¿ library(knitr) ¿ opts$_c$*hunk*set(cache=FALSE,tidy=FALSE,highlight=FALSE) ¿ opts$_c$*hunk*set(cache = FALSE, tidy = FALSE, fig.align = "center") ¿ library(biogas) ¿ ¿ options(width=65) ¿ NTS: Delete following lines before submission! ¿ files ¡- list.files('../R', full.names = TRUE) ¿ for(i in files) source(i) ¿ files ¡- list.files('../data', full.names = TRUE) ¿ for(j in files) load(j)

# Calculating methane and biogas production and production rates using manometric methods

Nanna Løjborg and Sasha D. Hafner (sasha.hafner@eng.au.dk)

September 23, 2019

## 1 Introduction

Manometric biogas calculation methods are commonly used for calculation of cumulative biogas and methane production and production rates prior to determining methane potential and anaerobic biodegradability of a given substrate.. In manometric methods bottle headspace pressure is measured under constant temperature conditions from different techniques such as using pressure manometers or transducers, manometer assisted syringes, or low flow pressure [**?**]. The measured pressure is equivalent to a certain volume fraction and is converted to biogas and methane (if composition is provided) volume data using manometric calculation methods. Pressure manometers requires a pressure to build up inside the headspace of the bottles. Headspace pressure is typically measured with a portable manometer, which evaluates the pressure relative to atmospheric pressure. Determination of CH4 volume from volumetric measurements required some data processing. For the purpose of this, an additional function (`cumBgDataPrep()`) was created just to handle data preparation (refer to cumBgDataPrep.R). `cumBgDataPrep()` sorts and restructures biogas data from volumetric, manometric, gravimetric, and gas density measurements for further calculation of cumulative biogas and methane production, and production rates.

We developed a function to process manometric measurements only, to simplify functions within the biogas package, making it easier for new biogas package users to work with their data. This document provides a brief description of the manometric biogas calculation function (`cumBgMan()`) for new users. We have assumed that readers are familiar with biogas data collection and R.

## 2 Overview of the function

`cumBgMan()` is a "high-level" function within the biogas package. The purpose of `cumBgMan()` is to convert pressure data collected in the laboratory to cumulative biogas and $CH_4$ production and to calculate production rates. The function can handle data from any number of bottles. For simple operations

(e.g. interpolation and standardization of biogas volume) `cumBgMan()` is supported by calls to external "low-level" functions (refer to Section **??**, Table **??**). The "low-level" functions are straight-forward to use, and details can be found in their individual help files. This document describes the use of `cumBgMan()`.

In general, cumBg* functions are compiled of four sections: check arguments, restructuring and sorting data, interpolation if needed, and biogas standardization and calculations. Restructuring and sorting of data and interpolation are handled by the external functions `interp()` and `cumBgDataPrep()`, respectively. From `interp()` gas composition, cumulative biogas production, and other variables can be interpolated to a specified time if required. From `cumBgDataPrep()` "wide" and "long" data structure are restructured to "longcombo" data, which is required for `cumBgMan()` to further calculate cumulative biogas and $CH_4$ production and production rates. Additionally, data is sorted, headspace is added if provided, and composition data is corrected if it seems to be a percentage. Subsequently, the now restructured and sorted data is standardized in `cumBgVol()` by an external function called `stdVol()`.

Two manometric methods are commonly used to evaluate biogas pressure measurements. Method 1 is based on normalized CH4 concentrations, whereas method 2 accounts for the actual CH4 in the bottle headspace. Both methods are available through `cumBgMan()` and results is expected to be independent of method. The examples below describe cumulative biogas calculation on two different datasets. The "long" structured dataset is evaluated using manometric method 1, whereas the "longcombo" dataset is evaluated using manometric method 2, as true methane concentrations are provided. All external functions are within the biogas package

