

Package ‘biogas’

May 16, 2018

Type Package

Title Process Biogas Data and Predict Biogas Production

Version 1.10.3

Date 2018-05-16

Author Sasha D. Hafner [aut, cre],
Charlotte Rennuit [aut],
Jin Mi Triolo [ctb],
Ali Heidarzadeh Vazifehkhoran [ctb]

Maintainer Sasha D. Hafner <sasha.hafner@eng.au.dk>

VignetteBuilder knitr

Suggests knitr, ggplot2, testthat

Description High- and low-level functions for processing biogas data and predicting biogas production. Molar mass and calculated oxygen demand (COD') can be determined from a chemical formula. Measured gas volume can be corrected for water vapor and to (possibly user-defined) standard temperature and pressure. Gas quantity can be converted between volume, mass, and moles. Gas composition, cumulative production, or other variables can be interpolated to a specified time. Cumulative biogas and methane production (and rates) can be calculated using volumetric, manometric, or gravimetric methods for any number of reactors. With cumulative methane production data and data on reactor contents, biochemical methane potential (BMP) can be calculated and summarized, including subtraction of the inoculum contribution and normalization by substrate mass. Cumulative production and production rates can be summarized in several different ways (e.g., omitting normalization) using the same function. Biogas quantity and composition can be predicted from substrate composition and additional, optional data. Lastly, inoculum and substrate mass can be determined for planning BMP experiments.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-16 18:05:20 UTC

R topics documented:

calcCOD	2
comp	4
comp2	5
cumBg	6
interp	15
mass	17
mass2vol	18
massw	20
molMass	21
planBMP	22
predBg	24
s3compl	28
s3compw	29
s3lcombo	30
s3voll	31
s3volw	31
setup	32
setup2	33
stdVol	34
strawComp	36
strawMass	37
strawPressure	38
strawSetup	39
summBg	40
vol	44
vol2	45
vol2mass	46
vol2mol	48
Index	50

calcCOD	<i>Calculate Oxygen Demand</i>
---------	--------------------------------

Description

calcCOD is used to calculate the oxygen demand (“calculated oxygen demand”, or COD’ as described by Rittmann and McCarty (2001)) of a compound.

Usage

```
calcCOD(form)
```

Arguments

form a chemical formula, as a character vector, e.g., "C6H12O6" for glucose or "CH3COOH" for acetic acid, or c("C6H12O6", "CH3COOH") for both at once. Not case-sensitive for single letter elements (see 'Details').

Details

Based on Eqs. (2.2) and (2.3) in Rittmann and McCarty (2001) (p 128), but using molar mass calculated with [molMass](#). calcCOD is not case-sensitive as long as all elements have single letter abbreviations. The function is vectorized.

Value

A numeric vector with length equal to length of form with COD' in g of oxygen per g of compound.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

References

Rittmann, B., McCarty, P. 2001 *Environmental Biotechnology*. McGraw-Hill.

See Also

[molMass](#), [predBg](#)

Examples

```
calcCOD("C6H12O6")
calcCOD("CH3COOH")
calcCOD("CH3CH2OH")
calcCOD("CH4")
calcCOD("ch4")
calcCOD(c("C6H12O6", "CH3COOH", "CH3CH2OH"))
```

comp

Methane Content of Biogas

Description

Methane content (biogas composition) measurements from nine batch reactors.

Usage

```
data("comp")
```

Format

A data frame with 132 observations on the following 4 variables.

`id` identification code, a unique value for each reactor in the dataset. A factor with levels 2_1, 2_2, 2_3, 2_4, 2_5, 2_6, 2_7, 2_8, 2_9, 2_10, 2_11, and 2_12

`date.time` date and time of mass measurement, a POSIXct object.

`days` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`xCH4` biogas methane content as a mole fraction, excluding water and all other gases other than carbon dioxide, a numeric vector

Details

These data are meant to be example data for multiple functions, e.g., [interp](#), [cumBg](#), or [summBg](#). Reactors were 500 mL glass serum bottles with butyl rubber septa and screw caps. Methane and carbon dioxide contents were determined by gas chromatography using a thermal conductivity detector and normalised so methane and carbon dioxide sum to 1.0. Data in `vol`, `mass`, `comp`, and `setup` are from the same reactors.

Source

Measurements by Charlotte Rennuit and Ali Heidarzadeh Vazifekhoran.

Examples

```
data(comp)
```

`comp2`*Methane Content of Biogas*

Description

Methane content (biogas composition) measurements from 15 batch reactors.

Usage

```
data("comp2")
```

Format

A data frame with 360 observations on the following 3 variables.

`bottle` identification code, a unique value for each reactor in the dataset. A factor with levels 1_1, 1_2, 1_3, 2_1, 2_2, 2_2, ... through 5_3. The first number indicates the sample, the second the replicate.

`days` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`CH4.conc` biogas methane content as a mole fraction, excluding water and all other gases other than carbon dioxide, a numeric vector

Details

These data are meant to be example data for multiple functions, e.g., [interp](#), [cumBg](#), or [summBg](#). Reactors were 500 mL or 1000 mL glass serum bottles with butyl rubber septa and screw caps. Methane and carbon dioxide contents were determined by gas chromatography using a thermal conductivity detector and normalised so methane and carbon dioxide sum to 1.0. Data in `vol2`, `comp2`, and `setup2` are from the same reactors.

Source

Measurements made by Ali Heidarzadeh Vazifehkhoran

Examples

```
data(comp2)
```

cumBg

*Calculate Cumulative Biogas Production***Description**

cumBg (for *cumulative biogas*) calculates cumulative biogas, methane production and production rates from individual volume (or mass) and composition measurements for any number of reactors.

Usage

```

cumBg(
  # Main arguments
  dat, dat.type = 'vol', comp = NULL, temp = NULL,
  pres = NULL, interval = TRUE,
  data.struct = 'long',
  # Column names for volumetric method
  id.name = 'id', time.name = 'time', dat.name = dat.type,
  comp.name = 'xCH4',
  # Additional arguments for manometric and gravimetric methods
  pres.resid = NULL, temp.init = NULL, pres.init = NULL,
  rh.resid.init = 1, headspace = NULL,
  vol.hs.name = 'vol.hs', headcomp = 'N2',
  absolute = TRUE, pres.amb = NULL,
  # Additional arguments for GCA method
  mol.f.name = NULL, vol.syr = NULL,
  # Calculation method and other settings
  cmethod = 'removed', imethod = 'linear', extrap = FALSE,
  addt0 = TRUE, showt0 = TRUE,
  dry = FALSE,
  # Warnings and messages
  std.message = TRUE, check = TRUE,
  # Units and standard conditions
  temp.std = getOption('temp.std', as.numeric(NA)),
  pres.std = getOption('pres.std', as.numeric(NA)),
  unit.temp = getOption('unit.temp', 'C'),
  unit.pres = getOption('unit.pres', 'atm')
)

```

Arguments

dat	a data frame with reactor identification code; time of measurement (as numeric, or POSIX); and measured biogas volume, pressure in pres.unit, or total reactor mass (see dat.type argument). See Details section for details on units. Additional columns can be present—these will be returned in the output data frame. See data.struct argument for details on how data frames are structured.
dat.type	the type of data contained in dat. Use "vol" or "volume" for biogas volume (volumetric method will be used), "mass" for reactor masses (gravimetric

	method will be used) or "pres" ("pressure") for headspace pressure (pressure of the headspace will be converted to gas volume). Default is "vol".
comp	(optional) a data frame with the columns reactor identification code; time of measurement, (as numeric, or POSIX); and methane concentration within dry biogas as a mole fraction, considering only methane and carbon dioxide (unless cmethod = "total") or a single numeric value. If omitted, cumulative biogas volume will still be calculated and returned (but no methane data will be returned). The names of these columns are specified with id.name, time.name, and comp.name. Default is NULL.
temp	the temperature at which biogas volume was measured (when dat.type = "vol"), or of biogas just prior to exiting the reactor (when dat.type = "mass"). A length-one numeric vector. Degrees Celcius by default (see unit.temp argument). Default is NULL, which suppresses correction for temperature and pressure.
pres	the absolute pressure at which biogas volume was measured (when dat.type = "vol"), or of biogas just prior to exiting the reactor (when dat.type = "mass"). A length-one numeric vector or a character vector giving the name of the column in dat with the pressure measurements. Atmospheres by default (see unit.pres argument). Default is NULL, which suppresses correction for temperature and pressure. Not used for manometric method (when dat.type = "pres").
interval	do biogas measurements (volume or pressure) represent production only from the time interval between observations (default)? interval = FALSE means measured gas volume or pressure is cumulative. Applies to volumetric (dat.type = "vol") and manometric methods (dat.type = "pres"). The gravimetric method (dat.type = "mass") is cumulative by nature and the interval argument is not used. Default is TRUE.
data.struct	the structure of input data. The default of 'long' means separate objects for volume (or pressure or mass for manometric and gravimetric methods) and composition (if available). Each data frame must have reactor identification code and time columns with names specified with id.name and time.name. The dat data frame must have volume (or pressure or mass) data in a single column with the name specified by dat.name. The comp data frame must have biogas composition in a single column with the name specified by comp.name. For the data.struct = 'longcombo' option, the composition column should be in the dat data frame, and the comp argument is not used. For the data.struct = 'wide' option, two separate data frames are needed as in 'long', but there are no reactor identification code columns. Instead, in dat, volume (or pressure or mass) data are in a separate column for each bottle, and column names are reactor identification codes. Here, dat.name should be the name of the first column with volume (or pressure or mass) data. All following columns are assumed to also have volume (or pressure or mass) data. And in comp, biogas composition data are also in a separate column for each bottle, also with reactor identification codes for column names. Here, comp.name should be the name of the first column with biogas composition data, as for dat.
id.name	name of the reactor identification code column in dat. Must be the same in all data frames used in the function. Default is "id".

