octo15.temp$temp[which.min(abs(octo15.temp$times-oc
to15$times[i]+j))])
  }
}

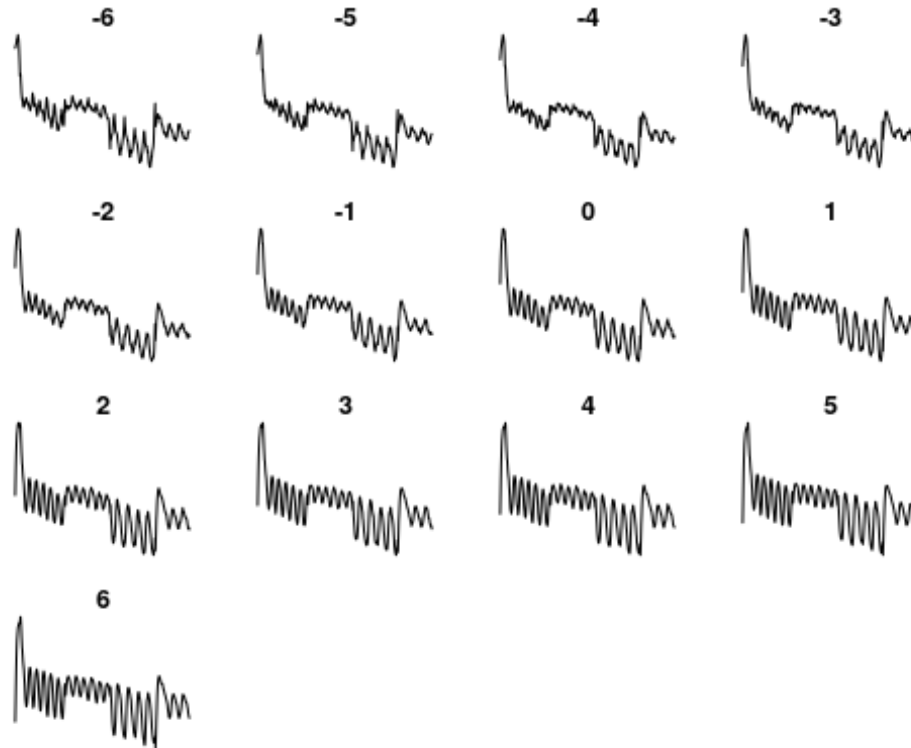
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo15[, 48+i]=filter(octo15[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo15$O21/((3.223674e-06*octo15[, 48+i]^3.686095+8.338341e-01)^2)~
octo15$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
octo15$O21adj=octo15$O21/((3.223674e-06*octo15[,48+time.off]^3.686095+8
.338341e-01)^2)
```

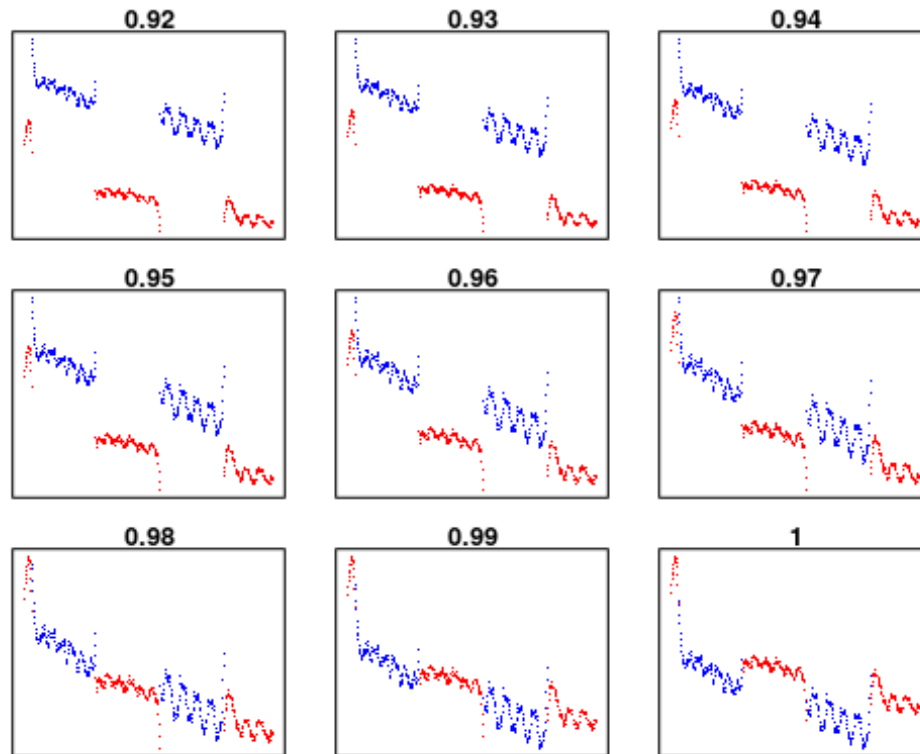
Correcting for light

```
par(mfrow=c(3,3))
par(mar=c(1,1,1,1))

for (i in seq(from=0.92,to=1,by=.01)){
  temp=octo15$O21adj
  temp[octo15$ambient1>20]=temp[octo15$ambient1>20]/i

  plot(temp[octo15$ambient1<20]~octo15$times[octo15$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))

  points(temp[octo15$ambient1>20]~octo15$times[octo15$ambient1>20],pch=".",
  col="blue")
  box()
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.98
```

```
octo15$O21adj2=octo15$O21adj
```

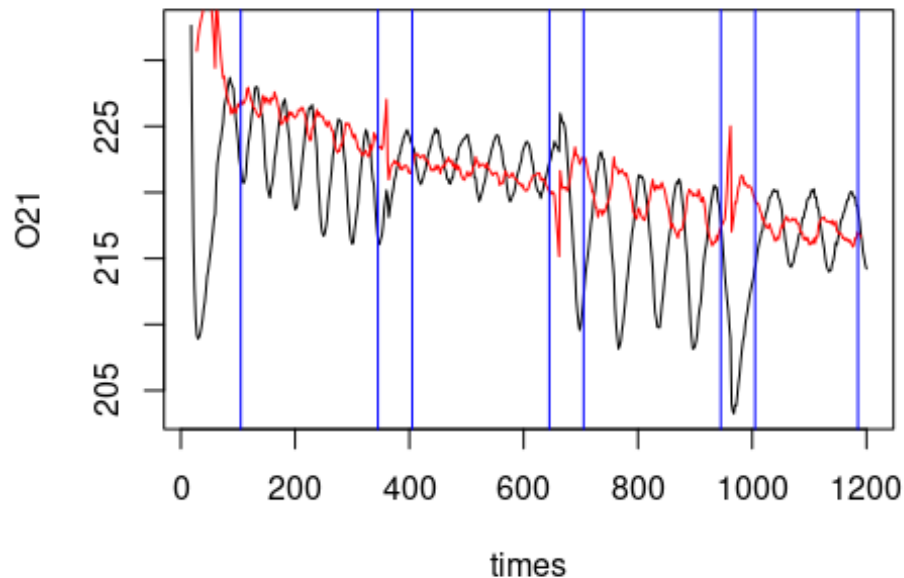
```
octo15$O21adj2[octo15$ambient1>20]=octo15$O21adj2[octo15$ambient1>20]/light.off
```

```
plot(O21~times,data=octo15,type="l")
```

```
lines(O21adj2~times,data=octo15,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo15$times>60+trim.begin)),
         min(which(octo15$times>360-trim.end)),
         min(which(octo15$times>360+trim.begin)),
         min(which(octo15$times>660-trim.end)),
         min(which(octo15$times>660-trim.begin)),
         min(which(octo15$times>960-trim.end)),
         min(which(octo15$times>960-trim.begin)),
         min(which(octo15$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[15]=15
```