## 2.1 Function arguments

The arguments for the cumBgMan function are:

```
## function (dat, comp = NULL, temp = NULL,
##           interval = TRUE, data.struct = "longcombo",
##           id.name = "id", time.name = "time",
##           dat.name = "pres", comp.name = "xCH4",
##           temp.init = NULL, pres.init = NULL, pres.resid = NULL,
##           rh.resid = NULL, rh.resid.init = 1,
##           headspace = NULL, vol.hs.name = "vol.hs",
##           absolute = TRUE, pres.amb = NULL,
##           cmethod = "removed", imethod = "linear",
##           extrap = FALSE, addt0 = TRUE, showt0 = TRUE,
##           dry = FALSE, empty.name = NULL,
##           std.message = !quiet, check = TRUE,
##           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"),
```

```
##            quiet = FALSE)
## NULL
```

Most of the arguments have default values, but to calculate $CH_4$ production we must provide values for at least `dat` (data frame with pressure measurements), `comp` (data frame with gas composition), and `temp` (biogas temperature)[1]. along with the names of a few columns in the input data frames. If `temp` argument is not provided, biogas volumes will not be standardized. Similar, if `comp` argument is not provided, `cumBgMan()` will return calculations on biogas only and no calculations on $CH_4$.

By default `interval = TRUE` and `data.struct` is set as "longcombo". "wide" and "long" structured data will be restructured to "longcombo" internally by `cumBgDataPrep()`, when specified by the `data.struct` argument. When data are cumulative, the interval argument should be set to FALSE (refer to Section **??**).

Similarly, there is an `id.name` argument for the bottle identification code (ID) column (default = "id"). For `data.struct = "wide"`, there is no ID column. Instead data for each bottle, have individual columns and column names, which are used as ID codes. Here, the name of the column containing the response variables (`dat.name`), is set as the name of the first column with response variables. All following columns are assumed to also have measurement data.

Furthermore, we need to specify the name of the time column containing time data using the `time.name` argument (default = "time"). If separate data frames are used for `dat` and `comp`, the name must match. Time data may be POSIX objects, but then t0 will not be added to rows by the cumBgMan function. In addition the `addt0` argument is used to add row with "time zero" (`time.name = 0`) for each bottle in order to calculate production rates for the first observation (default = TRUE). Whereas, `showt0` determines if the "time zero" should be returned in the output (default = TRUE if time.name is numeric and contains 0 and otherwise FALSE).

The `comp.name` argument is used to indicate which column within the `comp` data frame contains $CH_4$ concentrations (as mole fraction in dry biogas, normalized so the sum of mole fractions of $CH_4$ and $CO_2$ sum to unity). Default is $xCH_4$. `comp` may also just be a single value instead of a data frame or column. When providing a single value for `comp`, this value is applied to all observationsand the `comp.name` argument is not needed. The definition of xCH4 depends on the cmethod argument. By default (`cmethod = "removed"`) the function calculates volumes following [**?**] as the product of standardized volume of biogas removed and normalized $CH_4$ content. If results should be based on the sum of methane removed and methane remaining in the bottle headspace, `cmethod` should be set to `"total"`. When `cmethod = "total"`, $CH_4$ concentration is calculated using all components ($CH_4$, $CO_2$, $N_2$, $H_2S$, etc.) instead of $CH_4$ and

[1]. By default, temperature is in °C and pressure in atm, but these can be changed in the function call with the `temp.unit` and `pres.unit` arguments, or globally with `options`. The same default values apply for temperature and pressure for presentation of biogas and methane, but these can be changed in the function call with the `temp.std` and pres.std arguments.

$CO_2$ only.

If any CH4 measurements are missing, xCH4 is interpolated by the external "low-level" function `interp()`. Here, the `imethod` argument can be used to define interpolation method (default = "linear"), which is passed to `interp()`. Similar, an `extrap` argument is passed to `interp()` (default = "FALSE"). The `extrap` argument is used to indicate if composition data (`comp.name`) should be extrapolation (e.g., in the case of missing initial composition values). Extrapolation is constant (the value of the nearest observation is taken). In general, extrapolation and extensive interpolation should be avoided.