time.name	name of column containing time data in dat and comp data frames. Default is "time".
dat.name	name of column containing the primary response variable (volume or mass) in dat data frame. Default is dat.type value. See dat.type argument.
comp.name	name of column containing biogas mole fraction of methane in comp data frame. Default is "xCH4". Must be normalised so $x_{CH4} + x_{CO2} = 1.0$ unless cmethod = "total".
pres.resid	headspace pressure after venting. Either a single numeric value, assumed to be the same for all observations, or the name of a column in dat that contains these values for each observation. Used in manometric method only (dat.type = "pres"). Can be absolute (default) or gauge depending on the value of absolute. Length one numeric or character vector.
temp.init	initial headspace temperature in unit.temp units. Used to correct for effect of initial reactor headspace on mass loss for the gravimetric method (dat.type = "mass"), and to determine initial gas volume in the manometric method (dat.type = "pres"). Not used for volumetric method (dat.type = "vol").
pres.init	headspace pressure at the beginning of the experiment. Used in manometric method only (dat.type = "pres"). Can be absolute (default) or gauge depending on the value of absolute. Length one numeric vector.
rh.resid.init	relative humidity of the gas in the headspace at the beginning of the experiment. Typically should be 1.0 (the default). Length one numeric vector. Used in manometric method only (dat.type = "pres").
headspace	(optional) a data frame or length-one numeric vector with reactor headspace volume(s). If a data frame is used, it should at least contain a "id" (reactor identification code) column (see "id.name") and headspace volume column (see vol.hs.name argument). Required if method = "total" for the volumetric method, for initial headspace correction for the gravimetric method (see headcomp and temp.init) and for the manometric method using pressure measurements (dat.type = "pres"). Default is NULL.
vol.hs.name	name of column containing headspace volume data in optional headspace data frame. Default is "vol.hs".
headcomp	(optional) Composition of headspace for correction of initial headspace mass for gravimetric method only. If provided, correction will be applied to first mass loss observation for each individual reactor. See argument of the same name in mass2vol . Currently, the only option is "N2" for dinitrogen gas. Use of any other value will be ignored with a warning.
absolute	is the headspace pressure measured in the manometric method (dat.name and pres.resid values/columns) absolute or gauge pressure? Applies to manometric method only (dat.type = "pres"). Default is TRUE.
pres.amb	absolute ambient pressure needed to calculate absolute pressure from gauge pressure measurements. In atmospheres by default (see unit.pres argument). Only a single value is accepted. If ambient pressure differed among measurements, it is necessary to convert pressures to absolute values and use absolute = TRUE instead.
mol.f.name	name of the column in dat that contains the post-venting (f for final) quantity of methane in the syringe. Used for the absolute GC method (dat.type = "gca")

	only (Hansen et al., 2004). If venting was not done for a particular observation, the value in this column should be NA. The pre-venting column name is given in the <code>dat.name</code> column.
<code>vol.syr</code>	volume of the syringe used in the absolute GC method (Hansen et al., 2004) (<code>dat.type = "gca"</code>) in mL. Only a single value is accepted. If other volume units are used (e.g., microL), the units used for the (headspace) argument must match, and the output volume unit will be the same.
<code>cmethod</code>	method for calculating cumulative methane production. Applies to <code>dat.type = "vol"</code> and <code>dat.type = "pres"</code> methods only. Use "removed" to base production on <code>xCH4</code> and gas volumes removed (default). Use "total" to base it on the sum of methane removed and methane remaining in the reactor headspace. For "removed", <code>xCH4</code> should be calculated based on methane and CO2 only ($xCH4 + xCO2 = 1.0$). For "total", <code>xCH4</code> should be calculated including all biogas components (CH4, CO2, N2, H2S, etc.) except water. Length one character vector.
<code>imethod</code>	method used for interpolation of <code>xCH4</code> . This is passed as the method argument to <code>interp</code> . Length one character vector. Default is "linear" for linear interpolation.
<code>extrap</code>	should <code>comp.name</code> be extrapolated? Length one logical vector. This is passed as the <code>extrap</code> argument to <code>interp</code> . Default is FALSE.
<code>addt0</code>	is the earliest time in <code>dat</code> data frame "time zero" (start time)? If not, this argument adds a row with <code>time.name = 0</code> for each reactor in order to calculate production rates for the first observation. This addition is only made when <code>dat.type = "vol"</code> or <code>dat.type = "pres"</code> and <code>time.name</code> is numeric (or integer). Length-one logical vector. Default is TRUE. To return these additional rows in the output, see <code>showt0</code> .
<code>showt0</code>	should "time zero" rows be returned in the output? Can be convenient for plotting cumulative volumes. Only applies if <code>time.name</code> is numeric (or integer). These rows may have been present in the original data (<code>dat</code>) or added by the function (see <code>addt0</code>). Default value depends on <code>dat.time.name</code> column content. If <code>time.name</code> column is numeric and contains 0 then the default value is TRUE and otherwise FALSE.
<code>dry</code>	set to TRUE is volume data are standardised to dry conditions (e.g., AMPTS II data). Only applies to volumetric data (<code>dat.type = 'vol'</code>). The default (FALSE) means biogas is assumed to be saturated with water vapor.
<code>std.message</code>	should a message with the standard conditions be displayed? Default is TRUE.
<code>check</code>	should input data be checked for unreasonable values (with warnings)? Currently only composition values are checked. Default is TRUE.
<code>temp.std</code>	standard temperature for presentation of biogas and methane results. Length one numeric vector. Default value is 0 degrees C (set in <code>stdVol</code>). Argument is passed to <code>stdVol</code> .
<code>pres.std</code>	standard pressure for presentation of biogas and methane results. Length one numeric vector. Default value is 1.0 atm (101325 Pa) (set in <code>stdVol</code>). Argument is passed to <code>stdVol</code> .
<code>unit.temp</code>	temperature units for <code>temp</code> and <code>temp.std</code> arguments. Default is "C". Argument is passed to <code>stdVol</code> .

`unit.pres` pressure units for `pres` and `pres.std` arguments.
 Default is "atm". Argument is passed to `stdVol`.

Details

Using volume, mass, pressure, or molar quantity data from `dat` and gas composition from `comp`, this function will calculate standardised biogas and methane production (if `comp` is provided) for each observation, interpolating `comp.name` (from `comp` argument) to each `time.name` in `dat` if needed, and summing these for cumulative values. Use of volumetric measurements is the default case. Alternatively, if reactor mass is given in `dat`, a gravimetric approach (Hafner et al. 2015) can be applied by setting `dat.type` to "mass". In this case, to determine total cumulative biogas production for batch reactors, an initial mass should be included for each reactor. If pressure is measured in `dat`, a manometric method can be applied by setting `dat.type` to "pres". Lastly, molar quantity of methane can be used in the absolute GC method by setting `dat.type` to "gca".

For the volumetric and absolute GC methods, all volumes (input and output) have the same units, e.g., mL, L, SCF. For the gravimetric method, output volumes will be in mL if input masses are in g (or in L if input masses are in kg, etc.). For the manometric method, inputs units are specified with `unit.pres`, and output volumes are the same as headspace volume, e.g., mL or L.

Biogas composition (`comp.name` column in `comp`) is specified as the mole fraction of methane in dry biogas, normalised so mole fractions of methane and carbon dioxide sum to unity (Richards et al. 1991). Alternatively, if `method` is set to "total", biogas composition is the mole fraction of methane in dry biogas (include all the other gases except water).

Standard values and units for temperature and pressure can be globally set using the function `options`. See `stdVol`. To suppress volume correction to a "standard" temperature and pressure, leave `temp` as NULL (the default).

If `check = TRUE`, the input values of mole fraction of methane in biogas are checked, and a warning is returned if they are outside 0, 1.

Value

a data frame with all the columns originally present in `dat`, plus these others:

<code>vBg</code>	Standardised volume of biogas production for individual event.
<code>xCH4</code>	Interpolated mole fraction of methane in biogas.
<code>vCH4</code>	Standardised volume of methane production for individual event. Only if <code>comp</code> is provided.
<code>vhsCH4</code>	Standardised volume of methane present in reactor headspace. Only if <code>method = "total"</code> is used.
<code>cvBg</code>	Standardised cumulative volume of biogas production.
<code>cvCH4</code>	Standardised cumulative volume of methane production. Only if <code>comp</code> is provided.
<code>rvBg</code>	Production rate of biogas.
<code>rvCH4</code>	Production rate of methane. Only if <code>comp</code> is provided.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

References

- Hafner, S.D., Rennuit, C., Triolo, J.M., Richards, B.K. 2015. Validation of a simple gravimetric method for measuring biogas production in laboratory experiments. *Biomass and Bioenergy* **83**, 297-301.
- Hansen, T.L., Schmidt, J.E., Angelidaki, I., Marca, E., Jansen, J. la C., Mosbak, H. and Christensen, T.H. 2004. Method for determination of methane potentials of solid organic waste. *Waste Management* **24**, 393-400
- Richards, B.K., Cummings, R.J., White, T.E., Jewell, W.J. 1991. Methods for kinetic analysis of methane fermentation in high solids biomass digesters. *Biomass and Bioenergy* 1: 65-73.

See Also

[summBg](#), [interp](#), [stdVol](#), [options](#)

Examples

```
data("vol")
data("comp")

head(vol)
head(comp)

# Calculate cumulative production and rates from vol and comp
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1, id.name = "id", time.name = "days",
  comp.name = "xCH4", dat.name = "vol")
head(cum.prod)

# Note warnings and related NAs in results
# Set extrap = TRUE to extrapolate xCH4 to earliest times

# Calculate cumulative production and rates from vol and comp
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1, id.name = "id", time.name = "days",
  comp.name = "xCH4", dat.name = "vol", extrap = TRUE)
head(cum.prod)

# In this case, we can use default values for some column names, so this call is identical
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1, time.name = "days", extrap = TRUE)

# Plot results
## Not run:
# Not run just because it is a bit slow
library(ggplot2)
qplot(x = days, y = cvCH4, data = cum.prod, xlab = "Time (d)",
  ylab = "Cumulative methane production (mL)", color = id, geom = "line")

## End(Not run)

# Omit added time zero rows
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1, time.name = "days", extrap = TRUE,
  showt0 = FALSE)
```

```

head(cum.prod)

## Not run:
# Not run just because it is a bit slow
qplot(x = days, y = cvCH4, data = cum.prod, xlab = "Time (d)",
      ylab = "Cumulative methane production (mL)", color = id, geom = "line")

## End(Not run)

# Previous is different from never adding them in the first place (rates not calculated for first
# observations here)
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1, time.name = "days", extrap = TRUE,
  addt0 = FALSE)
head(cum.prod)

## Not run:
# Not run just because it is a bit slow
qplot(x = days, y = cvCH4, data = cum.prod, xlab = "Time (d)",
      ylab = "Cumulative methane production (mL)", color = id, geom = "line")

## End(Not run)

# Can use POSIX objects for time (but cumBg cannot add t0 rows here)
class(vol$date.time)
class(comp$date.time)
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1, time.name = "date.time", extrap = TRUE)
head(cum.prod)

## Not run:
# Not run just because it is a bit slow
qplot(x = date.time, y = cvCH4, data = cum.prod, xlab = "Time (d)",
      ylab = "Cumulative methane production (mL)", color = id, geom = "line")

## End(Not run)

# Can leave out composition data, and then CH4 is not included in results
cum.prod <- cumBg(vol, temp = 20, pres = 1, time.name = "days")
head(cum.prod)

# Leave out pres or temp, and results are not standardised
cum.prod <- cumBg(vol, time.name = "days")
head(cum.prod)

# Example with input data frames with different column names
data("vol2")
data("comp2")

head(vol2)
head(comp2)

cum.prod <- cumBg(vol2, comp = comp2, temp = 20, pres = 1,
  id.name = "bottle", time.name = "days",
  dat.name = "meas.vol", comp.name = "CH4.conc")

