

Package ‘respirometry’

October 8, 2016

Type Package

Title Tools for Conducting Respirometry Experiments

Version 0.2.0

Date 2016-10-07

Author Matthew A. Birk

Maintainer Matthew A. Birk <matthewabirk@gmail.com>

Description Provides tools to enable the researcher to more precisely conduct respirometry experiments. Strong emphasis is on aquatic respirometry. Tools focus on helping the researcher setup and conduct experiments. Analysis of the resulting data is not a focus since analyses are often specific to a particular setup, and thus are better created by the researcher individually. This package provides tools for intermittent, flow-through, and closed respirometry techniques.

Imports birk, graphics, lubridate, marelac, measurements, methods, seacarb, stats

License GPL-3

LazyData TRUE

Encoding UTF-8

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-08 08:48:13

R topics documented:

closed	2
co2_add	3
co2_flush	4
co2_rate	6
conv_NH4	7
conv_o2	8

correct_bubble	9
flush_carb	10
flush_o2	11
flush_water	12
goal_flush_pH	13
guess_TA	14
guess_when	16
import_presens	17
import_witrox	19
max_MO2	20
mean_pH	22
min_flow	23
peri_pump	24
predict_NH4	25
predict_pH	26
Q10	27
respirometry	28
RQ	28
scale_MO2	30

Index	32
--------------	-----------

closed	<i>Closed respirometry</i>
--------	----------------------------

Description

Returns the unknown parameter given 3 of 4 parameters to calculate respiration rate in a closed respirometer. This is useful both for basic closed respirometry setups, and also for the closed measurement phase of intermittent respirometry.

Usage

```
closed(MO2, delta_pO2, duration, vol, temp = 25, sal = 35,
      atm_pres = 1013.25)
```

Arguments

MO2	whole-animal oxygen consumption rate (umol / hour).
delta_pO2	desired change in pO2 (% air saturation).
duration	desired duration to reach delta_pO2 (minutes).
vol	volume of the respirometer (liter).
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[flush_water](#)

Examples

```
# I've read in the literature that my animal has an SMR of 200 umol/h. How large of a
# respirometer do I want if I want it to breathe down to 80% air saturation in 30 minutes?
closed(MO2 = 200, delta_pO2 = 100 - 80, duration = 30) # returns respirometer volume

# I've read in the literature that my animal has an SMR of 1000 umol/h. How long will it take to
# breathe down a 50 L respirometer by 10% air saturation?
closed(MO2 = 1000, delta_pO2 = 10, vol = 50) # returns the duration to breathe down the O2

# How does animal size affect how long my measurement periods last?
mass_range = seq(100, 400, 50)
dur_range = (closed(MO2 = scale_MO2(mass_1 = 100, MO2_1 = 400, mass_2 = mass_range)$MO2_2,
  delta_pO2 = 20, vol = 10))
plot(mass_range, dur_range, type = 'b')

# What is the MO2 if O2 drops 0.44 mg/l in 33 minutes when the respirometer volume is 30 L?
closed(delta_pO2 = conv_o2(o2 = 0.44, from = 'mg_per_l', to = 'percent_a.s.'), duration = 33,
  vol = 30)
```

co2_add

Calculate CO2 to add to water

Description

Calculates the moles of CO2 gas to be added to a volume of seawater to achieve the desired pCO2. Useful for ocean acidification experiments where CO2 treatments are desired.

Usage

```
co2_add(goal_pco2, start_pH, vol, temp = 25, sal = 35, TA = NULL,
  atm_pres = 1013.25)
```

Arguments

goal_pco2	the desired pCO2 in the water (uatm).
start_pH	pH of the water before CO2 is added (total scale).
vol	volume of the water (liter).
temp	temperature (°C). Default is 25 °C.

sal	salinity (psu). Default is 35 psu. If sal < 26 psu, then TA must be provided.
TA	(optional) total alkalinity (umol / kg). If undefined TA is estimated from salinity using guess_TA .
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

moles of CO₂ gas to be added to the seawater.

Note

It is assumed that all of the CO₂ added dissolves and remains in solution. This can be achieved (almost completely) by bubbling CO₂ according to Jokiel et al. 2014.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Jokiel PL, Bahr KD, Rodgers KS. 2014. Low-cost, high-flow mesocosm system for simulating ocean acidification with CO₂ gas. *Limnol Oceanogr Methods*. 12:313–322.

See Also

[co2_rate](#), [flush_carb](#), [carb](#), [peri_pump](#)

Examples

```
# I want the 50 L reservoir to have a pCO2 = 1000 uatm. It currently has a pH of 7.88.
# How many moles of CO2 gas should be added to the water to reach my desired pCO2?
co2_add(goal_pco2 = 1000, start_pH = 7.88, vol = 50)
```

co2_flush

Calculate CO₂ to add to flush reservoir

Description

Calculates the moles of CO₂ gas to be added to a seawater reservoir before flushing a respirometer to achieve the desired pCO₂ in the respirometer after the flush. Useful for ocean acidification experiments where CO₂ treatments are desired.

Usage

```
co2_flush(goal_pco2, resp_pH, resp_vol, flush_pH, flush_vol, flush_remain = 0,
temp = 25, sal = 35, TA = NULL, atm_pres = 1013.25)
```

Arguments

goal_pco2	the desired pCO ₂ in the respirometer after the flush (uatm).
resp_pH	pH inside the respirometer before the flush (total scale).
resp_vol	volume of the respirometer (liter).
flush_pH	pH of the reservoir water used for flushing before CO ₂ is added (total scale).
flush_vol	volume of the flush reservoir (liter).
flush_remain	volume of the flush reservoir that will remain after the flush (liter).
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu. If sal < 26 psu, then TA must be provided.
TA	(optional) total alkalinity (umol / kg). If undefined TA is estimated from salinity using guess_TA .
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

moles of CO₂ gas to be added to the flush reservoir.