Initial headspace temperature and pressure are required to determine initial gas volume in the bottles and can be set using the `temp.init` and `pres.init`, respectively. Default values are `NULL`. Similarly, post venting headspace pressure is required for manometric calculation methods and can be set using the `pres.resid` argument. If only a single value is provided, this will be used for all observations. Initial and post venting headspace pressure can be absolute or gauge, depending on the `absolute` argument. By default, pressure is absolute (`absolute = TRUE`). When `absolute = FALSE` pressure measurements need to be corrected using the `pres.amb` argument, representing the absolute ambient pressure. `pres.amb` can only be set as a single value and is set as 1 atm or 101.325 kPa by default.

Additionally, a data frame containing headspace volumes is required if `cmethod = "total"` (see description below) and should contain at least a headspace volume column (`vol.hs.name`) and a bottle identification column, with the same column name as in the `dat` and `comp` data frames. The headspace volume column can be set using the `vol.hs.name` argument (default = "vol.hs"). Initial relative humidity of gas in the headspace is set to 1 by default using the `rh.resid.init` argument. If values are provided for relative humidity of gas in the headspace after experiment was initiated, these can be defined by using the `rh.resid` argument.

## 2.2 Data Structures

Input data may be structured in one of three ways: "long", "wide", or "longcombo". Default is "longcombo", where the composition column is in the `dat` data frame and no composition data frame is required. All three data structure inputs are accepted, but the manometric calculation methods within `cumBgMan()` only process "longcombo" data. "wide" and "long" data are restructured internally by the low-level function `cumBgDataPrep()`. In the following examples all three data structures will be addressed. For more details about each of the three data structure see `biogas_quick_start` vignette.

# 3 Examples: Calculation of cumulative production of biogas and $CH_4$ and production rates from manometric methods

Calculation of cumulative biogas and $CH_4$ production and production rates, typically requires two data frames: Biogas volume measurements and biogas composition ($CH_4$ fraction)

## 3.1 "longcombo" data structure

In this example, we will use a "longcombo" example dataset included in the biogas package: `sludgeTwoBiogas` for both biogas volumes and composition and `sludgeTwoSetup` for grouping and headspace volumes. Substrate and inoculum masses provided from `sludgeTwoSetup` are not interesting before calculation of BMP using summBg() (refer to Section **??**).

These data are from an experiment carried out with seven different substrates all from primary wastewater sludge. The experiment included 24 batch bottles, all with inoculum and one with no substrate (blank):

- Three bottles with inoculum only (Blank50)
- Three bottles with wastewater sludge 25 (WWS25)
- Three bottles with wastewater sludge 25b (WWS25b)
- Three bottles with wastewater sludge 40 (WWS40)
- Three bottles with wastewater sludge 50 (WWS50)
- Three bottles with wastewater sludge 50b (WWS50b)
- Three bottles with wastewater sludge 60 (WWS60)
- Three bottles with wastewater sludge 75 (WWS75)

