Package: mimsy (via r-universe)

September 16, 2024

Type Package

Title Calculate MIMS Dissolved Gas Concentrations Without Getting a Headache

Version 0.6.5

Date 2024-07-18

Description Calculate dissolved gas concentrations from raw MIMS (Membrane Inlet Mass Spectrometer) signal data. Use mimsy() on a formatted CSV file to return dissolved gas concentrations (mg and microMole) of N2, O2, Ar based on gas solubility at temperature, pressure, and salinity. See references Benson and Krause (1984), Garcia and Gordon (1992), Stull (1947), and Hamme and Emerson (2004) for more information. Easily save the output to a nicely-formatted multi-tab 'Excel' workbook with mimsy.save(). Supports dual-temperature standard calibration for dual-bath MIMS setups.

License MIT + file LICENSE

URL `https://github.com/michelleckelly/mimsy`, `https://michelleckelly.github.io/mimsy/`

BugReports `https://github.com/michelleckelly/mimsy/issues`

LazyData true

RoxygenNote 7.3.1

Encoding UTF-8

Imports dplyr, magrittr, lubridate, openxlsx, utils

Suggests knitr, rmarkdown, kableExtra, xfun

VignetteBuilder knitr

Repository https://michelleckelly.r-universe.dev

RemoteUrl https://github.com/michelleckelly/mimsy

RemoteRef HEAD

RemoteSha 41b11b0bdaddacbe8bb4a1a0f52687b2ecfbac8d

Contents

mimsy   mimsy *Calculate MIMS dissolved gas concentrations*

Description

Return dissolved gas concentrations in units of micromole and milligram from membrane inlet mass spectrometer (MIMS) signal data

Usage

```
mimsy(
  data,
  baromet.press,
  units,
  bg.correct = FALSE,
  tz = Sys.timezone(),
  salinity = 0
)
```

Arguments

`data`	the dataframe of MIMS data
`baromet.press`	the ambient barometric pressure while samples processed on the MIMS. Can be a vector, if more than one reading was taken.
`units`	the units of barometric pressure. Must be one of "atm", "hPa", "psi", "bar", or "Torr".
`bg.correct`	If ‘FALSE‘ (default), no background correction is applied. If ‘TRUE‘, background correction is applied.
`tz`	a character string that specifies which time zone to parse the date with. Defaults to the user’s current time zone setting. The string must be a time zone that is recognized by the user’s OS.
`salinity`	the salinity of standards, in units of per mille. Defaults to 0.

Value

a list, containing 5 dataframes:

‘$results‘ displays a summarized output, containing just identification columns (such as the user’s input of ‘Label‘ or ‘Project‘) and calculated concentration values (ex. ‘N2_uMol‘ and ‘N2_mg‘ for the concentration of dissolved dinitrogen gas, in microMoles and milligrams, respectively).

'$results' only contains sample results, but see '$results.full' below for results from both samples and standards.

'$solubility.Concentrations' displays the calculated concentrations (microMole / kg) of oxygen, nitrogen, and argon gas in water at specific temperature, pressure, and salinity. Row names correspond to 'std.temps'. See Garcia and Gordon (1992, Eqn. 8) for further information on oxygen saturation concentration calculations and the equations used herein. See Hamme and Emerson (2004, Table 4, and Eq. 1, 2) for the expansion of Garcia and Gordon's methods for use with calculating nitrogen and argon saturation concentrations.

'$calibration.Factors' are computed by taking the solubility concentrations at standard temperature divided by the average MIMS signal reading of each 'Group' at said temperature. These values will then be used internally to calculate the calibration curve, and are made available to the user for transparency.

'$calibration.DriftCorrection' is a data frame containing the slopes and intercepts of the calibration curve for each dissolved gas (ex. 'calslope_28' for the slope of the dinitrogen calibration curve), and the drift corrected slope and intercept values (ex. 'DRIFT.calslope_28') which take into account the machine drift due to time that occurs in-between standard readings. These drift values are computed by taking the slope between successive calibration (slope or intercept) values. These values are used internally, and are made available to the user for transparency.

'$results.full' is a data frame that contains the entire output from 'mimsy()', including results from both standards and samples, the initial MIMS signal data and signal ratios, the calibration curve correction factors for each reading (ex. 'INTERPOLATED.calfactor_28'), and the final concentration values.

References

Garcia, H., and L. Gordon (1992), *Oxygen solubility in seawater: Better fitting equations*, Limnology and Oceanography, 37(6).

Benson, B. B. & Krause, D. (1984). *The concentration and isotopic fractionation of oxygen dissolved in freshwater and seawater in equilibrium with the atmosphere.* Limnology and Oceanography, 29(3), 620-632. doi:10.4319/lo.1984.29.3.0620

Stull, D. R. (1947). *Vapor Pressure of Pure Substances. Organic and Inorganic Compounds.* Industrial & Engineering Chemistry, 39(4), 517-540. doi: 10.1021/ie50448a022

Hamme, R. C. & Emerson, S. R. (2004). *The solubility of neon, nitrogen and argon in distilled water and seawater*, Deep-Sea Research I, 51(11), 1517-1528.

Examples

```
data <-
  read.csv(file = system.file("extdata", "data.csv", package = "mimsy"),
           header = TRUE, stringsAsFactors = FALSE)

results <- mimsy(data, baromet.press = 981.2, units = 'hPa')
```

mimsy.save — mimsy.save *Save output to an Excel workbook*

Description

Easily save the full output of the ‘mimsy‘ calculation function to a nicely-formatted, multi-tab Excel .xlsx file.

Usage

```
mimsy.save(x, file)
```

Arguments

`x`	object to be written to file, the output of ‘mimsy‘ function
`file`	desired file name with .xlsx ending. Example: "mimsyCalculations.xlsx"

Value

Outputs an excel workbook to the user’s working directory

Examples

```
data <-
  read.csv(file = system.file("extdata", "data.csv", package = "mimsy"),
           header = TRUE, stringsAsFactors = FALSE)

results <- mimsy(data, baromet.press = 981.2, units = 'hPa')

mimsy.save(results, file = file.path(tempdir(), "results.xlsx"))
```

Index