Note

It is assumed that the entire reservoir is drained into the respirometer during the flush. It is also assumed that all of the CO₂ added dissolves and remains in solution. This can be achieved (almost completely) by bubbling CO₂ according to Jokiel et al. 2014.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Jokiel PL, Bahr KD, Rodgers KS. 2014. Low-cost, high-flow mesocosm system for simulating ocean acidification with CO₂ gas. *Limnol Oceanogr Methods*. 12:313–322.

See Also

[co2_rate](#), [flush_carb](#), [carb](#), [peri_pump](#)

Examples

```
# I want the respirometer to have a pCO2 = 1000 uatm. It currently has a pH of 7.6 and is 90 L.
# If I have a 200 L reservoir with pH = 7.9 seawater, how much CO2 do I need
# to add to the flush reservoir?
co2_flush(goal_pco2 = 1000, resp_pH = 7.6, resp_vol = 90, flush_pH = 7.9, flush_vol = 200)
```

 co2_rate

Calculate CO2 to add to a respirometer intake flow

Description

Calculates the moles of CO₂ gas to be added to a respirometer intake seawater flow to achieve the desired pCO₂ in the respirometer. Useful for ocean acidification experiments where CO₂ treatments are desired. Can be used for acclimation before a trial begins or for use with flow-through respirometry.

Usage

```
co2_rate(goal_pco2, init_pH, flow_rate, temp = 25, sal = 35, TA = NULL,
         atm_pres = 1013.25, M02 = NULL, RQ = 1)
```

Arguments

goal_pco2	the desired pCO ₂ in the respirometer (uatm).
init_pH	ambient pH of the intake flow (total scale).
flow_rate	rate of water flow into the respirometer (liters / minute).
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu. If sal < 26 psu, then TA must be provided.
TA	(optional) total alkalinity (umol / kg). If undefined TA is estimated from salinity using guess_TA .
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.
M02	(optional) oxygen consumption rate (umol / hr). If defined, the CO ₂ to be added is reduced to compensate for the CO ₂ produced by the organism.
RQ	(optional) respiratory quotient: ratio of CO ₂ produced / O ₂ consumed. Only used if M02 is defined. Default is 1.

Value

moles of CO₂ gas to be added to the intake flow per minute.

Note

It is assumed that all of the CO₂ added dissolves and remains in solution. This can be achieved (almost completely) by bubbling CO₂ according to Jokiel et al. 2014.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Jokiel PL, Bahr KD, Rodgers KS. 2014. Low-cost, high-flow mesocosm system for simulating ocean acidification with CO₂ gas. *Limnol Oceanogr Methods*. 12:313–322.

See Also

[co2_add](#), [flush_carb](#), [carb](#), [peri_pump](#)

Examples

```
# I want the respirometer to have a pCO2 = 1000 uatm. How much CO2 per minute do I need
# to add to the intake flow if the ambient pH is 8.1 and it is flowing at 3 LPM?
co2_rate(goal_pco2 = 1000, init_pH = 8.1, flow_rate = 3)
```

conv_NH4

Convert between units of ammonium

Description

Ammonia or nitrogen excretion can be measured in a variety of ways. Convert between different measurements.

Usage

```
conv_NH4(n_waste, from = "umol_NH4", to = "all")
```

Arguments

n_waste	a numeric vector of the ammonia or nitrogen value(s).
from	a string describing the unit used to measure n_waste. Default is "umol_NH4" Options are: <ul style="list-style-type: none">• umol_NH4• mg_NH4• mg_N
to	a single string either describing the unit to which the conversion should be conducted (options are the same as in from), or the string "all" to return all units.

Details

The sum of NH₄⁺ and NH₃ species are considered. Conversions are based on relationships and values from the package [marelac](#).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

Examples

```
conv_NH4(n_waste = 100)
conv_NH4(n_waste = 100, from = 'mg_N')
conv_NH4(n_waste = 100, from = 'mg_N', to = 'umol_NH4')
```

 conv_o2

Convert between units of oxygen partial pressure and concentration

Description

Unfortunately, a consensus on the best way to express how much oxygen is in water has not been formed to date. Until then, this function converts between all commonly used forms of dissolved O₂ measurements.

Usage

```
conv_o2(o2 = 100, from = "percent_a.s.", to = "all", temp = 25,
        sal = 35, atm_pres = 1013.25)
```

Arguments

o2	a numeric vector of the O ₂ value(s). Default is 100.
from	a string describing the unit used to measure o ₂ . Default is "percent_a.s." Options are: <ul style="list-style-type: none"> • percent_a.s. (percent air saturation) • percent_o2 • hPa • kPa • torr • mmHg • inHg • mg_per_l • umol_per_l • mmol_per_l • ml_per_l • mg_per_kg • umol_per_kg • mmol_per_kg • ml_per_kg
to	a single string either describing the unit to which the conversion should be conducted (options are the same as in from), or the string "all" to return all units.
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Details

Conversions are based on relationships and values from the package [marelac](#).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

Examples

```
conv_o2(o2 = 50)
conv_o2(o2 = 1:50, from = "umol_per_l", to = "ml_per_l", temp = 10, sal = 0,
atm_pres = 1100)
conv_o2()[c('mmHg', 'kPa')]
```

correct_bubble	<i>Adjust respirometer volume for bubbles</i>
----------------	---

Description

Given the volume of the respirometer and the volume of bubbles or air space, the moles of O₂ in the system are calculated, and the volume of a respirometer holding the same quantity of O₂ with only water is returned.