Data were originally collected by Sergi Astals at the University of Queensland. For more information, please contact Sasha D. Hafner at `sasha.hafner@eng.au.dk`.

```
> #data("sludgeTwoBiogas")
>
> dim(sludgeTwoBiogas)

[1] 324   8

> head(sludgeTwoBiogas)

  id    time.d pres mass.init mass.final   xCH4   xCO2     xCH4n
1  1 0.0000000    0  187.4168   187.4168 0.0000 0.0000 0.0000000
2  1 0.6270833  368  187.4149   187.3814 0.1010 0.1004 0.5014896
```

```
3  1 1.6027778  286  187.3823   187.3568 0.2253 0.1610 0.5832255
4  1 1.9972222  134  187.3538   187.3447 0.2485 0.1800 0.5799300
5  1 2.7152778  142  187.3406   187.3287 0.3246 0.2062 0.6115298
6  1 3.6180556  168  187.3295   187.3150 0.3674 0.2222 0.6231343
```

```
> summary(sludgeTwoBiogas)
```

```
       id           time.d            pres
 Min.   : 1.0   Min.   : 0.000   Min.   :   0.0
 1st Qu.: 8.0   1st Qu.: 2.715   1st Qu.: 181.0
 Median :12.5   Median : 8.637   Median : 358.0
 Mean   :12.0   Mean   :18.113   Mean   : 565.4
 3rd Qu.:17.0   3rd Qu.:24.029   3rd Qu.: 753.0
 Max.   :21.0   Max.   :83.662   Max.   :3785.0
   mass.init       mass.final         xCH4
 Min.   :144.7   Min.   :144.7   Min.   :0.0000
 1st Qu.:163.6   1st Qu.:163.6   1st Qu.:0.5479
 Median :183.5   Median :183.5   Median :0.6201
 Mean   :184.3   Mean   :184.2   Mean   :0.5460
 3rd Qu.:200.0   3rd Qu.:199.9   3rd Qu.:0.6418
 Max.   :228.4   Max.   :228.4   Max.   :0.7377
      xCO2            xCH4n
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.2599   1st Qu.:0.6446
 Median :0.3012   Median :0.6602
 Mean   :0.2791   Mean   :0.6220
 3rd Qu.:0.3354   3rd Qu.:0.6727
 Max.   :0.6344   Max.   :0.7637
```

```
> #data("sludgeTwoSetup")
>
> dim(sludgeTwoSetup)
```

```
[1] 18  5
```

```
> head(sludgeTwoSetup)
```

```
  id descrip  vol.hs m.inoc  m.sub.vs
1  1 Blank50  79.980  80.02 0.0000000
2  2 Blank50  79.980  80.02 0.0000000
3  3 Blank50  79.950  80.05 0.0000000
4  7   WWS75 117.575  35.02 0.3588463
5  8   WWS75 117.535  35.06 0.3588463
6  9   WWS75 117.490  35.08 0.3600578
```

```
> summary(sludgeTwoSetup)
```

```
       id          descrip             vol.hs
 Min.   : 1.00   Length:18          Min.   : 38.91
 1st Qu.: 8.25   Class :character   1st Qu.: 63.19
 Median :12.50   Mode  :character   Median : 80.56
 Mean   :12.00                      Mean   : 80.05
 3rd Qu.:16.75                      3rd Qu.: 99.43
 Max.   :21.00                      Max.   :117.58
    m.inoc          m.sub.vs
 Min.   : 35.02   Min.   :0.0000
 1st Qu.: 50.06   1st Qu.:0.3591
 Median : 72.59   Median :0.5864
 Mean   : 68.39   Mean   :0.5605
 3rd Qu.: 80.05   3rd Qu.:0.8130
 Max.   :100.08   Max.   :1.0191
```

Calculating cumulative biogas and $CH_4$ production and production rates, is the first step in processing data from a BMP trial. Subsequently, BMP can be calculated by the high-level function summBg() included in the biogas package. Cumulative biogas and $CH_4$ production and production rates from pressure data with `sludgeTwoBiogas` and `sludgeTwoSetup` data frames as input, can be calculated from `cumBgMan()`.

To calculate $CH_4$ production from these "longcombo" data, we must provide values for at least `dat` and `comp`, which is in a combined data frame (we will use `sludgeTwoBiogas`), and `temp` (biogas temperature) along with the names of a few columns in our input data frame.

We can use the default values `"longcombo"`, `"id"`, and `"pres"` for the `data.struct`, `id.name`, and `dat.name` arguments, respectively. Whereas, the `time.name` and `comp.name` arguments need to be specified as `"time.d"` and `"xCH4n"`, respectively. Similar, default values can be used for `cmethod = "removed"`, evaluating $CH_4$ concentration based on normalized $CH_4$ and $CO_2$ values, for `imethod = "linear"`, resulting in internal linear interpolation of xCH4n by calling the `interp` function, and for `extrap = FALSE`.