```

```

head(cum.prod)
tail(cum.prod)

# Note warnings and related NAs in results
warnings()

# Set extrap = TRUE to avoid
cum.prod <- cumBg(vol2, comp = comp2, temp = 20, pres = 1,
  id.name = "bottle", time.name = "days",
  dat.name = "meas.vol", comp.name = "CH4.conc",
  extrap = TRUE)
head(cum.prod)

## Not run:
# Not run just because it is a bit slow
qplot(x = days, y = cvCH4, data = cum.prod, xlab = "Time (d)",
  ylab = "Cumulative methane production (mL)",
  color = bottle, geom = "line")

## End(Not run)

# Mass example
data("mass")

mass

# Need to specify data type with dat.type argument (using default
# values for id.name, dat.name, and comp.name)
cum.prod <- cumBg(mass, dat.type = "mass", comp = comp, temp = 35,
  pres = 1, time.name = "days")

cum.prod

# Drop time 0 rows
cum.prod <- cumBg(mass, dat.type = "mass", comp = comp, temp = 35,
  pres = 1, time.name = "days", showt0 = FALSE)

cum.prod

# Add initial headspace correction (alternatively, headspace could
# be a data frame with a different volume for each reactor)
cum.prod <- cumBg(mass, dat.type = "mass", comp = comp, temp = 35,
  pres = 1, time.name = "days", headspace = 300,
  headcomp = "N2", temp.init = 20, showt0 = FALSE)

cum.prod

# Pressure example
data("strawPressure")
data("strawComp")
data("strawSetup")

cum.prod <- cumBg(strawPressure, dat.type = 'pres',
  comp = strawComp, temp = 35,
  id.name = 'bottle', time.name = 'time',
  dat.name = 'pres', comp.name = 'xCH4',

```

```

pres.resid = 'pres.resid', temp.init = 20,
pres.init = 101.325,
headspace = strawSetup,
vol.hs.name = 'headspace',
extrap = TRUE,
unit.pres = 'kPa', pres.std = 101.325)

head(cum.prod)

# Absolute GC method (long format is the only option)

# Generate some data--two bottles with identical results
# Units for n1 and n2 are micromoles of CH4
biogas <- data.frame(id = rep(c('A', 'B'), each = 5),
                    time.d = rep(1:5, 2),
                    n1 = rep(c(1.1, 2.4, 3.8, 5.9, 2.3), 2),
                    n2 = rep(c(NA, NA, NA, 1.2, NA), 2))

# Bottles were vented after the measurements on day 4
biogas

# Syringe volume is 0.25 mL
# Headspace volume is 50 mL

cp <- cumBg(biogas, id.name = 'id', time.name = 'time.d',
           dat.name = 'n1', mol.f.name = 'n2', vol.syr= 0.25,
           headspace = 50, dat.type = 'gca')
cp

# Suppose the bottles had two difference headspace volumes
setup <- data.frame(id = c('A', 'B'), vol.hs = c(50, 60))

cp2 <- cumBg(biogas, id.name = 'id', time.name = 'time.d',
           dat.name = 'n1', mol.f.name = 'n2', vol.syr= 0.25,
           headspace = setup, vol.hs.name = 'vol.hs', dat.type = 'gca')
cp2

# Different data structures
# Load example data
data("s3voll")
data("s3volw")
data("s3compl")
data("s3compw")
data("s3lcombo")

# wide

# Check data structure first
s3volw

s3compw

```

```

cum.prod <- cumBg(s3volw, comp = s3compw, temp = 25, pres = 1,
  time.name = 'time.d',
  data.struct = 'wide',
  dat.name = 'D', comp.name = 'D',
  extrap = TRUE)

cum.prod

# longcombo
s3lcombo

cum.prod <- cumBg(s3lcombo, temp = 25, pres = 1,
  id.name = 'id', time.name = 'time.d',
  data.struct = 'longcombo',
  dat.name = 'vol.ml', comp.name = 'xCH4',
  extrap = TRUE)

# Compare wide and longcombo to long (default)
s3voll
s3compl

cum.prod <- cumBg(s3lcombo, comp = s3compl, temp = 25, pres = 1,
  id.name = 'id', time.name = 'time.d',
  dat.name = 'vol.ml', comp.name = 'xCH4',
  extrap = TRUE)

cum.prod

```

 interp

Interpolate Biogas Composition or Cumulative Production

Description

interp interpolates (or extrapolates) biogas composition (methane concentration) or cumulative production data to a wanted time using one of several possible methods.

Usage

```
interp(times, y, time.out, method = "linear", extrap = FALSE)
```

Arguments

times	measurement times. Numeric vector or POSIX.
y	response variable at times, e.g., mole fraction of methane in biogas, or biogas cumulative production. Numeric vector.
time.out	time or times at which interpolated values are needed. Numeric vector or POSIX.

method	method used for interpolation. Default is "linear", which uses the approx function. For spline interpolation, use "ffm" or the other options available (see spline).
extrap	should y be extrapolated? Logical. Default is FALSE. See 'Details'.

Details

interp is really a wrapper for the interpolation functions [approx](#) and [spline](#). For cumulative production, which (usually) must monotonically increase, method = "hyman" is the best choice. Extrapolation behavior depends on method. For method = "linear", extrap = TRUE simply returns the value of the closest y. See [approx](#) and [spline](#) for more information.

Value

Interpolated estimates of y at given times.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

See Also

[cumBg](#)

Examples

```
# Fake composition data
dat <- data.frame(time = c(1, 7, 14, 28),
                  xCH4 = c(0.3, 0.5, 0.61, 0.65))
interp(dat$time, dat$xCH4, time.out = 10)
interp(dat$time, dat$xCH4, time.out = 10, method = "natural")
interp(dat$time, dat$xCH4, time.out = c(10, 30))
interp(dat$time, dat$xCH4, time.out = c(10, 30), method = "natural")
interp(dat$time, dat$xCH4, time.out = c(10, 30), extrap = TRUE)

# Actual data
data(comp)
# Work with one reactor
bgc <- subset(comp, id=="2_1")

# With numeric time, interpolate to 1, 7, and 30 days
interp(bgc$days, bgc$xCH4, time.out = c(1, 7, 30))

# If extrapolation is OK
interp(bgc$days, bgc$xCH4, time.out = c(1, 7, 30), extrap = TRUE)

# Or POSIXct
interp(bgc$date.time, bgc$xCH4,
       time.out = as.POSIXct("2014-07-12 13:00:00"))

# For cumulative gas production
```



```
data(vol)
# Work with one reactor
bgv <- subset(vol, id=="2_1")
# Calculate cumulative volume
bgv <- cumBg(bgv, time.name = "days")
# The interpolate of cumulative production to 1, 7, and 30 days
interp(bgv$days, bgv$cvBg, time.out = c(1, 7, 30), method = "hyman")
```

mass

Mass Change of Batch Reactors

Description

Mass measurements for 9 batch anaerobic reactors.

Usage

```
data("mass")
```

Format

A data frame with 18 observations on the following 4 variables.

`id` identification code, a unique value for each reactor in the dataset. A factor with levels 2_1, 2_2, 2_3, 2_4, 2_5, 2_6, 2_10, 2_11, and 2_12

`date.time` date and time of mass measurement, a POSIXct object.

`days` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`mass` total reactor mass in g, a numeric vector.

`when` when was measurement taken? A character vector with two values.

Details

These data are meant to be example data for [mass2vol](#), for gravimetric determination of biogas production. Reactors were 500 mL glass serum bottles with butyl rubber septa and screw caps. Masses were measured with an electronic balance to 100 mg. Data in `vol`, `mass`, `massw`, `comp`, and `setup` are from the same reactors.

Source

Measurements by Charlotte Rennuit and Sasha Hafner.

Examples

```
data(mass)
```

mass2vol

*Calculate Biogas Production Gravimetrically***Description**

mass2vol calculates biogas volume based on measured reactor mass loss.

Usage

```
mass2vol(mass, xCH4, temp, pres,
         temp.std = getOption('temp.std', 0.0),
         pres.std = getOption('pres.std', 1.0),
         unit.temp = getOption('unit.temp', 'C'),
         unit.pres = getOption('unit.pres', 'atm'),
         value = "CH4", headspace = NULL, headcomp = 'N2',
         temp.init = NULL, std.message = TRUE)
```

Arguments

mass	reactor mass loss in g. A numeric vector.
xCH4	biogas mole fraction of methane. Must be normalised so $x_{CH4} + x_{CO2} = 1.0$. A numeric vector.
temp	the temperature of biogas as it exited the reactor. A length-one numeric vector. Degrees Celcius by default (see unit.temp argument).
pres	the pressure of biogas as it exited the reactor. A length-one numeric vector. Atmospheres (atm) by default (see unit.pres argument).
temp.std	standard temperature for presentation of biogas and methane results. Default value is 0 degrees C. Argument is passed to stdVol . Length one numeric vector.
pres.std	standard pressure for presentation of biogas and methane results. Default value is 1.0 atm. Argument is passed to stdVol . Length one numeric vector.
unit.temp	temperature units for temp and temp.std arguments. Default is "C". Argument is passed to stdVol .
unit.pres	pressure units for pres and pres.std arguments. Default is "atm". Argument is passed to stdVol .
value	what should be returned? Default is "CH4", which returns methane volume in mL. "all" returns biogas, methane, and carbon dioxide volumes.
headspace	(optional) reactor headspace volume in mL. Used to correct for effect of initial reactor headspace on mass loss. A numeric vector.
headcomp	(optional) composition of the initial reactor headspace. Only required if headspace argument is specified. Used to correct for effect of initial reactor headspace on mass loss. Currently, the only option is "N2" for dinitrogen gas. Use of any other value will be ignored with a warning.
temp.init	(optional) initial headspace temperature in unit.temp units. Used to correct for effect of initial reactor headspace on mass loss.
std.message	should a message with the standard conditions be displayed? Default is TRUE.