Usage

```
correct_bubble(resp_vol, bubble_vol, temp = 25, sal = 35,
atm_pres = 1013.25)
```

Arguments

resp_vol	volume of the respirometer (liter).
bubble_vol	volume of the gas bubbles or headspace (mL).
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Details

Depending on temperature and salinity, air holds 20,000x as much O₂ as water per unit volume, thus small air bubbles in a respirometer can dramatically increase the amount of O₂ an organism has to consume to lower the pO₂ or aqueous [O₂]. Thus air bubbles lead to underestimations of MO₂. To correct for this in MO₂ calculations after measurement, the volume of the respirometer can be increased. This function calculates the volume needed for MO₂ calculations as a function of the volume of air space. Caution: allowing air bubbles into a respirometer is not recommended, even with this post-measurement adjustment. A small error in bubble volume estimation can lead to a large error in calculated metabolic rate.

Value

The volume of a respirometer holding an equivalent quantity of O₂ filled only with water.

Note

Due to the high concentration of O₂ in air, very small errors in bubble volume estimates can lead to very large differences in the volume returned. Only trust the returned value if you are very confident of the accuracy of your bubble volume estimate.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[molvol](#)

Examples

```
correct_bubble(resp_vol = 50, bubble_vol = 10) # a 10 mL bubble makes a huge difference!  
  
correct_bubble(resp_vol = 50, bubble_vol = 1, temp = 10, sal = 0)  
# in calculating M02, a volume of 63.8 L should be used rather than the true 50 L.
```

flush_carb

Estimate carbonate chemistry after a flush

Description

Given the seawater pH inside the respirometer and in the flush reservoir, the new carbonate parameters (including pH) in the respirometer after the flush are estimated.

Usage

```
flush_carb(resp_vol, flow_rate, duration, resp_pH, flush_pH, temp = 25,  
           sal = 35, TA = NULL, atm_pres = 1013.25)
```

Arguments

resp_vol	volume of the respirometer (liter).
flow_rate	rate of water flow into the respirometer (liters / minute).
duration	duration of the flush (minutes).
resp_pH	pH inside the respirometer before the flush (total scale).
flush_pH	pH of the water used for flushing the respirometer (total scale).
temp	temperature (°C). Default is 25 °C.

sal	salinity (psu). Default is 35 psu. If sal < 26 psu, then TA must be provided.
TA	(optional) total alkalinity (umol / kg). If undefined TA is estimated from salinity using guess_TA .
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

A data frame returned by [carb](#).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[carb](#), [flush_water](#)

Examples

```
flush_carb(resp_vol = 90, flow_rate = 10, duration = 3, resp_pH = 7.8, flush_pH = 8.1)

# What will be the pH in the respirometer after this flush?
flush_carb(resp_vol = 90, flow_rate = 10, duration = 3, resp_pH = 7.8, flush_pH = 8.1)$pH
```

flush_o2	<i>Estimate dissolved O2 after a flush</i>
----------	--

Description

Calculate the pO2 or [O2] in a respirometer after a flush. Given 5 of the 6 parameters, the 6th parameter is calculated.

Usage

```
flush_o2(resp_vol, flow_rate, duration, resp_o2, flush_o2, final_o2)
```

Arguments

resp_vol	volume of the respirometer (liter).
flow_rate	rate of water flow into the respirometer (liters / minute).
duration	duration of the flush (minutes).
resp_o2	O2 inside the respirometer before the flush (units do not matter as long as it is consistent with flush_o2 and final_o2).
flush_o2	O2 of the water used for flushing the respirometer (units do not matter as long as it is consistent with resp_o2 and final_o2).
final_o2	O2 of the water in the respirometer at the end of the flush (units do not matter as long as it is consistent with resp_o2 and flush_o2).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[flush_water](#), [flush_carb](#)

Examples

```
# What will be the pO2 in the respirometer after this flush?
flush_o2(resp_vol = 90, flow_rate = 10, duration = 3, resp_o2 = 15, flush_o2 = 21)

# I want to bring the pO2 back up to 95% air saturation. How long do I need to flush?
flush_o2(resp_vol = 20, flow_rate = 2, resp_o2 = 75, flush_o2 = 99, final_o2 = 95)
```

flush_water

Find percent of water exchanged after a flush

Description

Calculate the proportion of water in a respirometer that is new after a flush. Useful for intermittent respirometry. Given 3 of the first 4 parameters, the 4th parameter is calculated.

Usage

```
flush_water(vol, flow_rate, duration, perc_fresh, plot = FALSE)
```

Arguments

vol	volume of the respirometer (liter).
flow_rate	rate of water flow into the respirometer (liters / minute).
duration	duration of the flush (minutes).
perc_fresh	percent of the respirometer volume that is new flushed water.
plot	logical. Plot the percent exchanged as a function of flow rate and duration to see what effect would result if the rate or duration are changed.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Steffensen JF. 1989. Some errors in respirometry of aquatic breathers: How to avoid and correct for them. *Fish Physiol Biochem.* 6:49–59. Equation 5.