Initial headspace temperature and pressure and post venting headspace pressure are set as constant values using the `temp.init`, `pres.init`, and `pres.resid` arguments, respectively. Initial and post venting headspace pressure can be absolute or gauge depending on the value of the `absolute` argument. We will set `absolute = FALSE` to account for pressure measurements provided in gauge. In order to calculate absolute pressure from gauge pressure measurements when `absolute = FALSE`, a single absolute ambient pressure value is required. Here we will set `pres.amp` to 1013 mbar. Note the unit of the pressure data in `strawPressure`. Default unit is atm, but this can be changed using the `unit.pres` argument. In this example we will set the `unit.pres = "mbar"` to match unit of `dat.name` and `pres.resid` column. Absolute ambient pressure should be provided in the same unit as defined in the `unit.pres` argument.

Finally, headspace volumes are provided from `sludgeTwoSetup`. The data frame containing headspace volumes is defined using the `headspace` argument,

whereas the default value `"vol.hs"` can be used for the column containing headspace volume data.

```
> cum.prod.lc <- cumBgMan(sludgeTwoBiogas, temp = 30,
+                         time.name = "time.d", comp.name = "xCH4n",
+                         temp.init = 30, pres.init = 0.0, pres.resid = 0,
+                         headspace = sludgeTwoSetup,
+                         pres.amb = 1013, absolute = FALSE,
+                         unit.pres = "mbar")
```

Note the message about standard temperature and pressure–it is important to make sure these values are correct, therefore users are reminded by a message[2]. Also, note warning about pressure unit. This warning is to make sure all pressure measurements are gauge, as absolute is set to FALSE, meaning pressure data are corrected to absolute pressure by the cumBgMan function.

The output looks like this:

```
> head(cum.prod.lc)
```

```
  id    time.d pres mass.init mass.final   xCH4   xCO2     xCH4n
1  1 0.0000000    0  187.4168   187.4168 0.0000 0.0000 0.0000000
2  1 0.6270833  368  187.4149   187.3814 0.1010 0.1004 0.5014896
3  1 1.6027778  286  187.3823   187.3568 0.2253 0.1610 0.5832255
4  1 1.9972222  134  187.3538   187.3447 0.2485 0.1800 0.5799300
5  1 2.7152778  142  187.3406   187.3287 0.3246 0.2062 0.6115298
6  1 3.6180556  168  187.3295   187.3150 0.3674 0.2222 0.6231343
  vol.hs temperature pres.resid pres.abs pres.resid.abs
1  79.98          30          0     1013           1013
2  79.98          30          0     1381           1013
3  79.98          30          0     1299           1013
4  79.98          30          0     1147           1013
5  79.98          30          0     1155           1013
6  79.98          30          0     1181           1013
   rh.resid pres.resid.prev rh.resid.prev temp.prev xCH4n.prev
1 1.0000000            1013     1.0000000        30  0.0000000
2 0.7335264            1013     1.0000000        30  0.0000000
3 0.7798306            1013     0.7335264        30  0.5014896
4 0.8831735            1013     0.7798306        30  0.5832255
5 0.8770563            1013     0.8831735        30  0.5799300
6 0.8577477            1013     0.8770563        30  0.6115298
        vBg      vCH4     cvBg    cvCH4     rvBg     rvCH4
1  0.000000  0.000000  0.00000  0.00000       NA        NA
2 26.168144 13.123052 26.16814 13.12305 41.72993 20.927125
3 19.530015 11.390402 45.69816 24.51345 20.01653 11.674149
```

[2]Remember that standard conditions can be set in the function call with `temp.std` and `pres.std`, or globally with `options()`.

```
4  8.861695  5.139163 54.55985 29.65262 22.46627 13.028864
5  9.743608  5.958506 64.30346 35.61112 13.56943  8.298113
6 11.573914  7.212103 75.87738 42.82323 12.82034  7.988791
```

```
> dim(cum.prod.lc)
```

```
[1] 324  24
```

The data frame that is returned has maintained the "longcombo" structure with all the original columns in `sludgeTwoBiogas`, plus additional columns from manometric biogas calculations.