Details

This function uses the method described in Hafner et al. (2015) to calculate biogas production from reactor mass loss. It is essential that the only change in reactor mass is due to biogas removal! Users are advised to read the original reference before applying the method. This function is vectorized. Instead of using this function directly when working with multiple measurements on multiple reactors, use the `cumBg` function (which can call up `mass2vol`).

Standard values and units of temperature and pressure can be globally set using the function `options`.

Value

volume of methane in mL as a numeric vector (if `value = "CH4"`) or a matrix with columns for biogas, methane, and carbon dioxide volumes in mL.

<code>vBg</code>	standardised volume of biogas in mL
<code>vCH4</code>	standardised volume of methane in mL
<code>vCH4</code>	standardised volume of carbon dioxide in mL

All volumes are standardised using `pres.std` and `temp.std` arguments (see `stdVol`).

Author(s)

Sasha D. Hafner and Charlotte Rennuit

References

Hafner, S.D., Rennuit, C., Triolo, J.M., Richards, B.K. 2015. Validation of a simple gravimetric method for measuring biogas production in laboratory experiments. *Biomass and Bioenergy* **83**, 297-301.

See Also

[vol2mass](#), [stdVol](#), [cumBg](#), [options](#)

Examples

```
# Volume of methane if measured mass loss was 3.1 g
mass2vol(3.1, xCH4 = 0.65, temp = 35, pres = 1)

# More details
mass2vol(3.1, xCH4 = 0.65, temp = 35, pres = 1, value = "all")

# Vectorized
data("massw")
massw$massloss <- massw$start - massw$end
massw$vCH4 <- mass2vol(massw$massloss, xCH4 = 0.65, temp = 35,
                      pres = 1)

massw
```

massw

Mass Change of Batch Reactors

Description

Mass measurements for 9 batch anaerobic reactors in a “wide” format (separate columns for initial and final mass).

Usage

```
data("massw")
```

Format

A data frame with 9 observations on the following 3 variables.

`id` identification code, a unique value for each reactor in the dataset. A factor with levels 2_1, 2_2, 2_3, 2_4, 2_5, 2_6, 2_10, 2_11, and 2_12

`start` total reactor mass at the start of its incubation in g, a numeric vector.

`end` total reactor mass at the end of its incubation in g, a numeric vector.

Details

These data are meant to be example data for application of [mass2vol](#), for gravimetric determination of biogas production. Reactors were 500 mL glass serum bottles with butyl rubber septa and screw caps. Masses were measured with an electronic balance to 100 mg. These are the same data as in `mass`, but simply in a “wide” format. Data in `vol`, `mass`, `massw`, `comp`, and `setup` are from the same reactors.

Source

Measurements by Charlotte Rennuit and Sasha Hafner.

Examples

```
data(massw)
```

`molMass`*Calculate Molar Mass of Compound*

Description

`molMass` calculates the molar mass of any organic compound (and some salts) based on its chemical formula.

Usage

```
molMass(form)
```

Arguments

<code>form</code>	a chemical formula, as a character vector, e.g., "C6H12O6" for glucose or "CH3COOH" for acetic acid, or <code>c("C6H12O6", "CH3COOH")</code> for both at once.
-------------------	--

Details

Standard atomic weights are from CIAAW, and were rounded to three or more digits (with the exception of Li) depending on the range of the reported interval for "normal materials". In general form should follow capitalization rules for elements (i.e., first letter capitalized, second lower-case). However, if `form` contains single-letter elements only, it is possible to use lower-case letters for all elements (but for clarity this is not recommended). The function is vectorized.

Value

Numeric vector with length equal to length of `form` with molar mass (g/mol)

Author(s)

Charlotte Rennuit and Sasha D. Hafner

References

CIAAW <<http://www.ciaaw.org/atomic-weights.htm>>

See Also

[calcCOD](#)

Examples

```
molMass("C6H12O6")
```

```
molMass("CH3COOH")
```

```
molMass("CH3CH2OH")
```

```

molMass("CH4")

molMass("ch4")

molMass(c("C6H12O6", "CH3COOH", "CH3CH2OH"))

# Case-sensitive for two letter elements
molMass("NaHCO3")

# Complex formulas OK
molMass("H3C(CH2)5COOH")

molMass("(C6H12O6)0.24999 (H3COOH)0.75001")

# Database is quite complete
molMass('CdSiO3')

# Use care for, e.g., hydrates
molMass('FeSO4(H2O)7')
```

planBMP

Calculate Inoculum and Substrate Mass for BMP Experiments

Description

planBMP assists in the design of BMP experiments. It can be used to determine inoculum and substrate masses based on inoculum-to-substrate ratio and volatile solids concentrations, or to calculate inoculum-to-substrate ratio based on masses.

Usage

```

planBMP(vs.inoc, vs.sub, isr = NA, m.inoc = NA, m.sub = NA,
        m.tot = NA, m.vs.sub = vs.sub * m.sub, digits = 3, warn = TRUE,
        nice = TRUE)
```

Arguments

vs.inoc	volatile solids (VS) concentration of inoculum (g/g = g VS per g inoculum). Required. Numeric vector.
vs.sub	volatile solids (VS) concentration of substrate (g/g = g VS per g substrate). Required. Numeric vector.
isr	inoculum-to-substrate ratio, VS mass basis. Optional. Numeric vector.
m.inoc	total mass of inoculum (g). Optional. Numeric vector.
m.sub	total mass of substrate (g). Optional. Numeric vector.
m.tot	total mass of mixture (inoculum plus substrate) (g). Optional. Numeric vector.
m.vs.sub	VS mass of substrate (g). Optional. Numeric vector.

digits	number of significant digits to display in output. Default of 3. Integer vector with length 1.
warn	control whether warnings are displayed. Default of TRUE. Logical vector with length 1.
nice	control whether output is formatted to look nice and make reading easier. Default of TRUE. Only applied for non-vectorized (length 1) calls. Logical vector with length 1.

Details

BMP experiments should be designed giving consideration to the inoculum-to-substrate ratio (ISR), the substrate VS mass, and the mixture VS concentration. This function calculates inoculum and substrate masses based on VS concentrations and ISR, along with either total mixture mass or substrate VS mass. Alternatively, it can be used to calculate ISR if the masses have been selected. Warnings are based on the guidelines of Holliger et al. (2016).

Value

A named numeric vector, or (if any of the first 7 input arguments have a length > 1, i.e., a vectorized call), a data frame. Names and interpretation are identical to the first 7 input arguments, and also include:

vs.mix	VS concentration in mixture (g/g)
m.vs.tot	total VS mass in mixture (g)

For non-vectorized calls, the results is returned invisibly and a easy-to-read summary is printed (see nice argument).

Note

Calculations used in this function are trivial, and they could also be done with a spreadsheet or even pencil and paper. The advantage here is ease and some flexibility. In addition to ISR and the other parameters included in this function, expected biogas production rate and bottle headspace volume are important, depending on the method. For more details, see Holliger et al. (2016).

Author(s)

Sasha D. Hafner

References

Holliger, C., Alves, M., Andrade, D., Angelidaki, I., Astals, S., Baier, U., Bougrier, C., Buffiere, P., Carbella, M., de Wilde, V., Ebertseder, F., Fernandez, B., Ficara, E., Fotidis, I., Frigon, J.-C., Fruteau de Lacroix, H., S. M. Ghasimi, D., Hack, G., Hartel, M., Heerenklage, J., Sarvari Horvath, I., Jenicek, P., Koch, K., Krautwald, J., Lizasoain, J., Liu, J., Mosberger, L., Nistor, M., Oechsner, H., Oliveira, J.V., Paterson, M., Pauss, A., Pommier, S., Porqueddu, I., Raposo, F., Ribeiro, T., Rusch Pfund, F., Stromberg, S., Torrijos, M., van Eekert, M., van Lier, J., Wedwitschka, H., Wierinck, I., 2016. Towards a standardization of biomethane potential tests. *Water Science and Technology* **74**, 2515-2522.

See Also

[cumBg](#), [summBg](#),

Examples

```
# Bottles are 500 mL, substrate is wastewater sludge.
# Assume we want no more than 250 mL reacting volume (~250 g)
# First try setting ISR and total mass.
# VS concentrations: 0.02 g/g in inoculum, 0.07 g/g for substrate, ISR = 2.
planBMP(vs.inoc = 0.02, vs.sub = 0.07, isr = 2, m.tot = 250)
# Get 31 g substrate, 220 g inoculum.

# After setup, we can check final values.
planBMP(vs.inoc = 0.018, vs.sub = 0.072, m.sub = 32, m.inoc = 218)
# We didn't quite meet our target in this case--next time use more inoculum to be sure

# We can alternatively specify substrate VS mass
planBMP(vs.inoc = 0.02, vs.sub = 0.07, isr = 2, m.vs.sub = 2)

# Some options
planBMP(vs.inoc = 0.02, vs.sub = 0.07, isr = 2, m.vs.sub = 2, nice = FALSE)
# Perhaps we want to use three different ISRs
planBMP(vs.inoc = 0.02, vs.sub = 0.07, isr = 2:4, m.vs.sub = 2, nice = FALSE)
```

predBg

Predict Biogas Production

Description

predBg predicts biogas and methane production based on composition and other, optional, details.

Usage

```
predBg(form = NULL, mass = 1, mol = NULL, fs = 0, fd = 1,
        mcomp = NULL, COD = NULL, conc.sub = NULL, pH = NULL,
        temp = NULL, mu = 0.1, shortform = NULL, value = "CH4")
```

Arguments

form	(optional) a chemical formula for the substrate, as a character vector of length one, e.g., "C6H12O6" for glucose or "CH3COOH" for acetic acid.
mass	mass of substrate present, in g. A numeric vector.
mol	(optional) moles of substrate present. A numeric vector.
fs	fraction of substrate used for microbial biomass production ($0 \leq fs \leq 1$). See "Details". A numeric vector.

fd	fraction of substrate that is degradable ($0 \leq \text{fd} \leq 1$). See "Details". A numeric vector.
mcomp	(optional) "macromolecule"-based composition of the substrate. A named numeric vector with relative masses of macromolecular groups or any chemical. Options for macromolecular groups include: VFA (volatile fatty acids, VFAs), protein, carbohydrate, lipid, and lignin. An empirical form will be calculated from fixed chemical formulas. To see the available options, use <code>biogas::std.forms</code> .
COD	(optional) chemical oxygen demand (COD) of the substrate (g oxygen). If provided, mass will be ignored. A numeric vector.
conc.sub	(optional) concentration of the substrate relative to water, as g substrate per kg water. Used only for carbon dioxide partitioning. A numeric vector.
pH	(optional) pH of the solution. Used only for carbon dioxide partitioning. A numeric vector.
temp	(optional) temperature of the system in degrees C. Used only for carbon dioxide partitioning. A numeric vector.
mu	(optional) ionic strength of the solution. Used only for carbon dioxide partitioning. A numeric vector.
shortform	should formula from form or mcomp arguments be shortened? Default is FALSE if form is used, TRUE if mcomp is used.
value	what should be returned? Four options are currently available. "CH4" (default) returns standardised methane volume only. "all" returns all available additional results. "reactionn" returns a numeric reaction (names are chemical species). "reactionc" returns a text reaction. Length-one character vector.