```
octo15.1=
```

```
(octo15$O21adj2[breaks[1]:breaks[2]]-
 octo15$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[15]=mean(octo15.1)*1000
```

```
resp$on1.o2[15]=mean(octo15$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[15]=mean(octo15$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo15.2=
```

```
(octo15$O21adj2[breaks[3]:breaks[4]]-
 octo15$O21adj2[breaks[3]:breaks[4]+1])*
```

```

    (3600/2)*
    0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[15]=mean(octo15.2)*1000
resp$off1.o2[15]=mean(octo15$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[15]=mean(octo15$tempv5_filt[breaks[3]:breaks[4]])

octo15.3=
(octo15$O21adj2[breaks[5]:breaks[6]]-
 octo15$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[15]=mean(octo15.3)*1000
resp$on2.o2[15]=mean(octo15$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[15]=mean(octo15$tempv5_filt[breaks[5]:breaks[6]])

octo15.4=
(octo15$O21adj2[breaks[7]:breaks[8]]-
 octo15$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[15]=mean(octo15.4)*1000
resp$off2.o2[15]=mean(octo15$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[15]=mean(octo15$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 16

Loading data

```

octo16=read.pyro('Octo25_08-18-2020.txt')
octo16.temp=read.csv('Octo25_08-18-2020_temp.txt')
octo16.temp$time=octo16.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo16=cbind(octo16[,1:28],matrix(nrow=nrow(octo16),ncol=26))
colnames(octo16)=c(colnames(octo16)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",

"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

"tempv5_filt","tempv4_filt","tempv3_filt","tempv2_filt","tempv1_filt","
temp.0_filt",

"temp.1_filt","temp.2_filt","temp.3_filt","temp.4_filt","temp.5_filt","
temp.6_filt")

```

```

for (j in -6:6){
  for (i in 1:nrow(octo16)){
    octo16[i,35+j]=mean(octo16.temp$temp[which.min(abs(octo16.temp$times-oc
to16$times[i]+j))])
  }
}

```

```

smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

```

```

for (i in -6:6){
  octo16[,48+i]=filter(octo16[,35+i],coeff)
}

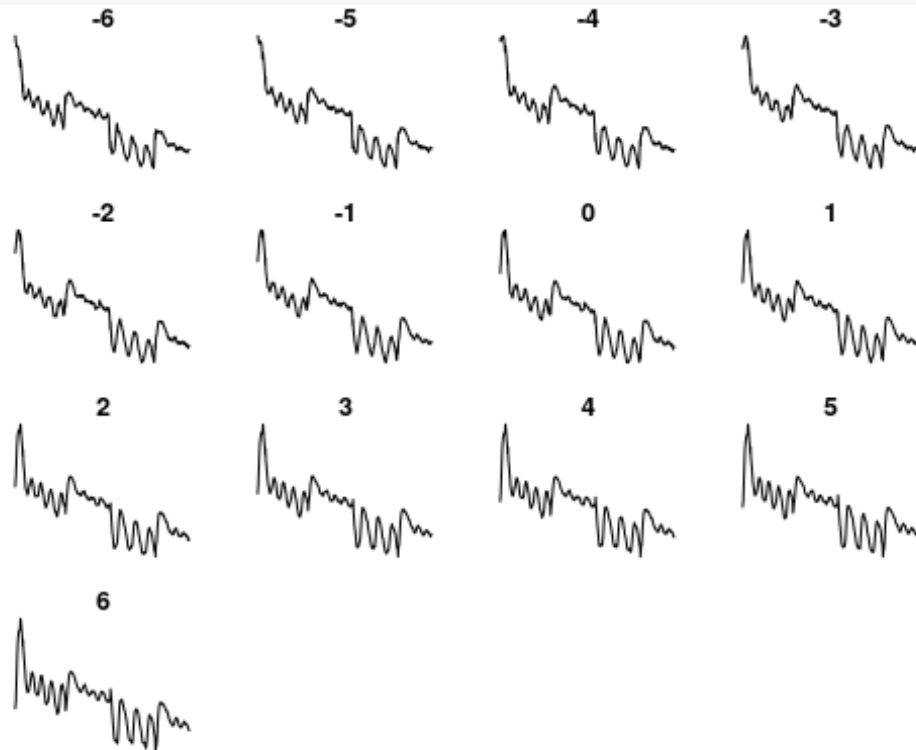
```

Choosing correct time offset

```

par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){
  plot(octo16$O21/((3.223674e-06*octo16[,48+i]^3.686095+8.338341e-01)^2)~
octo16$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-6
```

```
octo16$O21adj=octo16$O21/((3.223674e-06*octo16[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo16$O21adj
```

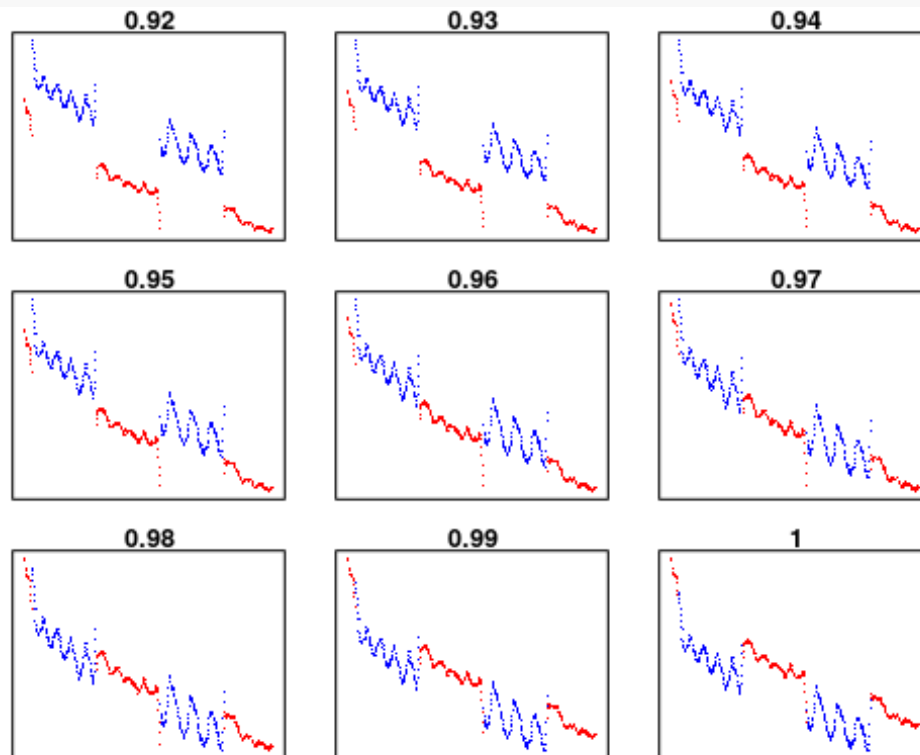
```
  temp[octo16$ambient1>20]=temp[octo16$ambient1>20]/i
```

```
  plot(temp[octo16$ambient1<20]~octo16$times[octo16$ambient1<20],pch=".",axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo16$ambient1>20]~octo16$times[octo16$ambient1>20],pch=".",col="blue")
```

```
  box()
```

```
}
```