See Also

[flush_carb, min_flow](#)

Examples

```
# What proportion of a 90 L respirometer is exchanged after 20 minutes of flow at 2 LPM?
flush_water(vol = 90, flow_rate = 2, duration = 20)

# Would it be worth it to extend the flush another five minutes? How much would that
# improve the exchange?
flush_water(vol = 90, flow_rate = 2, duration = 20, plot = TRUE)
# Another five minutes would increase exchange by nearly 10%.
# Perhaps that's worth the extra time...

# Visualize flushing
vol = 150
flow_rate = seq(0, 10, by = 0.5)
duration = 0:60
perc_fresh = outer(flow_rate, duration, function(flow_rate, duration){
  flush_water(vol = vol, flow_rate = flow_rate, duration = duration)$perc_fresh
})
persp(flow_rate, duration, perc_fresh, xlab = 'Flow rate (LPM)', ylab = 'Duration (min)',
      zlab = '% exchange', theta = 45, phi = 15, expand = 0.5, ticktype = 'detailed', nticks = 10)
```

goal_flush_pH	<i>Calculate goal pH of a flush reservoir to achieve the post-flush goal pCO₂</i>
---------------	--

Description

Calculates the pH of a flush reservoir that is needed to achieve the goal pCO₂ after the flush reservoir has been drained into the respirometer.

Usage

```
goal_flush_pH(goal_pco2, resp_pH, resp_vol, flush_vol, flush_remain = 0,
  temp = 25, sal = 35, TA = NULL, atm_pres = 1013.25)
```

Arguments

goal_pco2	the desired pCO ₂ in the respirometer after the flush (uatm).
resp_pH	pH inside the respirometer before the flush (total scale).
resp_vol	volume of the respirometer (liter).
flush_vol	volume of the flush reservoir (liter).
flush_remain	volume of the flush reservoir that will remain after the flush (liter).
temp	temperature (°C). Default is 25 °C.

sal	salinity (psu). Default is 35 psu. If sal < 26 psu, then TA must be provided.
TA	(optional) total alkalinity (umol / kg). If undefined TA is estimated from salinity using guess_TA .
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

pH needed in the flush reservoir to achieve the goal pCO₂ post-flush (total scale).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[co2_rate](#), [flush_carb](#), [carb](#), [peri_pump](#)

Examples

```
# I want the respirometer to have a pCO2 = 1000 uatm. It currently has a pH of 7.6 and is 90 L.
# If I have a 200 L reservoir which will be drained completely, what do I want
# the pH of the reservoir to be?
goal_flush_pH(goal_pco2 = 1000, resp_pH = 7.6, resp_vol = 90, flush_vol = 200)
```

guess_TA	<i>Estimate total alkalinity from salinity</i>
----------	--

Description

Estimate total alkalinity from salinity and temperature of surface seawater according to Lee et al. 2006. Useful when a rough guess of TA is needed because measuring TA is not possible or practical.

Usage

```
guess_TA(temp = 25, sal = 35, region = NULL, extend = TRUE)
```

Arguments

temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu. $31 \leq \text{sal} \leq 38$; may be narrower for some regions.
region	(optional) geographic region. Options are "(Sub)tropics", "Equatorial Upwelling Pacific", "North Atlantic", "North Pacific", and "Southern Ocean". Default is NULL. If undefined, the average from all these regions is used.
extend	logical. If salinity is ≤ 5 psu outside of the bounds defined by Lee et al. 2006 (see Details), should a guess be extrapolated? Default is TRUE.

Details

(Sub)tropics $\text{temp} \geq 20$ and $31 \leq \text{sal} \leq 38$

Equatorial Upwelling Pacific $\text{temp} \geq 18$ and $31 \leq \text{sal} \leq 36.5$

North Atlantic $0 \leq \text{temp} \leq 20$ and $31 \leq \text{sal} \leq 37$

North Pacific $\text{temp} \leq 20$ and $31 \leq \text{sal} \leq 35$

Southern Ocean $\text{temp} \leq 20$ and $33 \leq \text{sal} \leq 36$

Estimates total alkalinity using the equations provided by Lee et al. 2006 (Geophysical Research Letters). While these equations are designed for open ocean environments, they can provide a rough estimate even for coastal environments. For improved estimate accuracy, the geographic region can be provided. The North Pacific region is longitude-dependent so a longitude of 150 °W is assumed which provides a typical value within the range. Only applicable for surface waters, not very accurate for the ocean interior.

Value

An estimate of the total alkalinity (umol / kg). If NA or NaN are returned, confirm the temp and sal values are within acceptable ranges for the region of interest.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Lee K, Tong LT, Millero FJ, Sabine CL, Dickson AG, Goyet C, Park G-H, Wanninkhof R, Feely RA, Key RM. 2006. Global relationships of total alkalinity with salinity and temperature in surface waters of the world's oceans. Geophys Res Lett. 33:L19605.

See Also

[predict_pH](#)

Examples

```
guess_TA(temp = 22, sal = 33)
guess_TA(temp = 12, sal = 33, region = "North Atlantic")
guess_TA(temp = 20, sal = 31:35)

guess_TA(sal = 31) # salinity is within bounds
guess_TA(sal = 30) # salinity is outside the bounds and TA is extrapolated
guess_TA(sal = 30, extend = FALSE) # do not extrapolate TA
guess_TA(sal = 25, extend = TRUE) # will not extrapolate with sal > 5 psu out of bounds
```

`guess_when`*Estimate when the O2 level will reach a defined level*

Description

Estimates the time at which O2 will reach a defined level assuming a linear change in O2 over time.

Usage

```
guess_when(past_o2, past_time, goal_o2, plot = TRUE)
```

Arguments

<code>past_o2</code>	a numeric vector of at least two oxygen measurements previously during the trial.
<code>past_time</code>	a vector of timepoints corresponding to when <code>past_o2</code> values were recorded. Can be a numeric vector for duration since trial began or a POSIX vector of time values.
<code>goal_o2</code>	a numeric vector or single value describing the O2 level of interest.
<code>plot</code>	logical. Do you want to see a plot to visualize this prediction?