In these columns, `v` stands for (standardized) volume, `cv` (standardized) cumulative volume, `rv` stands for (standardized) volume production rate, and `Bg` and `CH4` for biogas and methane. So `cvBg` contains standardized cumulative biogas production and `cvCH4` contains standardized cumulative $CH_4$ production.

It is probably easier to understand the data in the output graphically. Here we will use the `ggplot` function from the `ggplot2` package to plot it.

```
> library(ggplot2)
> ggplot(cum.prod.lc, aes(time.d, cvCH4, colour = factor(id))) +
+   geom_point() +
+   geom_line(aes(group = id)) +
+   labs(x = "Time [d]", y = "cvCH4  [mL]", colour = "Bottle id")  +
+   theme_bw()
```

## 3.2 "long" data structure

In this example, we will use a long example data set included in the biogas package: `strawPressure` for headspace pressure in batch bottles, `strawComp` for methane content of biogas, and `strawSetup` for grouping and headspace volumes.

These data are interval-based BMP measurement of headspace pressure in 12 anaerobic bottles with straw as the substrate. The experiment included 12 batch bottles on the following six variables:

- Two bottles with inoculum and straw treated with treatment r3
- Two bottles with inoculum and straw treated with treatment r5
- Two bottles with inoculum and straw treated with treatment r6.5
- Two bottles with inoculum and straw treated with treatment r8
- Two bottles with inoculum and straw treated with treatment r8 no buff
- Two bottles with inoculum and straw treated with treatment r9