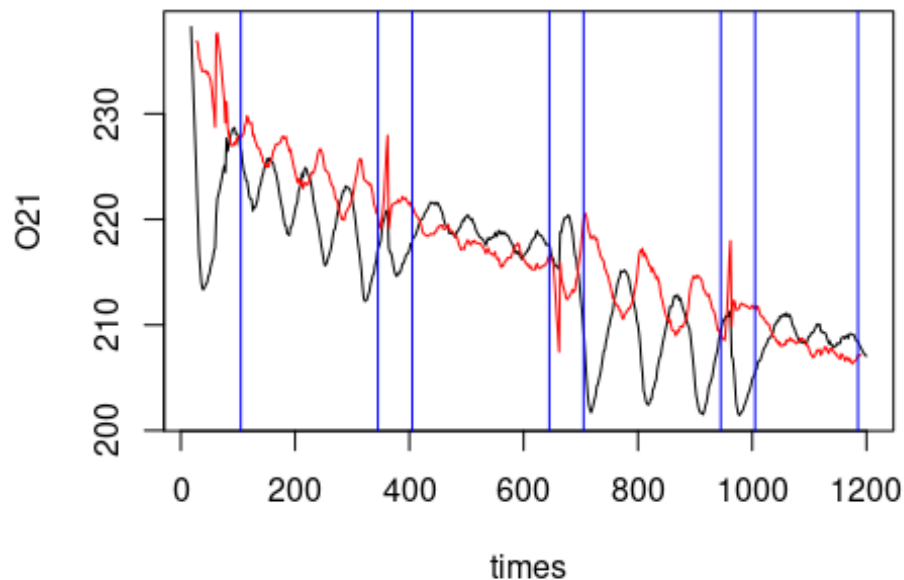
Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```

light.off=0.97
octo16$O21adj2=octo16$O21adj
octo16$O21adj2[octo16$ambient1>20]=octo16$O21adj2[octo16$ambient1>20]/light.off
plot(O21~times,data=octo16,type="l")
lines(O21adj2~times,data=octo16,col="red")
abline(v=c(60,360,660,960)+trim.begin,col="blue")
abline(v=c(360,660,960,1200)-trim.end,col="blue")

```



Defining the start and stop of respiration measurement

```

breaks=c(min(which(octo16$times>60+trim.begin)),
min(which(octo16$times>360-trim.end)),
min(which(octo16$times>360+trim.begin)),
min(which(octo16$times>660-trim.end)),
min(which(octo16$times>660-trim.begin)),
min(which(octo16$times>960-trim.end)),
min(which(octo16$times>960-trim.begin)),
min(which(octo16$times>1200-trim.end)))

```

Calculating Respiration rates and filling dataset

```

resp$octo[16]=16

```

```

octo16.1=
(octo16$O21adj2[breaks[1]:breaks[2]]-
octo16$O21adj2[breaks[1]:breaks[2]+1])*
(3600/2)*
0.000612 #volume of respirometer in liters (it is 612 microliters)

```



```

resp$on1[16]=mean(octo16.1)*1000
resp$on1.o2[16]=mean(octo16$O21adj2[breaks[1]:breaks[2]])
resp$on1.temp[16]=mean(octo16$tempv5_filt[breaks[1]:breaks[2]])

octo16.2=
(octo16$O21adj2[breaks[3]:breaks[4]]-
 octo16$O21adj2[breaks[3]:breaks[4]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[16]=mean(octo16.2)*1000
resp$off1.o2[16]=mean(octo16$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[16]=mean(octo16$tempv5_filt[breaks[3]:breaks[4]])

octo16.3=
(octo16$O21adj2[breaks[5]:breaks[6]]-
 octo16$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[16]=mean(octo16.3)*1000
resp$on2.o2[16]=mean(octo16$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[16]=mean(octo16$tempv5_filt[breaks[5]:breaks[6]])

octo16.4=
(octo16$O21adj2[breaks[7]:breaks[8]]-
 octo16$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[16]=mean(octo16.4)*1000
resp$off2.o2[16]=mean(octo16$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[16]=mean(octo16$tempv5_filt[breaks[7]:breaks[8]])

```

Octopus 17

Loading data

```

octo17=read.pyro('Octo26_08-18-2020.txt')
octo17.temp=read.csv('Octo26_08-18-2020_temp.txt')
octo17.temp$times=octo17.temp$millis/1000

```

Smoothing temperature at different time offsets

```

octo17=cbind(octo17[,1:28],matrix(nrow=nrow(octo17),ncol=26))
colnames(octo17)=c(colnames(octo17)[1:28],"tempv6","tempv5","tempv4","t
empv3","tempv2","tempv1",
"temp.0","temp.1","temp.2","temp.3","temp.4","temp.5","temp.6","tempv6_
filt",

```

```

"tempv5_filt", "tempv4_filt", "tempv3_filt", "tempv2_filt", "tempv1_filt", "
temp.0_filt",

"temp.1_filt", "temp.2_filt", "temp.3_filt", "temp.4_filt", "temp.5_filt", "
temp.6_filt")

for (j in -6:6){
  for (i in 1:nrow(octo17)){

octo17[i, 35+j]=mean(octo17.temp$temp[which.min(abs(octo17.temp$times-oc
to17$times[i]+j))])
  }
}

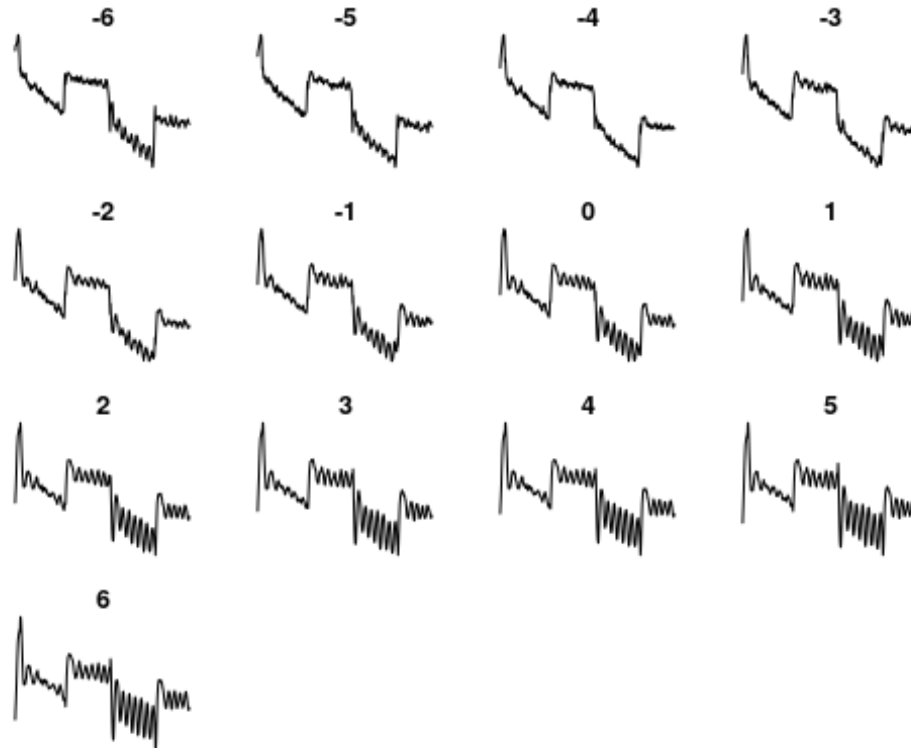
smooth=10
coeff=dnorm(seq(from=-3,to=3,length.out=smooth+1))/sum(dnorm(seq(from=-
3,to=3,length.out=smooth+1)))

for (i in -6:6){
  octo17[, 48+i]=filter(octo17[, 35+i],coeff)
}

Choosing correct time offset
par(mfrow=c(4,4))
par(mar=c(1,1,1,1))
for (i in -6:6){

plot(octo17$O21/((3.223674e-06*octo17[, 48+i]^3.686095+8.338341e-01)^2)~
octo17$times,type="l",ylab="",xlab="",main=i,axes=F)
}

```



Select the minimum offset that gives substantial improvement. Then put it into the “temp.off” parameter below.

Correcting for temperature using correct time offset

```
time.off=-4
```

```
octo17$O21adj=octo17$O21/((3.223674e-06*octo17[,48+time.off]^3.686095+8.338341e-01)^2)
```

Correcting for light