Value

A prediction of the time when O2 will reach `goal_o2`. If `past_time` is numeric, then a numeric value(s) will be returned. If POSIX, then POSIX will be returned.

Note

Viewing the plot can be valuable if the O2 consumption or production is not linear.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[predict_pH](#), [predict_NH4](#)

Examples

```
guess_when(past_o2 = rnorm(n = 10, mean = 100:91), past_time = 1:10, goal_o2 = 75, plot = FALSE)
guess_when(past_o2 = rnorm(n = 10, mean = 100:91, sd = 10), past_time = 1:10, goal_o2 = 75)
# Viewing the plot can be helpful to see how trustworthy the prediction is
# when signal:noise is low.
```

import_presens	<i>Import data from a PreSens O2 transmitter</i>
----------------	--

Description

Imports the standard text file output from most single channel PreSens fiber optic O2 transmitters and converts the data into a data frame.

Usage

```
import_presens(file, o2_unit = "percent_a.s.", date = "%d/%m/%y",
               sal = 35, all_cols = FALSE)
```

Arguments

file	a character string. The filepath for the file to be read.
o2_unit	a character string. The unit of O2 measurement to be output in the data.frame. Options are described in conv_o2 .
date	a character string. The date format to be passed to strptime .
sal	salinity of water sample (psu). Default is 35 psu. Ignored for Fibox 4 files since salinity is provided by the file.
all_cols	logical. For Fibox 4 files only. Should all columns (including calibration data and serial numbers) be output?

Details

The following PreSens fiber optic O2 transmitters are supported:

- Fibox 4
- Fibox 3
- Fibox 3 trace
- Fibox 3 LCD trace
- Microx TX3
- Microx TX3 trace

If you would like support for another PreSens O2 meter, email the package maintainer a data file from the device you would like supported. It is very important to note that the PreSens fiber optics O2 transmitters that are supported with this function (except the Fibox 4) DO NOT account for salinity (i.e. they assume salinity = 0 ppt). If the water sample measured was not fresh water, the oxygen concentrations (e.g. mg per liter or umol per liter) are incorrect in the PreSens txt file. This function corrects these O2 concentrations based on the salinity value defined by the `sal` argument. Absolute partial pressures (i.e. hPa and torr) will also be slightly different due to the slight influence of salinity on water's vapor pressure. This difference is typically ~0.05% of the recorded value.

Value

A data frame is returned.

TIME Date and time, POSIXct format.

DURATION Duration of measurement trial (minutes).

O2 Oxygen measurement in desired unit as determined by `o2_unit`.

PHASE Phase recorded. Phase is inversely related to O2.

AMPLITUDE Amplitude recorded. Amplitude is an indicator of the quality of the signal. A low amplitude warning is produced by the transmitter below 2500.

TEMP Temperature recorded or defined at beginning of measurement trial.

ATM_PRES Atmospheric pressure (mbar).

SAL Salinity (psu).

ERROR_CODE Error code from transmitter. See PreSens user manual for translation of error code.

Note

Oxygen conversions are estimates based on the [marelac](#) package and therefore differ slightly from the conversions provided by PreSens.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[import_witrox](#), [conv_o2](#)

Examples

```
## Not run:

# Import a Fibox 3 file.
file <- system.file('extdata', 'fibox_3_file.txt', package = 'respirometry')
import_presens(file, o2_unit = 'umol_per_l', sal = 25)

# Import a Fibox 4 file.
file <- system.file('extdata', 'fibox_4_file.csv', package = 'respirometry')
import_presens(file = file, date = '%d-%b-%Y')

## End(Not run)
```

import_witrox

Import data from a Loligo Systems Witrox transmitter

Description

Imports the standard txt file output from Loligo Systems Witrox fiber optic O2 transmitters and converts the data into one or more data frames.

Usage

```
import_witrox(file, o2_unit = "percent_a.s.",
              date = "%m/%d/%Y %I:%M:%S %p", overwrite_sal = NULL,
              drop_channels = TRUE, split_channels = FALSE)
```

Arguments

file	a character string. The filepath for the file to be read.
o2_unit	a character string. The unit of O2 measurement to be output in the data.frame. Options are described in conv_o2
date	a character string. The date format to be passed to strptime .
overwrite_sal	Default NULL. To overwrite the salinity value(s) from calibration, enter a single numeric value for all channels or a numeric vector with values for each channel. Salinity of water sample (psu).
drop_channels	logical. Should channels without any O2 data be dropped? Default is TRUE.
split_channels	logical. Should a list of data frames be returned with a separate data frame for each channel? Default is FALSE.

Details

The following Loligo Systems fiber optic O2 transmitters are supported:

- Witrox 4

If you would like support for the Witrox 1, email me a data file from this device.

Value

A data frame (or list of data frames) is returned.

TIME Date and time, POSIXlt format.

DURATION Duration of measurement trial (minutes).

ATM_PRES Atmospheric pressure (mbar).

CH_X_PHASE Phase recorded. Phase is inversely related to O2.

CH_X_TEMP Temperature recorded or defined at beginning of measurement trial.

CH_X_SAL Salinity (psu).

CH_X_O2 Oxygen measurement in desired unit as determined by o2_unit.

... Channel columns (CH_...) are repeated for each channel.

If `split_channels = TRUE`, then "CH_X_" is removed from the column names and multiple data frames are returned in a list.