Details

predBg is a flexible function that returns different details depending on the data provided. Calculations can be based on form, COD, or mcomp (at least one of these is required). See 'Examples' for more information.

In its simplest usage, predBg calculates theoretical biochemical methane potential (theoretical BMP).

With more parameters, it can also predict carbon dioxide partitioning, total biogas production and composition, as well as microbial biomass production and nitrogen requirement. Stoichiometry is based on Eq. 13.5 in Rittmann and McCarty (2001). Partitioning of carbon dioxide is based on an equilibrium speciation model using temperature-dependent parameters (Henry's law constant and dissociation constants) based on Hafner et al. (2012). predBg is vectorized for all arguments except mcomp, and will recycle argument elements as needed.

Value

Standardised volume (at 0 degrees C and 1 atmosphere) of methane produced in mL (for value = "CH4"), or a data frame with some of these columns (depending on provided arguments):

form	empirical chemical formula of substrate, typically from the input argument
mass	substrate mass in g, typically from the input argument
mol.mass	substrate molar mass in g/mol

moles	moles of substrate
COD	total calculated oxygen demand (COD') based on form (or echoed chemical oxygen demand (COD)) of substrate in g oxygen
fs	fs argument echoed
fe	Rittmann and McCarty's fe ($f_e = 1 - f_s$)
fd	fd argument echoed
conc. sub	conc. sub argument echoed
temp	temp argument echoed
pH	pH argument echoed
hydro	hydrolytic water consumption (g water)
fCH4	moles methane produced divided by the sum of methane and carbon dioxide moles. Equal to xCH4 in biogas if carbon dioxide (and related aqueous species) in solution are negligible
xCH4	mole fraction of methane in dry biogas
vCH4	standardised volume (dry, 0 degrees C, 1.0 atm) of methane produced in mL
mCH4	mass of methane produced in g
mCO2	mass of carbon dioxide produced in g (including both biogas and inorganic carbon in solution)
mCO2Bg	mass of carbon dioxide in biogas in g
mCO2.sol	mass of inorganic carbon in solution in g
cTIC	concentration of inorganic carbon in solution in mol/kg (per kg water)
m.bio	mass of microbial biomass produced, VS only, in g
N.req	nitrogen required for production of required microbial biomass in g of N. Negative value indicates mineralization

Note

Predictions will only be as good as the parameter values provided, and maybe not even that good. f_s should be interpreted as $f_{s'}^*$ in Rittmann and McCarty (2001), i.e., the effective value after decay of microbial biomass, and not f_s^0 . The original reference (Section 2.3 and Eq. (3.33) in Rittmann and McCarty (2001)) and the predBg vignette provide more details. Partitioning of carbon dioxide is based on equilibrium between all biogas produced and the solution, and represents a continuous reactor running at steady-state.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

References

- Hafner, S.D. 2007 *Ammonia Speciation in Anaerobic Digesters*. PhD dissertation, Cornell University.
- Hafner, S.D., Montes, F., Rotz, C.A. 2012 The role of carbon dioxide in emission of ammonia from manure. *Atmospheric Environment* **66**, 63-71.

Moller, H.B., Sommer, S.G., Ahring, B.K. 2004 Methane productivity of manure, straw and solid fractions of manure. *Biomass and Bioenergy* **26**, 485-495.

Rittmann, B., McCarty, P. 2001 *Environmental Biotechnology*. McGraw-Hill.

Triolo, J.M., Sommer, S.G., Moller, H.B., Weisbjerg, M.R., Jiang, X.Y. 2011 A new algorithm to characterize biodegradability of biomass during anaerobic digestion: Influence of lignin concentration on methane production potential. *Bioresource Technology* **102**, 9395-9402.

See Also

[calcCOD](#), [molMass](#)

Examples

```
# BMP of cellulose in mL CH4/g
predBg("C6H10O5")

# How much is produced in a real reactor? Assume 10% goes to
# biomass production
predBg("C6H10O5", fs = 0.1)

# And substrate is 80% degradable
predBg("C6H10O5", fs = 0.1, fd = 0.8)

# More detailed results
predBg("C6H10O5", value = "all")

# Given a substrate with a COD of 1.4 g
predBg(COD = 1.4)

# But CH4 production is reduced if substrate is 80% degradable
# and 10% goes to cell synthesis
predBg(COD = 1.4, fd = 0.8, fs = 0.1)

# Can use macromolecule composition, e.g., for pig manure
predBg(mcomp = c(vfa = 0.2, protein = 0.25, carbohydrate = 0.32,
                lipid = 0.15, lignin = 0.08),
       value = "all")

# Function expects sum(mcomp) == 1 but this is not required
# But unless mass is set, mass is taken as sum(mcomp)
predBg(mcomp = c(vfa = 20, protein = 25, carbohydrate = 32,
                lipid = 15, lignin = 8),
       value = "all")

# Specify mass if this is not the case
predBg(mcomp = c(vfa = 20, protein = 25, carbohydrate = 32,
                lipid = 15, lignin = 8),
       mass = 1, value = "all")

# Can mix formulas and macromolecular groups in mcomp
predBg(mcomp = c(C6H12O6 = 0.5, protein = 0.5))
```

```

# For CO2 partitioning, must provide conc.sub, pH, and temp
# cattle manure example from Hafner (2007)
predBg("C13H20O11N", mass = 1, fs = 0.1, fd = 0.56,
       conc.sub = 85, pH = 7.7, temp = 35, value = "all")

# Or, mix of waste paper and waste vegetable oil from vignette
predBg(mcomp = c(C6H10O5 = 5/6, C54H100O7 = 1/6), mass = 1,
       fd = 0.8, fs = 0.1, conc.sub = 50, pH = 7.5, temp = 35,
       value = "all")

# Note that form can also be used for mixtures, but here it is
# molar ratios that are specified
predBg('(C6H10O5)5(C54H100O7)1)', mass = 1,
       fd = 0.8, fs = 0.1, conc.sub = 50, pH = 7.5, temp = 35,
       value = "all")

# Function is vectorized for everything except mcomp, e.g.,
# fs
predBg("C6H10O5", fs = c(0, 0.05, 0.1), value = "all")

# form
predBg(c("C6H10O5", "C10H19O3N", "CH3CH2OH"), value = "all")

# pH
predBg("C13H20O11N", conc.sub = 85, fs = 0.1, fd = 0.56,
       pH = 7+0:10/10, temp = 35, value = "all")

```

s3compl

Methane Content of Biogas from Batch Reactors

Description

Methane content (biogas composition) measurements from three batch bottles, organized in a “long” format.

Usage

```
data("s3compl")
```

Format

A data frame with 14 observations on the following 3 variables:

`id` identification code, a unique value for each bottle in the dataset.

`time.d` elapsed time of mass measurements (from bottle setup) in days, a numeric vector.

`xCH4` biogas methane content as a mole fraction, excluding water and all other gases other than carbon dioxide, a numeric vector

Details

These data are meant to be example data for `cumBg`. Reactors were 300 mL glass serum bottles with butyl rubber septa and screw caps. Substrate was municipal wastewater sludge. Methane and carbon dioxide contents were determined by gas chromatography using a thermal conductivity detector and normalised so methane and carbon dioxide sum to 1.0. Data in `s3vol1`, `s3comp1`, `s3volw`, `s3compw`, and `s3lcombo` are from the same bottles.

Source

Sasha D. Hafner. Measurements by Johanna Maria Pedersen.

Examples

```
data(s3comp1)
```

s3compw

Methane Content of Biogas from Batch Reactors

Description

Methane content (biogas composition) measurements from three batch bottles, organized in a “wide” format.

Usage

```
data("vol")
```

Format

A data frame with 5 observations on the following 4 variables:

`time.d` elapsed time of mass measurements (from bottle setup) in days, a numeric vector.

`D` biogas methane content at `time` from bottle “D” as mole fraction, a numeric vector.

`E` biogas methane content at `time` from bottle “E” as mole fraction, a numeric vector.

`F` biogas methane content at `time` from bottle “F” as mole fraction, a numeric vector.

Details

These data are meant to be example data for `cumBg`. Reactors were 300 mL glass serum bottles with butyl rubber septa and screw caps. Substrate was municipal wastewater sludge. Methane and carbon dioxide contents were determined by gas chromatography using a thermal conductivity detector and normalised so methane and carbon dioxide sum to 1.0. Data in `s3vol1`, `s3comp1`, `s3volw`, `s3compw`, and `s3lcombo` are from the same bottles.

Source

Sasha D. Hafner. Measurements by Johanna Maria Pedersen.

Examples

```
data(s3compw)
```

s3lcombo

Biogas Volume and Methane Content from Batch Bottles

Description

Biogas volume and methane content measurements from three batch bottles, organized in a “long” format.

Usage

```
data("s3lcombo")
```

Format

A data frame with 21 observations on the following 4 variables:

`id` identification code, a unique value for each bottle in the dataset.

`time.d` elapsed time of mass measurements (from bottle setup) in days, a numeric vector.

`vol.ml` volume of biogas removed at `time` in mL, a numeric vector.

`xCH4` biogas methane content as a mole fraction, excluding water and all other gases other than carbon dioxide, a numeric vector

Details

These data are meant to be example data for `cumBg`. Reactors were 300 mL glass serum bottles with butyl rubber septa and screw caps. Substrate was municipal wastewater sludge. Volume was measured using syringes. Methane and carbon dioxide contents were determined by gas chromatography using a thermal conductivity detector and normalised so methane and carbon dioxide sum to 1.0. Data in `s3vol1`, `s3compl`, `s3volw`, `s3compw`, and `s3lcombo` are from the same bottles.