```
par(mfrow=c(3,3))
```

```
par(mar=c(1,1,1,1))
```

```
for (i in seq(from=0.92,to=1,by=.01)){
```

```
  temp=octo17$O21adj
```

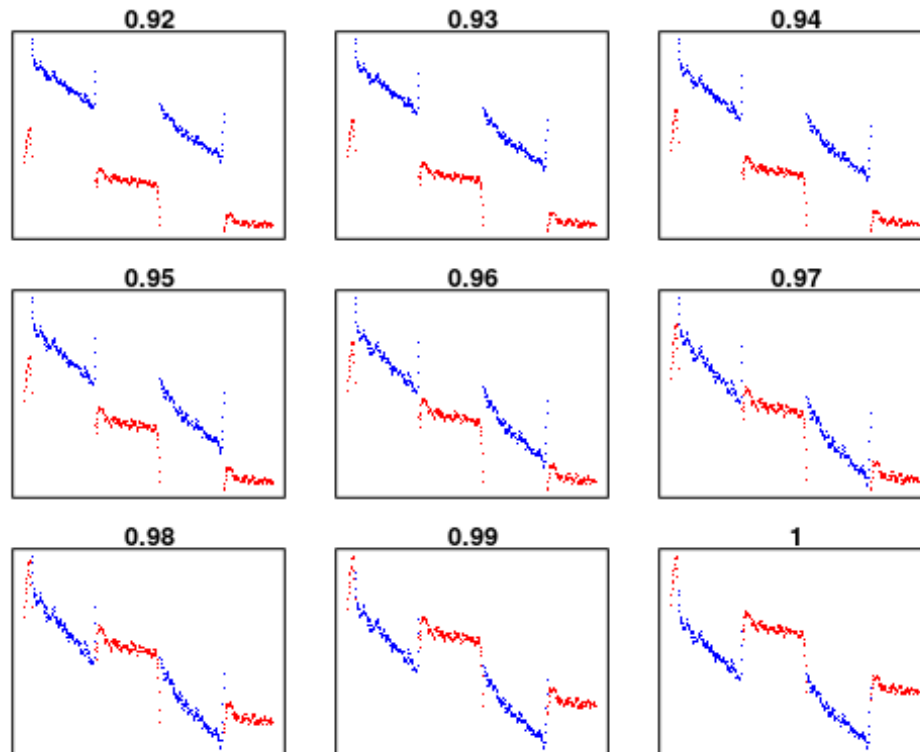
```
  temp[octo17$ambient1>20]=temp[octo17$ambient1>20]/i
```

```
  plot(temp[octo17$ambient1<20]~octo17$times[octo17$ambient1<20],pch=".",
  axes=F,main=i,col="red",ylim=range(temp,na.rm = T))
```

```
  points(temp[octo17$ambient1>20]~octo17$times[octo17$ambient1>20],pch=".",
  col="blue")
```

```
  box()
```

```
}
```



Choose the offset that best aligns the ends of the blue dots to the red dots.

Applying light correction

```
light.off=0.97
```

```
octo17$O21adj2=octo17$O21adj
```

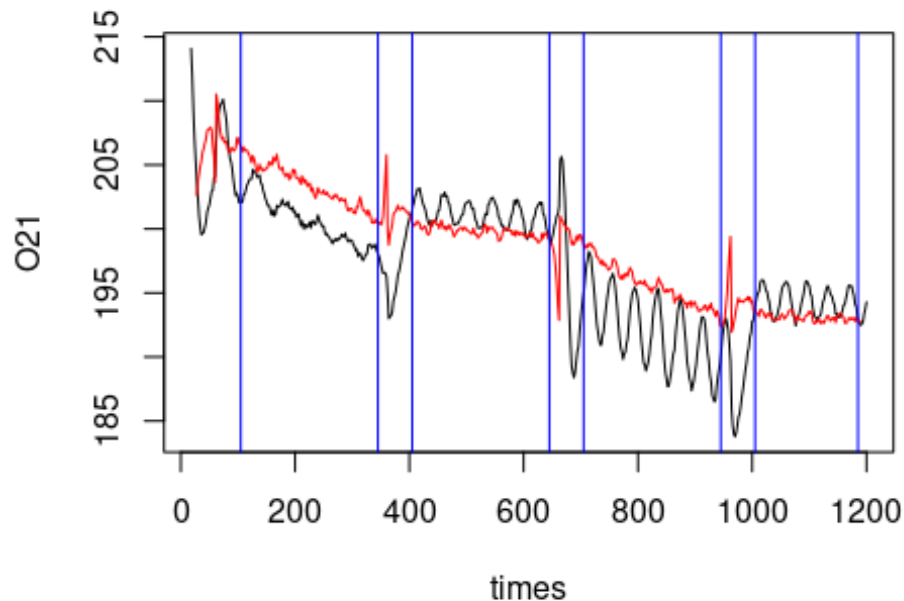
```
octo17$O21adj2[octo17$ambient1>20]=octo17$O21adj2[octo17$ambient1>20]/light.off
```

```
plot(O21~times,data=octo17,type="l")
```

```
lines(O21adj2~times,data=octo17,col="red")
```

```
abline(v=c(60,360,660,960)+trim.begin,col="blue")
```

```
abline(v=c(360,660,960,1200)-trim.end,col="blue")
```



Defining the start and stop of respiration measurement

```
breaks=c(min(which(octo17$times>60+trim.begin)),
         min(which(octo17$times>360-trim.end)),
         min(which(octo17$times>360+trim.begin)),
         min(which(octo17$times>660-trim.end)),
         min(which(octo17$times>660-trim.begin)),
         min(which(octo17$times>960-trim.end)),
         min(which(octo17$times>960-trim.begin)),
         min(which(octo17$times>1200-trim.end)))
```

Calculating Respiration rates and filling dataset

```
resp$octo[17]=17
```

```
octo17.1=
```

```
(octo17$O21adj2[breaks[1]:breaks[2]]-
 octo17$O21adj2[breaks[1]:breaks[2]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)
```

```
resp$on1[17]=mean(octo17.1)*1000
```

```
resp$on1.o2[17]=mean(octo17$O21adj2[breaks[1]:breaks[2]])
```

```
resp$on1.temp[17]=mean(octo17$tempv5_filt[breaks[1]:breaks[2]])
```

```
octo17.2=
```

```
(octo17$O21adj2[breaks[3]:breaks[4]]-
 octo17$O21adj2[breaks[3]:breaks[4]+1])*
```

```

      (3600/2)*
      0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off1[17]=mean(octo17.2)*1000
resp$off1.o2[17]=mean(octo17$O21adj2[breaks[3]:breaks[4]])
resp$off1.temp[17]=mean(octo17$tempv5_filt[breaks[3]:breaks[4]])

octo17.3=
(octo17$O21adj2[breaks[5]:breaks[6]]-
 octo17$O21adj2[breaks[5]:breaks[6]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$on2[17]=mean(octo17.3)*1000
resp$on2.o2[17]=mean(octo17$O21adj2[breaks[5]:breaks[6]])
resp$on2.temp[17]=mean(octo17$tempv5_filt[breaks[5]:breaks[6]])

octo17.4=
(octo17$O21adj2[breaks[7]:breaks[8]]-
 octo17$O21adj2[breaks[7]:breaks[8]+1])*
 (3600/2)*
 0.000612 #volume of respirometer in liters (it is 612 microliters)

resp$off2[17]=mean(octo17.4)*1000
resp$off2.o2[17]=mean(octo17$O21adj2[breaks[7]:breaks[8]])
resp$off2.temp[17]=mean(octo17$tempv5_filt[breaks[7]:breaks[8]])