Note

Oxygen conversions are estimates based on the [marelac](#) package.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[import_presens](#), [conv_o2](#)

Examples

```
## Not run:
file <- system.file('extdata', 'witrox_file.txt', package = 'respirometry')
import_witrox(file, o2_unit = 'umol_per_l')

# Oops. I forgot to change the salinity value when I calibrated
# the instrument. Override the values in the file for 35 psu.
import_witrox(file, o2_unit = 'umol_per_kg', overwrite_sal = 35)

# I want each channel as a separate data frame.
data_list = import_witrox(file, split_channels = TRUE)
data_list$CH_3 # here's the channel 3 data frame.

## End(Not run)
```

max_MO2

Maximum MO2 supported by flow rate

Description

Calculates the maximum oxygen consumption rate (MO2) supported by a respirometer with a given flow rate. Useful for ensuring an acclimating animal maintains a normoxic environment.

Usage

```
max_MO2(flow_rate, min_pO2 = 90, pO2_in = 100, temp = 25, sal = 35,
        atm_pres = 1013.25)
```

Arguments

flow_rate	water flow rate into respirometer (liters / min).
min_pO2	minimum pO2 acceptable in respirometer (% air saturation). Default is 90% air saturation.
pO2_in	pO2 of water entering respirometer (% air saturation). Default is 100% air saturation.
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

The maximum whole-animal oxygen consumption rate (umol / hr) that can be sustained.

Note

Keep in mind that most organisms are very stressed upon being placed in a respirometer and their MO2 may be much higher than basal MO2.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Steffensen JF. 1989. Some errors in respirometry of aquatic breathers: How to avoid and correct for them. *Fish Physiol Biochem.* 6:49–59. Equation 8.

See Also

[min_flow](#), [flush_water](#)

Examples

```
max_MO2(flow_rate = 1)

# What is the maximum MO2 organism I can place in my respirometer and still maintain at
# least 75% air saturation when the intake fresh water is 1.5 LPM, 10 °C and 90% air saturated?
(max_mo2 = max_MO2(flow_rate = 1.5, min_pO2 = 75, pO2_in = 90, temp = 10, sal = 0))

# If a 300 g individual has an MO2 of 2000 umol/hr, how big of an animal can I use?
scale_MO2(mass_1 = 300, MO2_1 = 2000, MO2_2 = max_mo2) # I can almost support a 1 kg individual!
```

mean_pH	<i>Mean pH by [H+]</i>
---------	------------------------

Description

Calculates mean pH from a vector of pH values by averaging [H+] rather than numerical pH values.

Usage

```
mean_pH(pH, na.rm = FALSE, ...)
```

Arguments

pH	a numeric vector of pH values.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Details

Since pH is on a logarithmic scale, averaging pH values directly does not provide the true arithmetic mean of what is likely truly important to the organism, [H+] (however, see Boutilier and Shelton 1980). Thus, the pH values are converted to [H+] then averaged and converted back to a mean pH value.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Boutilier RG, Shelton G. 1980. The statistical treatment of hydrogen ion concentration and pH. J Exp Biol. 84:335–339.

Examples

```
mean_pH(c(7, 8)) # 7.26 rather than 7.5!
```

min_flow	<i>Minimum flow rate to support MO2</i>
----------	---

Description

Calculates the minimum flow rate into a respirometer required to maintain a high pO₂. Useful for ensuring an acclimating animal maintains a normoxic environment. It can also be used to estimate the flow rate needed for a given pO₂ decrease desired for flow-through respirometry.

Usage

```
min_flow(MO2, min_pO2 = 90, pO2_in = 100, temp = 25, sal = 35,  
         atm_pres = 1013.25)
```

Arguments

MO2	whole-animal oxygen consumption rate (umol / hour).
min_pO2	minimum pO ₂ acceptable in respirometer (% air saturation). Default is 90% air saturation.
pO2_in	pO ₂ of water entering respirometer (% air saturation). Default is 100% air saturation.
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

The flow rate (liters / min) into the respirometer required for the steady state pO₂ to be min_pO₂.

Note

Keep in mind that most organisms are very stressed upon being placed in a respirometer and their MO₂ may be much higher than basal MO₂.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

References

Steffensen JF. 1989. Some errors in respirometry of aquatic breathers: How to avoid and correct for them. *Fish Physiol Biochem.* 6:49–59. Equation 8.

See Also

[max_MO2](#), [flush_water](#)

Examples

```

min_flow(MO2 = 1000)

# What is the minimum flow rate required to maintain at least 75% air saturation in a
# respirometer with an organism(s) with an oxygen consumption rate of 1000 umol/h
# when the intake fresh water is 10 °C and 90% air saturated?
min_flow(MO2 = 1000, min_pO2 = 75, pO2_in = 90, temp = 10, sal = 0)

```

peri_pump

Calculate peristaltic pump gaseous flow rate

Description

Given the number of moles of a gas, calculates the liters to run through a peristaltic pump.

Usage

```

peri_pump(mol, species = "O2", temp = 25, reg_pres, reg_unit = "psi",
          atm_pres = 1013.25)

```

Arguments

mol	number of moles to go through a peristaltic pump.
species	character string describing the gas species. Options are available from molvol . Default is "O2".
temp	temperature (°C). Default is 25 °C.
reg_pres	gauge pressure from the gas regulator into the peristaltic pump.
reg_unit	unit used in reg_pres. Default is "psi".
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Details

Most mass flow controllers are programmed with a "standard condition" something like 0 °C and 1013 mbar for which they account for the pressure and temperature of an incoming gas source. For setups without expensive mass flow controllers, a more affordable alternative is to use a peristaltic pump. These do not account for variations in incoming gas pressure and temperature and thus, it must be calculated to set the peristaltic pump to the correct RPM.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[co2_rate](#), [co2_add](#)

Examples

```
peri_pump(mol = 0.5, species = 'O2', temp = 10, reg_pres = 5, reg_unit = "kPa")  
# To flow 0.5 moles of O2, then flow 11.1 L.
```

predict_NH4

Predict NH4 concentration post-respiration

Description

Predicts the [NH4+] of seawater after a defined amount of oxygen consumption.