Bottles 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 bottles. More details can be found in the help file for these data.

```
> #data("strawPressure")
>
> dim(strawPressure)

[1] 72  5

> head(strawPressure)

  bottle             date.time time     pres pres.resid
1      1 2016-04-28 10:30:00.00  1.9 153.2034    101.629
2      2 2016-04-28 10:30:00.00  1.9 152.0888    101.629
3      3 2016-04-28 10:30:00.00  1.9 142.0576    101.629
4      4 2016-04-28 10:30:00.00  1.9 142.1590    101.629
5      5 2016-04-28 10:30:00.00  1.9 138.7139    101.629
6      6 2016-04-28 10:30:00.00  1.9 139.5245    101.629

> summary(strawPressure)

     bottle                       date.time       time
 Min.   : 1.00   2016-04-28 10:30:00.00:12   Min.   : 1.900
 1st Qu.: 3.75   2016-04-29 16:00:00.00:12   1st Qu.: 3.120
 Median : 6.50   2016-04-30 13:20:00.00:12   Median : 4.965
 Mean   : 6.50   2016-05-02 11:00:00.00:12   Mean   : 5.987
```

```
 3rd Qu.: 9.25   2016-05-04 14:30:00.00:12   3rd Qu.: 8.060
 Max.   :12.00   2016-05-09 10:45:00.00:12   Max.   :12.910
      pres          pres.resid
 Min.   :103.9   Min.   :101.0
 1st Qu.:125.3   1st Qu.:101.6
 Median :138.8   Median :101.8
 Mean   :140.2   Mean   :101.7
 3rd Qu.:155.0   3rd Qu.:102.0
 Max.   :180.7   Max.   :102.1

> #data("strawComp")
>
> dim(strawComp)

[1] 63  4

> head(strawComp)

  bottle              date.time time   xCH4
1      1 2016-04-28 10:30:00.00  1.9 0.4189
2      2 2016-04-28 10:30:00.00  1.9 0.4219
3      3 2016-04-28 10:30:00.00  1.9 0.3395
4      4 2016-04-28 10:30:00.00  1.9 0.3015
5      5 2016-04-28 10:30:00.00  1.9 0.2769
6      6 2016-04-28 10:30:00.00  1.9 0.2850

> summary(strawComp)

     bottle                         date.time       time
 Min.   : 1.000   2016-04-28 10:30:00.00:12   Min.   : 1.900
 1st Qu.: 3.000   2016-05-04 14:30:00.00:11   1st Qu.: 3.120
 Median : 6.000   2016-04-30 13:20:00.00:10   Median : 4.010
 Mean   : 6.175   2016-05-09 10:45:00.00: 9   Mean   : 5.724
 3rd Qu.: 9.000   2016-04-29 15:35:00.00: 1   3rd Qu.: 8.060
 Max.   :12.000   2016-04-29 15:40:00.00: 1   Max.   :12.910
                  (Other)               :19
      xCH4
 Min.   :0.01187
 1st Qu.:0.30020
 Median :0.43490
 Mean   :0.39450
 3rd Qu.:0.48925
 Max.   :0.88720

> #data("strawSetup")
>
> dim(strawSetup)
```

```
[1] 12  6
```

```
> head(strawSetup)
```

```
  bottle treatment                 start sub.mass inoc.mass
1      1        r3 2016-04-26 13:00:00.00     5.27     45.36
2      2        r3 2016-04-26 13:00:00.00     5.27     43.88
3      3        r5 2016-04-26 13:00:00.00     5.28     26.27
4      4        r5 2016-04-26 13:00:00.00     5.30     26.95
5      5      r6.5 2016-04-26 13:00:00.00     5.31     20.37
6      6      r6.5 2016-04-26 13:00:00.00     5.29     20.61
  headspace
1     491.3
2     493.3
3     502.9
4     502.4
5     502.4
6     505.8
```

```
> summary(strawSetup)
```

```
     bottle           treatment                    start
 Min.   : 1.00   r3        :2   2016-04-26 13:00:00.00:12
 1st Qu.: 3.75   r5        :2
 Median : 6.50   r6.5      :2
 Mean   : 6.50   r8        :2
 3rd Qu.: 9.25   r8 no buff:2
 Max.   :12.00   r9        :2
    sub.mass       inoc.mass       headspace
 Min.   :5.240   Min.   :14.57   Min.   :491.3
 1st Qu.:5.270   1st Qu.:16.48   1st Qu.:502.4
 Median :5.280   Median :18.76   Median :502.9
 Mean   :5.284   Mean   :23.31   Mean   :503.1
 3rd Qu.:5.300   3rd Qu.:26.44   3rd Qu.:507.5
 Max.   :5.340   Max.   :45.36   Max.   :509.5
```

As with the "longcombo" data, cumulative production of $CH_4$ is needed in order to calculate BMP. Again, we can calculate these with the `cumBgMan` function, using `strawPressure`, `strawComp`, and `strawSetup` data frames as input.

To calculate $CH_4$ production from these "long" structured data, we must provide values for at least `dat`, `comp`, and `temp` along with the names of a few columns in our input data frame. The `dat` and `comp` arguments are set as the data frames: `strawPressure` and `strawComp`, whereas `temp` is set as single values of 31 °C. For `data.struct != "longcombo"` the data structure needs to be specified. Here we set `data.struct = "long"` which is internally passed to `cumBgDataPrep()` and restructured to `"longcombo"` structure prior to entering

the manometric calculations Furthermore, we need to specify the name of the id column in `strawPressure` as `bottle` using the `id.name` argument.