Adding Octopus mass to dataset
resp$lognumber=c(1,2,4,5,6,7,9,10,11,13,14,15,16,23,24,25,26)
resp$mass=c(136,153,59,85,155,46,41,188,152,218,85,390,189,NA,155.5,26.
5,308)

write.csv(resp[,1:13],"skin_resp.csv")

```

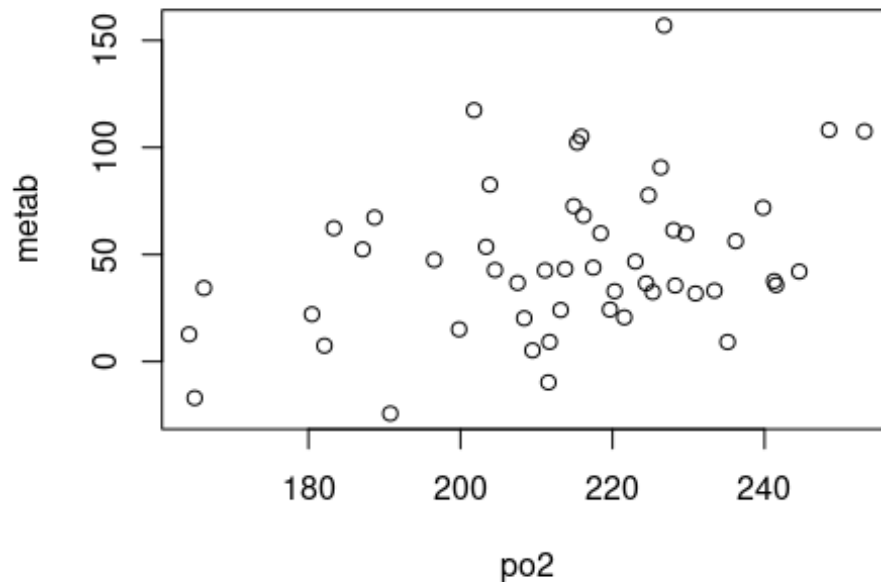
Finding the relationship between pO2 and measured metabolic rate

```

resp=resp[1:17,]
conform=data.frame(cbind(
  c(resp$on1, resp$off1, resp$on2),
  c(resp$on1.o2, resp$off1.o2, resp$on2.o2)
))
colnames(conform)=c("metab", "po2")

plot(metab~po2, data=conform)

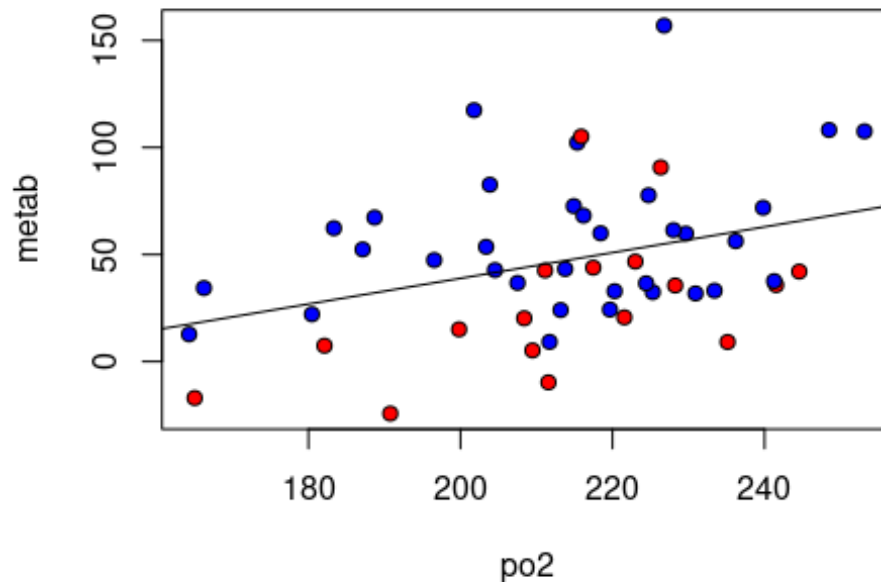
```



```
conform.lm=lm(metab~po2,data=conform)
summary(conform.lm)
```

```
##
## Call:
## lm(formula = metab ~ po2, data = conform)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -57.664 -23.752  -4.742  20.380 101.940
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -81.2646    48.5618  -1.673   0.1006
## po2           0.6008     0.2260   2.658   0.0106 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.67 on 49 degrees of freedom
## Multiple R-squared:  0.126, Adjusted R-squared:  0.1082
## F-statistic: 7.066 on 1 and 49 DF, p-value: 0.01058
```

```
plot(metab~po2,data=conform,pch=21,bg=c(rep("blue",17),rep("red",17)))
abline(conform.lm)
```



```

cor.metab=
  conform.lm$residuals+
  predict(conform.lm,newdata = data.frame(po2=240))

corrected=data.frame(cbind(cor.metab[1:17],cor.metab[18:34],cor.metab[3
5:51]))

colnames(corrected)=c("on1","off1","on2")

corrected$diff=((corrected$on1+corrected$on2)/2)-corrected$off1

resp$diff=((resp$on1+resp$on2)/2)-resp$off1

area=read.csv("area.csv")
colnames(area)=c("octo","blue","red","diff")

library(respirometry)

## Loading required package: PKNCA

bubbles=rep(0,17)
bubbles[16]=0.0213
bubbles[4]=0.0245
bubbles.vol=(4/3)*pi*(bubbles/2)^3

bubble.corrections=correct_bubble(0.000612,bubbles.vol,19,sal=30)/0.000

```


612

```
correct.b=corrected
correct.b$on1=correct.b$on1*bubble.corrections
correct.b$off1=correct.b$off1*bubble.corrections
correct.b$on2=correct.b$on2*bubble.corrections
correct.b$diff=((correct.b$on1+correct.b$on2)/2)-correct.b$off1

skin=data.frame(cbind(area,correct.b[c(1:13,16,17),]))
colnames(skin)[c(4,8)]=c("delta_area","delta_metab")

skin$per=skin$delta_metab/skin$delta_area
skin$temp_cor=skin$per*2^((11-19)/10) ##temperature correction to 11C
from 19C assuming a Q10 of 2
```

Hypothesis test to determine if chromatophore metabolic rate is greater than 0.

```
shapiro.test(skin$temp_cor[skin$delta_area>0])
```

```
##
## Shapiro-Wilk normality test
##
## data:  skin$temp_cor[skin$delta_area > 0]
## W = 0.69011, p-value = 0.0002924
```

The data is not normally distributed, so we compare using a non-parametric Wilcoxon signed-rank test to see if our metabolic rates are significantly different than 0

```
wilcox.test(skin$temp_cor,alternative="greater",mu=0)
```

```
##
## Wilcoxon signed rank exact test
##
## data:  skin$temp_cor
## V = 94, p-value = 0.02768
## alternative hypothesis: true location is greater than 0
```

with a p-value of ~0.03, I can reject the null hypothesis and accept the alternative hypothesis: that the true metabolic rate is greater than 0

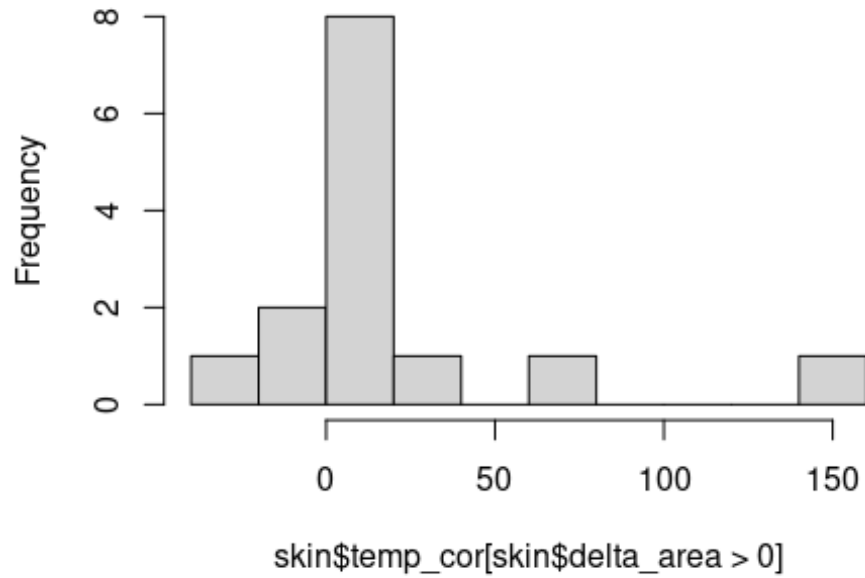
```
resamp=numeric()
for (i in 1:100000){
  resamp[i]=median(sample(skin$temp_cor[skin$delta_area>0],replace =
T))
}
```