Usage

```
predict_NH4(o2_drop = 10, o2_unit = "percent_a.s.", o2_nh4_ratio,  
temp = 25, sal = 35, atm_pres = 1013.25)
```

Arguments

o2_drop	a numeric value or vector describing the change in O2. Default is 10.
o2_unit	a string describing the unit used to measure o2_drop. Default is "percent_a.s." Options are from conv_o2 .
o2_nh4_ratio	molar ratio of O2 consumed to NH4+ produced.
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Details

Given a known amount of oxygen consumed and an estimated O:N ratio, the amount of NH4 produced can be estimated. Production or consumption of ammonium by "background" microbes or conversion of ammonium to nitrite and nitrate is ignored since bacteria in the respirometer are typically sought to be in low levels.

Value

The predicted NH4+ produced in mg/l.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[conv_o2](#), [conv_NH4](#)

Examples

```
predict_NH4(o2_drop = 25, o2_nh4_ratio = 10)
```

predict_pH	<i>Predict pH post-respiration</i>
------------	------------------------------------

Description

Predicts the pH of seawater after a defined amount of oxygen consumption.

Usage

```
predict_pH(start_o2 = 100, end_o2, start_pH, temp = 25, sal = 35,
           RQ = 1, TA = NULL, all_carb = FALSE)
```

Arguments

start_o2	pO ₂ at the start of the measurement (% air saturation). Default is 100% air saturation.
end_o2	pO ₂ at the end of the measurement (% air saturation).
start_pH	seawater pH (total scale) at the start of the measurement.
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu. If sal < 26 psu, then TA must be provided.
RQ	respiratory quotient: ratio of CO ₂ produced / O ₂ consumed. Default is 1.
TA	(optional) total alkalinity (umol / kg). If undefined TA is estimated from salinity using guess_TA .
all_carb	logical. Should all carbonate chemistry parameters be returned? Default is FALSE.

Details

Given a known amount of oxygen consumed and an estimated respiratory quotient (see [Q10](#)), the amount of CO₂ produced can be estimated. From this CO₂ production estimate, the carbonate chemistry of the seawater can be estimated. Atmospheric pressure is assumed.

Value

If all_carb is FALSE, then a list of the predicted pH (total scale) at the end of the measurement and the predicted pCO₂ (uatm) are returned. If all_carb is TRUE, then the predicted carbonate chemistry parameters are returned from [carb](#).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[carb, guess_TA](#)

Examples

```
predict_pH(end_o2 = 75, start_pH = 8.1)
predict_pH(start_o2 = 75, end_o2 = 50, start_pH = 7.96, temp = 15, sal = 33, RQ = 0.88)

# I know pH at the end was 7.8, but what was pH at the beginning?
predict_pH(start_o2 = 75, end_o2 = 100, start_pH = 8.013536) # reverse the order
```

Q10

Parameters of Q10 Temperature Coefficient

Description

Returns the unknown parameter given 4 of 5 parameters from Q10 temperature coefficient calculation for chemical or biological systems.

Usage

```
Q10(Q10, R1, R2, T1, T2)
```

Arguments

Q10	factor by which rate changes due to 10 °C increase in temperature.
R1	rate 1.
R2	rate 2.
T1	temperature 1 (in °C).
T2	temperature 2 (in °C).

Details

Given four parameters, the fifth parameter will be returned.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[scale_M02](#)

Examples

```

Q10(R1 = 5, R2 = 10, T1 = 10, T2 = 20) # Returns Q10; = 2
Q10(Q10 = 2.66, R1 = 5, T1 = 10, T2 = 20) # Returns R2; = 13.3

# My species has an M02 of 9.5 umol/g/h at 10 *C. What M02 should I expect at 13 *C?
Q10(Q10 = 2, R1 = 9.5, T1 = 10, T2 = 13) # expect ~11.7 umol/g/h at 13 *C.

# A 100 g individual at 10 *C has an M02 of 1270 umol/h. How much
# would a 250 g individual likely consume at 14 *C?
Q10(Q10 = 2, R1 = scale_M02(mass_1 = 100, M02_1 = 1270, mass_2 = 250)$M02_2, T1 = 10, T2 = 14)

# Visualize M02 scaling by mass and temperature:
mass = seq(10, 200, 10)
temp = 10:25
base_mass = 50
base_temp = 20
base_M02 = 750
mo2 = outer(mass, temp, function(mass, temp){
  scale_M02(mass_1 = base_mass, mass_2 = mass, M02_1 = Q10(Q10 = 2, R1 = base_M02,
    T1 = base_temp, T2 = temp)$R2)$M02_2
})
persp(mass, temp, mo2, xlab = 'Mass (g)', ylab = 'Temperature (*C)', zlab = 'M02 (umol / hr)',
  theta = 35, phi = 15, expand = 0.5, ticktype = 'detailed', nticks = 10)

```

 respirometry

Tools for Conducting Respirometry Experiments

Description

Provides tools to enable the researcher to more precisely conduct respirometry experiments. Strong emphasis is on aquatic respirometry. Tools focus on helping the researcher setup and conduct experiments. Analysis of the resulting data is not a focus since analyses are often specific to a particular setup, and thus are better created by the researcher individually. This package provides tools for intermittent, flow-through, and closed respirometry techniques.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

RQ

Calculate respiratory quotient

Description

Calculates the respiratory quotient (RQ), or ratio of CO₂ produced to O₂ consumed between observations. To calculate CO₂ produced, either DIC or both pH and TA must be provided.