Source

Sasha D. Hafner. Measurements by Johanna Maria Pedersen.

Examples

```
data(s3lcombo)
```

`s3voll`*Biogas Volume from Batch Bottles*

Description

Biogas volume measurements from three batch bottles, organized in a “long” format.

Usage

```
data("s3voll")
```

Format

A data frame with 21 observations on the following 4 variables:

`id` identification code, a unique value for each bottle in the dataset.

`time.d` elapsed time of mass measurements (from bottle setup) in days, a numeric vector.

`vol.ml` volume of biogas removed at `time` in mL, a numeric vector.

`cvol.ml` cumulative sum of `vol`.

Details

These data are meant to be example data for `cumBg`. Reactors were 300 mL glass serum bottles with butyl rubber septa and screw caps. Substrate was municipal wastewater sludge. Volume was measured using syringes. Data in `s3voll`, `s3compl`, `s3volw`, `s3compw`, and `s3lcombo` are from the same bottles.

Source

Sasha D. Hafner. Measurements by Johanna Maria Pedersen.

Examples

```
data(s3voll)
```

`s3volw`*Biogas Volume from Batch Reactors*

Description

Biogas volume measurements from three batch bottles, organized in a “wide” format.

Usage

```
data("vol")
```

Format

A data frame with 7 observations on the following 5 variables:

`time.d` elapsed time of mass measurements (from bottle setup) in days, a numeric vector.

`D` volume of biogas removed at `time` from bottle “D” in mL, a numeric vector.

`E` volume of biogas removed at `time` from bottle “E” in mL, a numeric vector.

`F` volume of biogas removed at `time` from bottle “F” in mL, a numeric vector.

Details

These data are meant to be example data for `cumBg`. Reactors were 300 mL glass serum bottles with butyl rubber septa and screw caps. Volume was measured using syringes. Data in `s3vol1`, `s3comp1`, `s3volw`, `s3compw`, and `s3lcombo` are from the same bottles.

Source

Sasha D. Hafner. Measurements by Johanna Maria Pedersen.

Examples

```
data(vol)
```

```
setup
```

Setup Details for Batch Reactors

Description

Description of reactor substrates and information on reactor, substrates and, inoculum initial masses for twelve anaerobic reactors.

Usage

```
data("setup")
```

Format

A data frame with 12 observations on the following 11 variables:

`id` identification code, a unique value for each reactor in the dataset. A factor with levels 2_1, 2_2, 2_3, 2_4, 2_5, 2_6, 2_7, 2_8, 2_9, 2_10, 2_11, and 2_12

`descrip` description of substrate, a factor with levels A B inoc.

`msub` mass of substrate added to the reactor at the start in g, a numeric vector.

`minoc` mass of inoculum added to the reactor at the start in g, a numeric vector.

`mvs.sub` mass of volatile solids (VS) of substrate added to the reactor at the start in g, a numeric vector.

`mvs.inoc` mass of volatile solids (VS) of inoculum added to the reactor at the start in g, a numeric vector.

`mcod.sub` mass of chemical oxygen demand (COD) of substrate added to the reactor at the start in g oxygen, a numeric vector.

`mcod.inoc` mass of chemical oxygen demand (COD) of inoculum added to the reactor at the start in g oxygen, a numeric vector.

`m.tot` total mass added to the reactor at the start in g, a numeric vector.

`mvs.tot` total mass of VS in the reactor at the start in g, a numeric vector.

`mcod.tot` total mass of COD in the reactor at the start in g oxygen, a numeric vector.

Details

These data are meant to be example data for [summBg](#). Reactors were 500 mL glass serum bottles with butyl rubber septa and screw caps. Data in `vol`, `mass`, `massw`, `comp`, and `setup` are from the same reactors.

Source

Measurements made by Charlotte Rennuit and Ali Heidarzadeh Vazifekhoran

Examples

```
data(setup)
```

setup2

Setup Details for Some Batch Reactors

Description

Information on reactor substrates and substrate and inoculum masses for 15 anaerobic reactors.

Usage

```
data("setup2")
```

Format

A data frame with 15 observations on the following 4 variables:

`bottle` identification code, a unique value for each reactor in the dataset. A factor with levels 1_1, 1_2, 1_3, 2_1, 2_2, 2_2, ... through 5_3. The first number indicates the sample, the second the replicate.

`description` description of substrate, a factor.

`sub.vs` mass of substrate volatile solids (VS) added to the reactor at the start in g, a numeric vector

`inoc.mass` mass of inoculum added to the reactor at the start in g, a numeric vector.

Details

These data are meant to be example data for [summBg](#). Reactors were 500 mL or 1000 mL glass serum bottles with butyl rubber septa and screw caps. Data in vol2, xCH42, and setup2 are from the same reactors.

Source

Measurements made by Ali Heidarzadeh Vazifehkhoran

Examples

```
data(setup2)
```

stdVol	<i>Correct Gas Volume to 'Standard' Conditions</i>
--------	--

Description

stdVol corrects gas volumes to dry conditions at a specified temperature and pressure.

Usage

```
stdVol(vol, temp, pres, rh = 1,
       temp.std = getOption('temp.std', as.numeric(NA)),
       pres.std = getOption('pres.std', as.numeric(NA)),
       unit.temp = getOption('unit.temp', 'C'),
       unit.pres = getOption('unit.pres', 'atm'),
       std.message = TRUE, warn = TRUE)
```

Arguments

vol	measured gas volume at temp temperature, pres pressure, and rh relative humidity. A numeric vector.
temp	temperature of gas in degrees C by default (see unit.temp). A numeric vector.
pres	pressure of gas in atm by default (see unit.pres). A numeric vector.
rh	relative humidity of the gas ($0 \leq rh \leq 1$). Typically should be 1.0 (the default). A numeric vector.
temp.std	"standard" temperature. Default value is 0 degrees C.
pres.std	"standard" pressure. Default value is 1.0 atm.
unit.pres	pressure units for pres and pres.std arguments. Options are "atm" (the default), "Pa", "kPa", "hPa", and "bar".
unit.temp	temperature units for temp and temp.std arguments. Options are "C" (degrees Celcius), "F" (degrees Fahrenheit), and "K" (Kelvin).
std.message	should a message with the standard conditions be displayed? Default is TRUE.
warn	if TRUE, will return a warning if temperature is below 0 C or above 100 C, or if pressure is below 50 kPa or above 150 kPa.

Details

Standardisation is done in a two-step process following Hafner et al. (2015). First, the contribution of water vapor is removed and volume is corrected to the standard pressure:

$$vd = vm*(pm - rh*pw)/ps$$

where vd = dry volume at standard pressure ps , vm = measured volume at pressure pm , pw = saturation vapor pressure of water (all pressures in Pa), and rh = relative humidity (rh above). Water vapor pressure pw is calculated from the Magnus form equation given in Alduchov and Eskridge (1996) (Eqs. (21) and (22)). In the second step, the volume is adjusted for temperature.

$$vs = vd*Ts/T$$

where vs = standardised volume and Ts = standardisation temperature (K, converted from `temp.std` argument). This approach is based on Charles's and Boyle's laws. Comparison with calculations using the Peng-Robinson equation of state suggests that error in `stdVol` is around 0.1% for typical biogas with volume measured at 25 degrees C, and higher at higher temperatures (up to 0.3% at 55 degrees C).

Standard temperature and pressure and their units can be defined by the user using the `temp.std`, `pres.std`, `temp.unit`, and `pres.unit` arguments. Alternatively, standard values and units of temperature and pressure can be globally set using the function `options`. Default values are 0 degrees C and 1.0 atm. `stdVol` is vectorized, and if one argument has a shorter length than the others, it will be recycled.

Value

Standardised gas volume in the same units as `vol`. A numeric vector.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

References

Hafner, S.D., Rennuit, C., Triolo, J.M., Richards, B.K. 2015. Validation of a simple gravimetric method for measuring biogas production in laboratory experiments. *Biomass and Bioenergy* **83**, 297-301.

Richards, B.K., Cummings, R.J., White, T.E., Jewell, W.J. 1991. Methods for kinetic analysis of methane fermentation in high solids biomass digesters. *Biomass and Bioenergy* **1**, 65-73.

See Also

[cumBg](#), [summBg](#), [stdVol](#), [options](#)

Examples

```
# 100 mL, measured at 35 C
stdVol(100, temp = 35, pres = 1)
```

```
# Or, with different units
stdVol(100, temp = 35, pres = 103, unit.pres = "kPa", pres.std = 101.325)
```

```

# Vectorized
data(vol)
head(vol)
vol$vol.std <- stdVol(vol$vol, temp = 20, pres = 1.02)
head(vol)

#using options() (recommended!)
oldoptions <- options(temp.std = 273.15, pres.std = 101325,
                      unit.temp = 'K', unit.pres = 'Pa')
vol$vol.std <- stdVol(vol$vol, temp = 293.15, pres = 101325)
head(vol)
options(oldoptions)

```

strawComp

Methane Content of Biogas

Description

Methane content (biogas composition) measurements from 12 anaerobic batch reactors with straw for substrate.

Usage

```
data("comp")
```

Format

A data frame with 63 observations on the following 4 variables.

bottle identification code, a unique value for each reactor in the dataset. Integer with all values from 1 to 12.

date.time date and time of mass measurement, a POSIXct object.

time elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

xCH4 biogas methane content as a mole fraction, excluding water and all other gases other than carbon dioxide, a numeric vector

Details

These data are meant to be example data for [cumBg](#). Reactors were ca. 600 mL glass serum bottles with butyl rubber septa and screw caps. Pressure was measured using an electronic manometer. Data in `strawMass`, `strawSetup`, and `strawPressure` are from the same reactors.

Source

Measurements by Charlotte Rennuit.

Examples

```
data(strawComp)
```

strawMass	<i>Mass Change of Batch Reactors</i>
-----------	--------------------------------------

Description

Mass measurements for 12 batch anaerobic reactors with ground straw as a substrate.

Usage

```
data("mass")
```

Format

A data frame with 89 observations on the following 4 variables.

`bottle` identification code, a unique value for each reactor in the dataset. Integer with all values from 1 to 12.

`date.time` date and time of mass measurement, a POSIXct object.

`time` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`mass` total reactor mass in g, a numeric vector.

Details

These data are meant to be example data for `mass2vol` or `cumBg`, for gravimetric determination of biogas production. Reactors were ca. 600 mL glass serum bottles with butyl rubber septa and screw caps. Masses were measured with an electronic balance to 10 mg. Data in `strawPressure`, `strawSetup`, and `strawComp` are from the same reactors.