```
quantile(resamp,probs=c(.025,.975))
```

```
##      2.5%      97.5%
## 3.452322 18.050318
```

```
hist(skin$temp_cor[skin$delta_area>0],breaks=10)
```

Histogram of skin\$temp_cor[skin\$delta_area > 0]



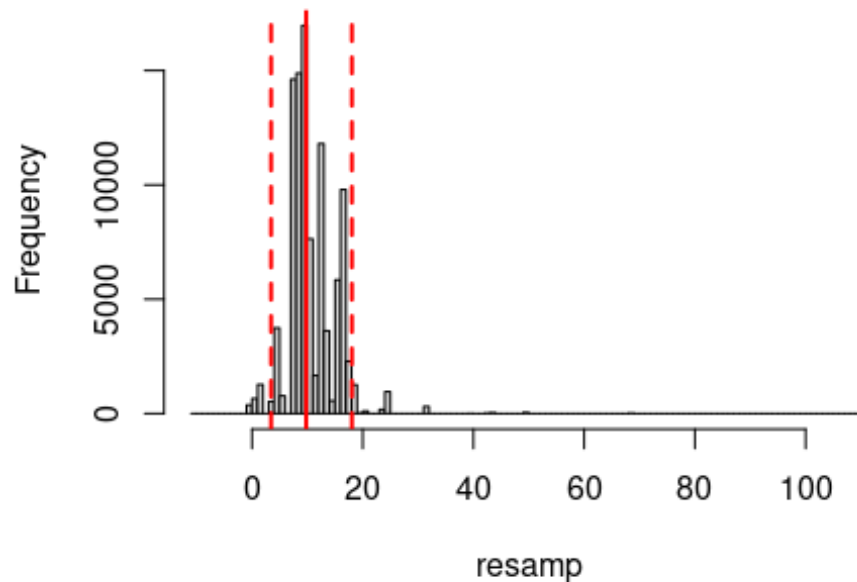
```
median(skin$temp_cor[skin$delta_area>0])
```

```
## [1] 9.78941
```

```
hist(resamp,breaks=100, main="")
```

```
abline(v=median(skin$temp_cor[skin$delta_area>0]),col="red", lwd=2)
```

```
abline(v=quantile(resamp,probs=c(.025,.975)),lty=2,col="red",lwd=2)
```



Estimating as percentage of total metabolic rate

```

all=read.csv('all_resp.csv')
all$total_resp=all$resp*all$mass
seibel=read.csv('seibel1997.csv')
seibel$total_resp=seibel$resp*seibel$mass
seibel$cor_resp=seibel$total_resp*2^((11-c(5,5,10,10,10))/10)
reg=data.frame(mass=c(all$mass,seibel$mass),total_resp=c(all$total_resp
,seibel$cor_resp))

resp.lm=lm(log(reg$total_resp)~log(reg$mass))
scaling.power=resp.lm$coefficients[2]

lm(log(all$total_resp)~log(all$mass))

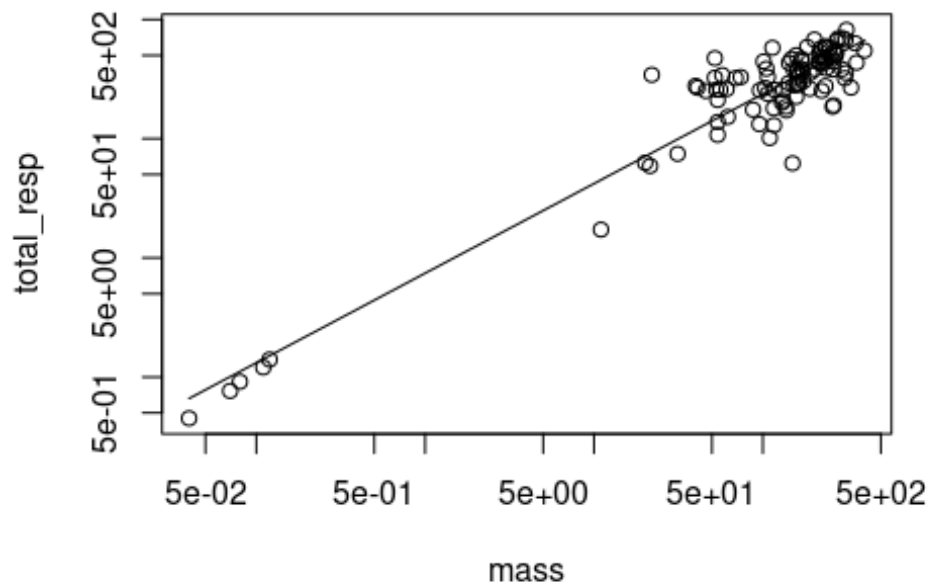
##
## Call:
## lm(formula = log(all$total_resp) ~ log(all$mass))
##
## Coefficients:
## (Intercept)  log(all$mass)
##      2.8745      0.5794

plot(total_resp~mass,data=reg,log="xy")
resp.nls=nls(total_resp~M*mass^scaling.power, data=reg,start=list(M=2))
resp.nls

```

```
## Nonlinear regression model
## model: total_resp ~ M * mass^scaling.power
## data: reg
## M
## 7.433
## residual sum-of-squares: 1767194
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 5.56e-10

lines(predict(resp.nls,newdata=data.frame(mass=seq(from=0.04,to=400,length.out=100)))~seq(from=0.04,to=400,length.out=100))
```



```
octoarea=data.frame(mass=c(33,377), area=c(14826,96929))

area.nls=nls(area~M*mass^t,
data=octoarea,start=list(M=1000,t=1),algorithm="port")
area.nls

## Nonlinear regression model
## model: area ~ M * mass^t
## data: octoarea
## M t
## 1001.0886 0.7709
## residual sum-of-squares: 6.191e-19
##
## Algorithm "port", convergence message: X-convergence (3)
```

```

png("Area_by_mass.png",width=6,height=6,units="in",res=300,pointsize=12
)
plot(area~mass,data=octoarea,ylim=c(0,100000),xlim=c(0,400),axes=F,ylab
="",xlab="",
      type="n")
mtext(expression("Area (10,000 mm2*)"),side=2,line=2.5)
mtext("Mass (g)",side=1,line=2.5)
box()
axis(1)
axis(2,at=seq(from=0,to=100000,by=20000),labels = c(0,2,4,6,8,10))
lines(predict(area.nls,newdata=data.frame(mass=seq(from=0,to=400,by=1))
)~seq(from=0,to=400,by=1))
points(area~mass,data=octoarea,pch=21,bg="red",cex=1.3)
dev.off()

## png
## 2

mass.pred=200
((predict(area.nls,newdata =
data.frame(mass=mass.pred))/2)*quantile(resamp,probs=c(.025,.5,.975))/1
000)/mass.pred

##      2.5%      50%      97.5%
## 0.5131926 1.4552097 2.6832055

predict(resp.nls,newdata = data.frame(mass=mass.pred))/mass.pred

## log(reg$mass)
##      1.994925

Individual Estimates
octo.est=(predict(area.nls,newdata =
data.frame(mass=skin$octomass))/2)*skin$temp_cor/1000