Usage

```
RQ(o2, o2_unit = "percent_a.s.", pH = NULL, TA = NULL, DIC = NULL,
   temp = 25, sal = 35, atm_pres = 1013.25)
```

Arguments

o2	a numeric vector of O2 values with a length of at least 2.
o2_unit	a string describing the unit used to measure o2. Default is "percent_a.s." Options are from conv_o2 .
pH	pH (total scale). Elements must align with o2 vector.
TA	total alkalinity (umol / kg). May be either a vector with length equal to o2 or a single numeric value.
DIC	dissolved inorganic carbon (umol / kg). Elements must align with o2 vector.
temp	temperature (°C). Default is 25 °C.
sal	salinity (psu). Default is 35 psu.
atm_pres	atmospheric pressure (mbar). Default is 1013.25 mbar.

Value

ratio of CO2 produced to O2 consumed.

Note

If you want a rough estimate of RQ, but only have pH measurements, TA can be estimated from salinity using [guess_TA](#).

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[conv_o2](#), [guess_TA](#)

Examples

```
o2_observations = c(21, 18, 14.5, 7)
pH_observations = c(8.05, 7.98, 7.86, 7.65)
TA_observations = c(2222, 2219, 2208, 2214)

RQ(o2 = o2_observations, o2_unit = 'kPa', pH = pH_observations,
   TA = TA_observations, temp = 20, sal = 33)

DIC_observations = c(2222, 2250, 2284, 2355)
RQ(o2 = o2_observations, o2_unit = 'kPa', DIC = DIC_observations)

RQ(o2 = o2_observations, o2_unit = 'kPa', pH = pH_observations, TA = 2032)
```

 scale_MO2

Scale metabolic rate by size

Description

For most organisms, metabolic rate does not scale linearly, but rather according to a power function. This function estimates MO2 or size of an individual organism given the MO2 and size of another individual of a different size.

Usage

```
scale_MO2(mass_1, MO2_1, mass_2, MO2_2, b = 0.75)
```

Arguments

mass_1	animal mass for MO2_1.
MO2_1	metabolic rate for mass_1.
mass_2	animal mass for MO2_2.
MO2_2	metabolic rate for mass_2.
b	scaling coefficient for MO2. Default is 0.75.

Details

$$(MO2 = b0 * M^b)$$

where $b0$ is species-specific normalization constant, M is mass and b is the scaling coefficient which is 0.75 for many organisms.

For scaling of **mass-specific** metabolic rates, use $b = -0.25$ rather than $b = 0.75$.

Author(s)

Matthew A. Birk, <matthewabirk@gmail.com>

See Also

[Q10](#)

Examples

```
# I know a species has an SMR of 800 umol O2/h at 200 g.
# What would be a likely SMR for a 300 g individual?
scale_MO2(mass_1 = 200, MO2_1 = 800, mass_2 = 300)

# Some squids have a much higher scaling coefficient:
scale_MO2(mass_1 = 200, MO2_1 = 800, mass_2 = 300, b = 0.92)
```

```
# A 100 g individual at 10 *C has an M02 of 1270 umol/h. How much
# would a 250 g individual likely consume at 14 *C?
Q10(Q10 = 2, R1 = scale_MO2(mass_1 = 100, M02_1 = 1270, mass_2 = 250)$M02_2, T1 = 10, T2 = 14)

# Visualize M02 scaling by mass and temperature:
mass = seq(10, 200, 10)
temp = 10:25
base_mass = 50
base_temp = 20
base_M02 = 750
mo2 = outer(mass, temp, function(mass, temp){
  scale_MO2(mass_1 = base_mass, mass_2 = mass, M02_1 = Q10(Q10 = 2, R1 = base_M02,
    T1 = base_temp, T2 = temp)$R2)$M02_2
})
persp(mass, temp, mo2, xlab = 'Mass (g)', ylab = 'Temperature (*C)', zlab = 'M02 (umol / hr)',
  theta = 35, phi = 15, expand = 0.5, ticktype = 'detailed', nticks = 10)
```

Index

carb, [4](#), [5](#), [7](#), [11](#), [14](#), [26](#), [27](#)
closed, [2](#)
co2_add, [3](#), [7](#), [24](#)
co2_flush, [4](#)
co2_rate, [4](#), [5](#), [6](#), [14](#), [24](#)
conv_NH4, [7](#), [25](#)
conv_o2, [8](#), [17–20](#), [25](#), [29](#)
correct_bubble, [9](#)

flush_carb, [4](#), [5](#), [7](#), [10](#), [12–14](#)
flush_o2, [11](#)
flush_water, [3](#), [11](#), [12](#), [12](#), [21](#), [23](#)

goal_flush_pH, [13](#)
guess_TA, [4–6](#), [11](#), [14](#), [14](#), [26](#), [27](#), [29](#)
guess_when, [16](#)

import_presens, [17](#), [20](#)
import_witrox, [18](#), [19](#)

marelac, [7](#), [9](#), [18](#), [20](#)
max_MO2, [20](#), [23](#)
mean_pH, [22](#)
min_flow, [13](#), [21](#), [23](#)
molvol, [10](#), [24](#)

peri_pump, [4](#), [5](#), [7](#), [14](#), [24](#)
predict_NH4, [16](#), [25](#)
predict_pH, [15](#), [16](#), [26](#)

Q10, [26](#), [27](#), [30](#)

respirometry, [28](#)
respirometry-package (respirometry), [28](#)
RQ, [28](#)

scale_MO2, [27](#), [30](#)
strptime, [17](#), [19](#)