Source

Measurements by Charlotte Rennuit.

Examples

```
data(strawMass)
```

strawPressure	<i>Headspace Pressure in Batch Reactors</i>
---------------	---

Description

Interval-based measurements of headspace pressure in 12 anaerobic batch reactors with straw as the substrate.

Usage

```
data("strawPressure")
```

Format

A data frame with 72 observations on the following 5 variables:

`bottle` identification code, a unique value for each reactor in the dataset. Integer with all values from 1 to 12.

`date.time` date and time of mass measurement, a POSIXct object.

`time` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`pres` absolute bottle headspace pressure at the stated time, in kPa. Biogas accumulated in bottles from the previous time.

`pres.resid` absolute bottle headspace pressure after venting, in kPa.

Details

These data are meant to be example data for `cumBg`. Reactors were ca. 600 mL glass serum bottles with butyl rubber septa and screw caps. Pressure was measured using an electronic manometer. Data in `strawMass`, `strawSetup`, and `strawComp` are from the same reactors.

Source

Measurements by Charlotte Rennuit.

Examples

```
data(vol)
```

`strawSetup`*Setup Details for Batch Reactors*

Description

Description of reactor substrates and information on reactor, substrates and, inoculum initial masses for 12 anaerobic reactors with straw for substrate.

Usage

```
data("setup")
```

Format

A data frame with 12 observations on the following 6 variables:

`bottle` identification code, a unique value for each reactor in the dataset. Integer with all values from 1 to 12.

`treatment` treatment of the substrate (ground straw), a factor.

`start` starting date and time, when the reactors were set up.

`sub.mass` mass of substrate added to the reactor at the start in g, a numeric vector.

`inoc.mass` mass of inoculum added to the reactor at the start in g, a numeric vector.

`headspace` bottle headspace volume, in mL. A numeric vector.

Details

These data are meant to be example data for `summBg`. Reactors were ca. 600 mL glass serum bottles with butyl rubber septa and screw caps. Masses were measured with an electronic balance to 10 mg. Data in `strawPressure`, `strawMass`, and `strawComp` are from the same reactors.

Source

Measurements by Charlotte Rennuit.

Examples

```
data(setup)
```

 summBg

Summarise and Normalise Cumulative Methane Production

Description

From cumulative gas production, use `summBg` to standardise, interpolate, subtract inoculum contribution, normalise by substrate mass, and summarise the output calculating mean and standard deviations for each type of sample (groups of replicates). The function is flexible: some, all, or none of these operations can be carried out in a call. Typically `summBg` is used to calculate biochemical methane potential (BMP) from cumulative methane production.

Usage

```
summBg(vol, setup, id.name = "id", time.name = "time",
       descrip.name = "descrip", inoc.name = NULL, inoc.m.name = NULL,
       norm.name = NULL, norm.se.name = NULL, vol.name = "cvCH4",
       imethod = "linear", extrap = FALSE, when = 30,
       show.obs = FALSE, show.rates = FALSE, show.more = FALSE,
       sort = TRUE, quiet = FALSE)
```

Arguments

<code>vol</code>	a data frame with the columns bottle identification code; time of measurement (as numeric, or POSIX); and the response variable biogas volume or cumulative volume. The names of these columns can be specified with <code>id.name</code> , <code>time.name</code> , and <code>vol.name</code> . Or, the default names can be used. Additional columns can be present– these will be returned in the output data frame.
<code>setup</code>	a data frame containing information to summarise, subtract inoculum effect or normalise the data. Should contain at least the column <code>id.name</code> (bottle identification code), and the additional columns depending on the operations desired. See ‘Details’ for more information.
<code>id.name</code>	name of the bottle identification code column in <code>vol</code> and <code>setup</code> . Should be the same in all data frames used in the function. Default is <code>"id"</code> .
<code>time.name</code>	name of column containing time data in <code>vol</code> . Default is <code>"time"</code> .
<code>descrip.name</code>	(optional) name of column containing a description of bottle substrate (or a code for this) in <code>setup</code> data frame. The summary of the data will be made following this column. Default is <code>"descrip"</code> .
<code>inoc.name</code>	(optional) the value in the <code>descrip.name</code> (<code>setup</code> data frame) column that is used to describe the inoculum-only bottles. Length one character vector. Argument not used by default.
<code>norm.name</code>	(optional) the name of the column in <code>setup</code> that has the mass to perform the normalisation (typically substrate VS mass). Length one character vector. Argument not used by default.

norm.se.name	(optional) the name of the column in setup that has the standard error of the data in the column norm.name. Length one character vector. If provided, this error in substrate mass will be included in the standard error and standard deviation given in the output. Argument not used by default.
inoc.m.name	(optional) the name of the column in setup that has inoculum mass. If specified, these data are used to subtract the inoculum contribution to methane production. Length one character vector. Default is "inoc".
vol.name	the name of the column in vol that contains the response variable used for the summary—typically cumulative methane volume. The default is "cvCH4" for cumulative volume of methane (CH4).
imethod	the interpolation method to be used. This is passed as the method argument to interp . Length one character vector. Default is "linear" for linear interpolation.
extrap	should extrapolation be carried out? Set to TRUE if extrapolation wanted. Length one logical vector. This is passed as the extrap argument to interp . Default is FALSE
when	value(s) of time.name in vol data frame at which the results should be evaluated. Alternatively, set to "end" (to return the latest values for each bottle) or "meas" (to return a value for each measured time for each bottle). To select a time for each bottle based on the methane production rate, use, e.g., "1p3d" for 1% per day for at least 3 d (any values can be used, even something impractical like "0.1p10d"). In this case, the earliest time where production rate drops below 1% of cumulative production per day for at least 3 days will be used. Where this time differs for multiple bottles with the same descrip.name, the latest one will be used. Numeric or character vector.
show.obs	set to TRUE to return all observations, otherwise means and standard deviations are returned. Default is FALSE.
show.rates	set to TRUE to return all observations with relative rates. Used to check rates for evaluation time. The when argument will be ignored if show.rates = TRUE. Default is FALSE.
show.more	set to TRUE to return additional details, including fraction of total methane produced by inoculum. Default is FALSE.
sort	controls sorting of results, which is by descrip.name and then time.name by default. Set to FALSE to use order of descrip.name from the setup data frame. Default is TRUE.
quiet	use to suppress messages. Default is FALSE.

Details

summBg was primarily designed to calculate the biochemical methane potential (BMP) from cumulative methane production of a set of batch bottles through these steps: interpolation of cumulative production to a specified time (if needed), subtraction of apparent inoculum contribution, normalisation of the results by substrate mass (typically volatile solids (VS) mass, but could be the mass of anything within the bottle) and calculation of mean and standard deviation for each sample type (set of replicates, identified by descrip.name).

If needed `summBg` can return values for all observations and be used for simpler operations e.g., determining cumulative biogas production at some specified time or normalising gas volume by different substrate characteristics.

To summarise data, the setup data frame should have a column with a description of bottle substrate (or a code for this). The name of the column is set by the `descrip.name` argument. If the inoculum effect is to be subtracted out, a column named `descrip.name` (for identifying replicates) and a column with the mass of inoculum present (any units), with a name set by `inoc.m.name` are both required in setup. To normalise by substrate mass (or any mass to be used for normalisation), an additional column with the mass of substrate is needed in setup –its name is set by `norm.name`.

This function is probably easier to understand by example. See ‘Examples’.

Value

a data frame, with the columns:

<code>descrip</code>	from the input data frames
<code>mean</code>	mean of the response variable <code>vol.name</code>
<code>sd</code>	standard deviation of the response variable <code>vol.name</code>
<code>se</code>	standard error of the response variable <code>vol.name</code>
<code>n</code>	number of bottles

Note

Reported standard deviation and standard error includes an estimate of variability from subtracting the inoculum contribution when this is done.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

See Also

[cumBg](#)

Examples

```
data("vol")
data("comp")
data("setup")

# First need to calculate cumulative methane production data
cum.prod <- cumBg(vol, comp = comp, temp = 20, pres = 1,
                 time.name = "days", extrap = TRUE)

head(cum.prod)

# Cumulative methane production (default) at 30 d
# Uses default names for some columns
summBg(vol = cum.prod, setup = setup, time.name = "days", when = 30)
```

```

# Or total cumulative biogas
summBg(vol = cum.prod, setup = setup, time.name = "days",
       vol.name = "cvBg", when = 30)

# Cumulative CH4 only, subtract inoculum contribution
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = 30)

# And normalise by mvs.sub column (mass of substrate VS here) (so the result is BMP)
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = 30,
       norm.name = "mvs.sub")

# Same example, but return results for three times
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = c(10, 30, 60),
       norm.name = "mvs.sub")

# Back to earlier example, but return all individual observations
# (and total production and individual contributions of substrate
# and inoculum)
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = 30,
       norm.name = "mvs.sub", show.obs = TRUE)

# Something different: interpolated biogas production rates
summBg(vol = cum.prod, setup = setup, time.name = "days",
       vol.name = "rvBg", when = 30, show.obs = TRUE)

# The when argument could also be 'meas', 'end', or '1p3d' (or related) for any of these examples
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = "end",
       norm.name = "mvs.sub")

summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = "meas",
       norm.name = "mvs.sub")

summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = "1p3d",
       norm.name = "mvs.sub")

# If you want to apply the 1% criterion but also want a single fixed time for all bottles,
# you have to use two calls
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = "1p3d",
       norm.name = "mvs.sub")

# From the first call, the longest time is 42 days, so use when = 42
summBg(vol = cum.prod, setup = setup, time.name = "days",
       inoc.name = "inoc", inoc.m.name = "minoc", when = 42,
       norm.name = "mvs.sub")

```

```

# If an error is thrown because a bottle doesn't meet the 1% criterion, use
# show.rates = TRUE to see rates
## Not run:
# Will return error
cpshort <- cum.prod[cum.prod$days < 10, ]
summBg(vol = cpshort, setup = setup, time.name = "days",
        inoc.name = "inoc", inoc.m.name = "minoc", when = "1p",
        norm.name = "mvs.sub")

## End(Not run)

# So then use this to see which bottles are causing problems
cpshort <- cum.prod[cum.prod$days < 10, ]
summBg(vol = cpshort, setup = setup, time.name = "days",
        inoc.name = "inoc", inoc.m.name = "minoc", when = "1p",
        norm.name = "mvs.sub", show.rates = TRUE)

# Example with dataset with different column names
data("vol2")
data("comp2")
data("setup2")

# First need to calculate cumulative methane production data
cum.prod <- cumBg(vol2, comp = comp2, temp = 20, pres = 1,
                 id.name = "bottle", time.name = "days",
                 dat.name = "meas.vol", comp.name = "CH4.conc",
                 extrap = TRUE)

head(cum.prod)

# Cumulative CH4 production at 30 d, subtract inoculum contribution
# and normalise by sub.vs column (mass of substrate VS here) (look at setup2).
summBg(vol = cum.prod, setup = setup2, id.name = "bottle",
        time.name = "days", descrip.name = "description",
        inoc.name = "Inoculum", inoc.m.name = "inoc.mass",
        norm.name = "sub.vs", when = 30)

```

vol

Biogas Volume from Batch Reactors

Description

Biogas volume measurements from twelve batch reactors.