Plotting this kinda insane data
#masses=seq(from=10,to=400,by=1)
masses=exp(1)^seq(from=log(25),to=log(400),length.out = 500)
skin.est=data.frame(lo=numeric(),hi=numeric(),mean=numeric())

for (i in 1:length(masses)){
  skin.est[i,1:2]=(predict(area.nls,newdata =
data.frame(mass=masses[i]))/2)*quantile(resamp,probs=c(.025,.975))/1000
  #skin.est[i,3]=(predict(area.nls,newdata =
data.frame(mass=masses[i]))/2)*mean(resamp)/1000
  skin.est[i,3]=(predict(area.nls,newdata =
data.frame(mass=masses[i]))/2)*median(skin$temp_cor[skin$delta_area>0]
/1000
)
}

svg(file="Chromatophore_MR.svg",height=7,width=11,pointsize=18)
plot(total_resp~mass,data=all,ylim=c(20,4000),xlim=c(20,1000),type="n",

```

```

log="xy",ylab="",xlab="",axes=F)

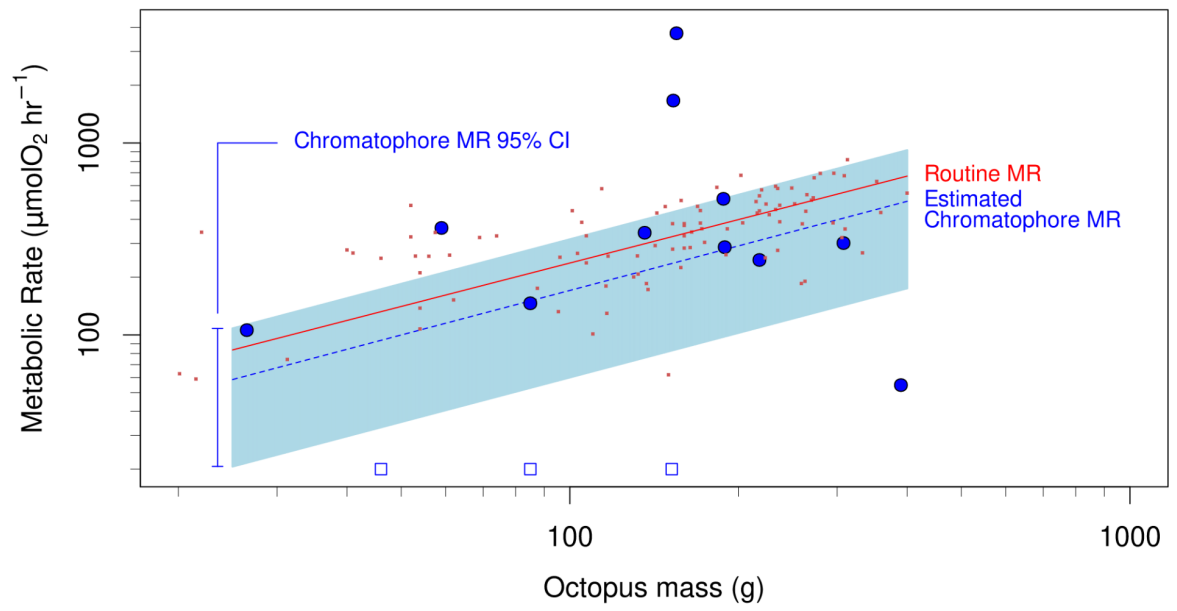
for (i in 1:length(masses)){

lines(c(masses[i],masses[i]),c(skin.est[i,1],skin.est[i,2]),col="lightblue",lwd=2)
}

lines(masses,skin.est$mean,lty=2,col="blue")
points(octo.est[skin$delta_area>0]~skin$octomass[skin$delta_area>0],pch=21,bg="blue")
points(total_resp~mass,data=all,pch=".",col="indianred3",cex=3)
points(rep(20,3)~skin$octomass[skin$delta_area>0&skin$temp_cor<0],pch=2, col="blue")

lines(predict(resp.nls,newdata=data.frame(mass=masses))~masses,col="red")
box()
axis(1,at=c(100,1000))
rug(seq(from=20,to=90,by=10),side=1,ticks=-0.02)
rug(seq(from=200,to=900,by=100),side=1,ticks=-0.02)
axis(2,at=c(10,100,1000))
rug(seq(from=20,to=90,by=10),side=2,ticks=-0.02)
rug(seq(from=200,to=900,by=100),side=2,ticks=-0.02)
rug(seq(from=2000,to=4000,by=1000),side=2,ticks=-0.02)
mtext(expression("Metabolic Rate ("*mu*"molO"[2]*"hr"^-1*")"),side=2,cex=1,line=2.5)
mtext("Octopus mass (g)",side=1,cex=1,line=2.5)
text(400,predict(resp.nls,newdata=list(mass=400)),"Routine MR",cex=0.8,pos=4,col="red")
#text(400,predict(resp.nls,newdata=list(mass=400))-100,"(resting octopus in dark with",cex=0.6,pos=4,col="red")
#text(400,predict(resp.nls,newdata=list(mass=400))-200,"little or no chromatophore usage)",cex=0.6,pos=4,col="red")
text(400,max(skin.est$mean),"Estimated",pos=4,cex=0.8,col="blue")
text(400,max(skin.est$mean)-110,"Chromatophore MR",pos=4,cex=0.8,col="blue")
lines(c(23,24),rep(min(skin.est$lo),2),col="blue")
lines(c(23,24),rep(min(skin.est$hi),2),col="blue")
lines(c(23.5,23.5),c(min(skin.est$lo),min(skin.est$hi)),col="blue")
lines(c(23.5,23.5,30),c(130,1000,1000),col="blue")
text(30,1000,"Chromatophore MR 95% CI",cex=0.8,pos=4,col="blue")
dev.off()

```



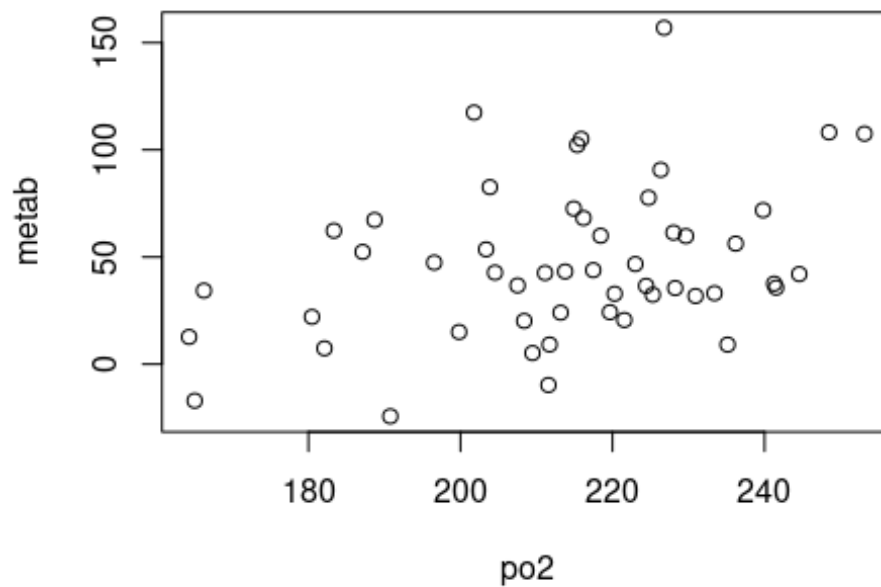
Chromatophore metabolic rate estimated per octopus

Skin Analysis

```
resp=read.csv("skin_resp.csv")
```

Finding the relationship between pO2 and measured metabolic rate

```
resp=resp[1:17,]  
conform=data.frame(cbind(  
  c(resp$on1,resp$off1,resp$on2),  
  c(resp$on1.o2,resp$off1.o2,resp$on2.o2)  
))  
colnames(conform)=c("metab","po2")  
plot(metab~po2,data=conform)
```



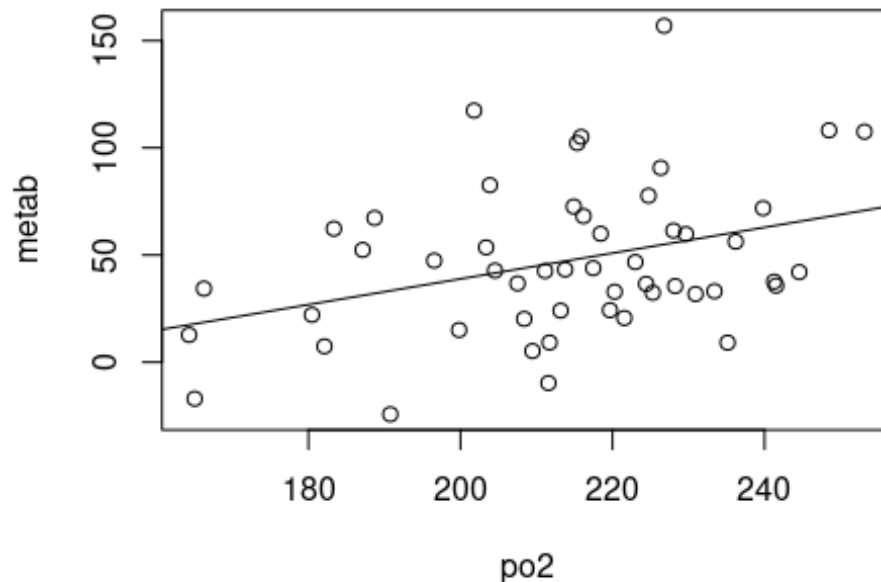
```
conform.lm=lm(metab~po2,data=conform)  
summary(conform.lm)
```

```
##  
## Call:  
## lm(formula = metab ~ po2, data = conform)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -57.664 -23.752  -4.742   20.380  101.940   
##  
## Coefficients:
```



```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -81.2646   48.5618  -1.673   0.1006
## po2          0.6008    0.2260   2.658   0.0106 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.67 on 49 degrees of freedom
## Multiple R-squared:  0.126, Adjusted R-squared:  0.1082
## F-statistic: 7.066 on 1 and 49 DF,  p-value: 0.01058
```

```
plot(metab~po2,data=conform)
abline(conform.lm)
```



```
cor.metab=
  conform.lm$residuals+
  predict(conform.lm,newdata = data.frame(po2=240))
```

```
corrected=data.frame(cbind(cor.metab[1:17],cor.metab[18:34],cor.metab[35:51]))
```

```
colnames(corrected)=c("on1","off1","on2")
```

```
corrected$diff=((corrected$on1+corrected$on2)/2)-corrected$off1
```

```
resp$diff=((resp$on1+resp$on2)/2)-resp$off1
```

```

area=read.csv("area.csv")
colnames(area)=c("octo", "blue", "red", "diff")

library(respirometry)

## Loading required package: PKNCA

bubbles=rep(0,17)
bubbles[16]=0.0213
bubbles[4]=0.0245
bubbles.vol=(4/3)*pi*(bubbles/2)^3

bubble.corrections=correct_bubble(0.000612,bubbles.vol,19,sal=30)/0.000
612

correct.b=corrected
correct.b$on1=correct.b$on1*bubble.corrections
correct.b$off1=correct.b$off1*bubble.corrections
correct.b$on2=correct.b$on2*bubble.corrections
correct.b$diff=((correct.b$on1+correct.b$on2)/2)-correct.b$off1

skin=data.frame(cbind(area,correct.b[c(1:13,16,17),]))
colnames(skin)[c(4,8)]=c("delta_area","delta_metab")

skin$per=skin$delta_metab/skin$delta_area
t.test(skin$per[skin$delta_area>0],alternative="greater",mu=0)

##
## One Sample t-test
##
## data:  skin$per[skin$delta_area > 0]
## t = 1.9785, df = 13, p-value = 0.03473
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  4.112207      Inf
## sample estimates:
## mean of x
##  39.1992

mean(skin$per[skin$delta_area>0])

## [1] 39.1992

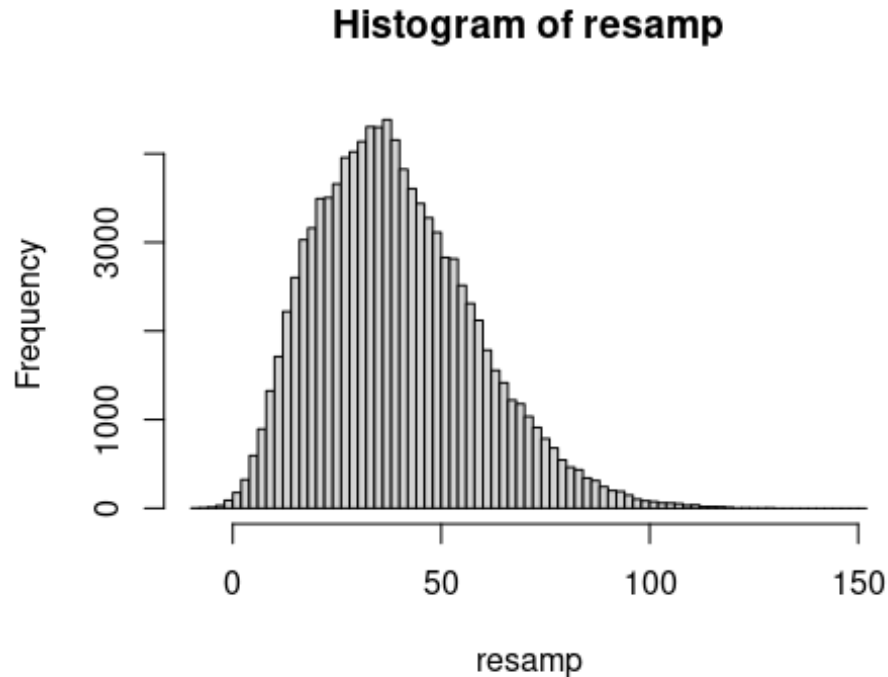
resamp=numeric()
for (i in 1:100000){
  resamp[i]=mean(sample(skin$per[skin$delta_area>0],replace = T))
}

quantile(resamp,probs=c(.025,.975))

```

```
##      2.5%      97.5%
## 8.615838 81.853898

hist(resamp,breaks=100)
```



```
Metabolic rate per mm by function of octopus size
skin$octomass=c(136,153,59,85,155,46,41,188,152,218,85,390,189,NA,155.5
,26.5,308)[as.numeric(rownames(skin))]
```

```
plot(per~octomass,data=skin[skin$delta_area>0,],xlab="Mass
(g)",ylab=expression("Metabolic Rate per
mm"^2*""),mgp=c(2.75,1,0),pch=21,bg="black")
axis(1,at=seq(from=50,to=350,by=50))
axis(2,at=seq(from=-50,to=300,by=100))
mass.lm=lm(per~octomass,data=skin[skin$delta_area>0,])
summary(mass.lm)
```

```
##
## Call:
## lm(formula = per ~ octomass, data = skin[skin$delta_area > 0,
##    ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -80.298 -28.834 -21.312  -8.348  226.568
##
## Coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 43.96207   39.01017   1.127   0.282
## octomass   -0.03044    0.21171  -0.144   0.888
##
## Residual standard error: 77.09 on 12 degrees of freedom
## Multiple R-squared:  0.00172,    Adjusted R-squared:  -0.08147
## F-statistic: 0.02067 on 1 and 12 DF,  p-value: 0.8881
```

```
abline(mass.lm)
```