Usage

```
data("vol")
```

Format

A data frame with 288 observations on the following 4 variables:

`id` identification code, a unique value for each reactor in the dataset. A factor with levels 2_1, 2_2, 2_3, 2_4, 2_5, 2_6, 2_7, 2_8, 2_9, 2_10, 2_11, and 2_12

`date.time` date and time of mass measurement, a POSIXct object.

`days` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`vol` volume of biogas removed at `date.time` in mL, a numeric vector.

Details

These data are meant to be example data for multiple functions, e.g., `stdVol`, `cumBg`, or `summBg`. Reactors were 500 mL glass serum bottles with butyl rubber septa and screw caps. Volume was measured using syringes. Data in `vol`, `mass`, `xCH4`, and `setup` are from the same reactors.

Source

Measurements by Charlotte Rennuit and Ali Heidarzadeh Vazifekhoran.

Examples

```
data(vol)
```

vol2

Biogas Volume from Batch Reactors

Description

Biogas volume measurements from 15 batch reactors.

Usage

```
data("vol2")
```

Format

A data frame with 216 observations on the following 4 variables:

`bottle` identification code, a unique value for each reactor in the dataset. A factor with levels 1_1, 1_2, 1_3, 2_1, 2_2, 2_2, ... through 5_3. The first number indicates the sample, the second the replicate.

`days` elapsed time of mass measurements (from reactor setup) in days, a numeric vector.

`meas.vol` volume of biogas removed at `days` in mL, a numeric vector

Details

These data are meant to be example data for multiple functions, e.g., `stdVol`, `cumBg`, or `summBg`. Reactors were 500 mL or 1000 mL glass serum bottles with butyl rubber septa and screw caps. Volume was measured using syringes. Data in `vol2`, `xCH4`, and `setup2` are from the same reactors.

Source

Measurements by Ali Heidarzadeh Vazifehkhoran.

Examples

```
data(vol2)
```

vol2mass

Calculate Mass of Biogas

Description

`vol2mass` calculates the mass of biogas removed from a reactor, based on its composition, temperature, and pressure. This function is the inverse of `mass2vol`.

Usage

```
vol2mass(volBg, xCH4, temp.hs, temp.vol, pres.hs, pres.vol,
  unit.temp = getOption('unit.temp', 'C'),
  unit.pres = getOption('unit.pres', 'atm'),
  rh.hs = 1, rh.vol = 1)
```

Arguments

<code>volBg</code>	measured (not standardised) biogas volume in mL. Numeric vector.
<code>xCH4</code>	mole fraction of methane within biogas (dry, methane and carbon dioxide only). Numeric vector.
<code>temp.hs</code>	temperature of biogas just prior to removal, in the units specified in <code>unit.temp</code> (default of degrees C). Numeric vector.
<code>temp.vol</code>	temperature of biogas at the time of volume measurement, in the units specified in <code>unit.temp</code> (default of degrees C). Numeric vector.
<code>pres.hs</code>	pressure of biogas just prior to removal, in the units specified in <code>unit.pres</code> (default of atm). Identical to <code>pres</code> argument in <code>mass2vol</code> . Numeric vector.
<code>pres.vol</code>	pressure of gas at the time of measurement in atm by default (see <code>unit.pres</code>). Identical to <code>pres</code> argument of <code>stdVol</code> . Numeric vector.
<code>unit.pres</code>	pressure units. Options are "atm" (the default), "Pa", "kPa", "hPa", and "bar". Length-one character vector.
<code>unit.temp</code>	temperature units. Options are "C" (degrees Celcius, the default), "F", and "K". Length-one character vector.

rh.hs	relative humidity of the reactor headspace just prior to biogas removal. Length one numeric vector between zero and 1.0.
rh.vol	relative humidity of the biogas at the time of volume measurement. Length one numeric vector between zero and 1.0.

Details

This function is vectorized. Argument elements will be recycled as needed. Note that this function is conceptually but not numerically the inverse of `mass2vol`, because the `volBg` argument here is not standardised, and is assumed to be saturated with water vapor just prior to removal. The mass that is calculated may not be equal to the mass of the biogas at the time of volume measurement—as long as `temp.vol` is less than `temp.hs`, some of the water lost from the reactor condenses and is not present in the biogas at the time of volume measurement.

Standard values and units of temperature and pressure can be globally set using the function [options](#).

Value

biogas mass in g as a numeric vector.

Author(s)

Sasha D. Hafner and Charlotte Rennuit

References

Hafner, S.D., Rennuit, C., Triolo, J.M., Richards, B.K. In review. A gravimetric method for measuring biogas production. *Biomass and Bioenergy*.

See Also

[mass2vol](#), [options](#)

Examples

```
# Mass loss from reactor for 100 mL biogas measured at 20 degrees C
# and 1.0 atm, with headspace at 1.5 atm and 35 degrees C at the
# time of biogas exit
vol2mass(100, xCH4 = 0.65, temp.hs = 35, temp.vol = 20,
        pres.hs = 1.5, pres.vol = 1)

# If the measured volume has already been standardised to dry
# conditions at 0 C and 1 atm
vol2mass(100, xCH4 = 0.65, temp.hs = 35, temp.vol = 0,
        pres.hs = 1.5, pres.vol = 1, rh.vol = 0)

# Here vol2mass *is* numerically the inverse of mass2vol
vol2mass(mass2vol(1.234, xCH4 = 0.65, temp = 35, pres = 1.5,
                value = "Bg"),
        xCH4 = 0.65, temp.hs = 35, temp.vol = 0, pres.hs = 1.5,
        pres.vol = 1, rh.vol = 0)
```

vol2mol

*Calculate Moles of a Gas***Description**

vol2mol calculates the moles of a gas for a measured volume, temperature, and pressure.

Usage

```
vol2mol(vol, gas = "CH4", temp, pres, rh = 0,
unit.temp = getOption('unit.temp', 'C'),
unit.pres = getOption('unit.pres', 'atm'),
tp.message = TRUE)
```

Arguments

vol	measured gas volume in mL. Numeric vector.
gas	chemical formula of gas. Currently options are c("CH4", "CO2", "N2", "H2"). Character vector.
temp	temperature of gas at time of volume measurement. Numeric vector.
pres	pressure of gas at time of volume measurement, in the units specified in unit.pres (default of atm). Numeric vector.
rh	relative humidity of the gas at time of measurement. Length one numeric vector between zero and 1.0. Default is zero (dry gas).
unit.temp	temperature units. Options are "C" (degrees Celcius, the default), "F", and "K". Length-one character vector.
unit.pres	pressure units. Options are "atm" (the default), "Pa", "kPa", "hPa", and "bar". Length-one character vector.
tp.message	should a message display temp and pres to avoid unit mistakes? Length one logical vector.

Details

This function uses a simple and approximate approach for the conversion, based on Charles's and Boyle's laws, with NIST values for 0 degrees C and 1.0 atm taken as the reference state (Lemmon et al. 2011). Measured volume is normalized to 1.0 atm and 0 degrees C using `stdVol`, and the result is divided by the molar volume at the same conditions, as reported by NIST. Resulting error should be within 0.5% for temperature and pressure close to ambient, and is usually below 0.2%.

Value

gas amount in moles as a numeric vector.

Author(s)

Sasha D. Hafner

References

Lemmon EW, McLinden MO, Friend DG. Thermophysical Properties of Fluid Systems. In Linstrom PJ, Mallard WG, editors. *Chemistry WebBook*, NIST Standard Reference Database Number 69. Gaithersburg, MD: National Institute of Standards and Technology, 2011.

See Also

[vol2mass](#), [mass2vol](#), [options](#)

Examples

```
# Assume we have measured 253 mL CH4 at 1.0 atm and 22 C, dry
vol2mol(253, "CH4", temp = 22, pres = 1)
```

```
# Compare to results to NIST values for the least ideal gas
# Results are within 0.5%
# Expect 0.04108 mol
vol2mol(1000, "CO2", temp = 25, pres = 1)
```

```
# Expect 0.08258 mol
vol2mol(1000, "CO2", temp = 25, pres = 2)
```

```
# Expect 0.07482 mol
vol2mol(1000, "CO2", temp = 55, pres = 2)
```

Index

*Topic **chron**

cumBg, 6
interp, 15
summBg, 40

*Topic **datasets**

comp, 4
comp2, 5
mass, 17
massw, 20
s3compl, 28
s3compw, 29
s3lcombo, 30
s3voll, 31
s3volw, 31
setup, 32
setup2, 33
strawComp, 36
strawMass, 37
strawPressure, 38
strawSetup, 39
vol, 44
vol2, 45

*Topic **manip**

cumBg, 6
mass2vol, 18
planBMP, 22
stdVol, 34
summBg, 40
vol2mass, 46
vol2mol, 48

*Topic **smooth**

interp, 15

*Topic **univar**

summBg, 40

approx, 16

calcCOD, 2, 21, 27

comp, 4

comp2, 5

cumBg, 4, 5, 6, 16, 19, 24, 29–32, 35–38, 42,
45, 46

interp, 4, 5, 9, 11, 15, 41

mass, 17

mass2vol, 8, 17, 18, 20, 37, 46, 47, 49

massw, 20

molMass, 3, 21, 27

options, 10, 11, 19, 35, 47, 49

planBMP, 22

predBg, 3, 24

s3compl, 28

s3compw, 29

s3lcombo, 30

s3voll, 31

s3volw, 31

setup, 32

setup2, 33

spline, 16

stdVol, 9–11, 18, 19, 34, 35, 45, 46, 48

strawComp, 36

strawMass, 37

strawPressure, 38

strawSetup, 39

summBg, 4, 5, 11, 24, 33–35, 39, 40, 45, 46

vol, 44

vol2, 45

vol2mass, 19, 46, 49

vol2mol, 48