We can use the default values `"time"`, `"pres"`, and `"xCH4"` for the `time.name`, `dat.name` and `comp.name` arguments, respectively. The id and time columns create a link between the two data frames. If observations are not from the same time this is solved by the external `interp` function by interpolation. Similar, default values can be used for `cmethod = "removed"`, evaluating $CH_4$ concentration based on normalized $CH_4$ and $CO_2$ values and for `imethod = "linear"`, resulting in internal linear interpolation of xCH4 by calling the `interp` function. Additionally, the response variables are interval data only and hence, we can use the default `interval = TRUE`.

In addition to interpolation for later observations, an extrapolation argument (`extrap`) can be provided if required. We do not have initial biogas composition (compare the heads of `strawPressure` and `strawComp`) so extrapolation is required to calculate initial volumes and rates. Therefore, we need to set `extrap = TRUE`.

Initial headspace temperature and pressure are set as constant values using the `temp.init` and `pres.init` arguments, respectively. Whereas, headspace pressure after venting is provided from the `strawPressure` data frame and can be defined by assigning the column name to the `pres.resid` argument. Similar to the "longcombo" data example, we will set `absolute = FALSE` to account for pressure measurements provided in gauge. In order to calculate absolute pressure from gauge pressure measurements when `absolute = FALSE`, a single absolute ambient pressure value is required. To correct gauge pressure data to absolute pressure, we will set `pres.amp` to 101.3 kPa. Again, note the unit of the pressure data in `strawPressure`. Pressure unit is set as atm by default, but this can be changed using the `unit.pres` argument. In this example we will set the `unit.pres = "kpa"` to match unit of `dat.name` and `pres.resid` column.

Finally, headspace volumes are provided from `strawSetup`. The data frame containing these volumes is defined using the `headspace` argument, whereas the column is set as `vol.hs.name = "headspace"`.

```
> cum.prod.l <- cumBgMan(strawPressure, comp = strawComp, temp = 31,
+                        data.struct = "long",
+                        id.name = "bottle",
+                        temp.init = 21.55, pres.resid = "pres.resid", pres.init = 0.0,
+                        headspace = strawSetup, vol.hs.name = "headspace",
+                        pres.amb = 101.3, absolute = FALSE,
+                        extrap = TRUE,
+                        unit.pres = "kPa")
> head(cum.prod.l)

  bottle             date.time time     pres pres.resid
1      1                  <NA> 0.00       NA         NA
2      1 2016-04-28 10:30:00.00 1.90 153.2034   101.6290
3      1 2016-04-29 16:00:00.00 3.12 171.9485   101.8316
```

```
4      1 2016-04-30 13:20:00.00 4.01 140.7404   101.8316
5      1 2016-05-02 11:00:00.00 5.92 168.7061   102.1356
6      1 2016-05-04 14:30:00.00 8.06 168.4022   102.0343
       xCH4 headspace temperature pres.abs pres.resid.abs
1        NA        NA          NA       NA             NA
2 0.4189000     491.3          31 254.5034       202.9290
3 0.4363556     491.3          31 273.2485       203.1316
4 0.4413000     491.3          31 242.0404       203.1316
5 0.5014287     491.3          31 270.0061       203.4356
6 0.5152000     491.3          31 269.7022       203.3343
   rh.resid pres.resid.prev rh.resid.prev temp.prev xCH4.prev
1        NA             NA            NA        NA        NA
2 0.7973527        101.3000     1.0000000     21.55 0.0000000
3 0.7433951        202.9290     0.7973527     31.00 0.4189000
4 0.8392467        203.1316     0.7433951     31.00 0.4363556
5 0.7534481        203.1316     0.8392467     31.00 0.4413000
6 0.7539216        203.4356     0.7534481     31.00 0.5014287
       vBg      vCH4      cvBg    cvCH4     rvBg     rvCH4
1   0.0000   0.00000    0.0000   0.0000       NA        NA
2 644.8100 270.11092  644.8100 270.1109 339.3737 142.16364
3 302.1496 131.84467  946.9597 401.9556 247.6636 108.06940
4 164.3333  72.52027 1111.2929 474.4759 184.6441  81.48345
5 287.9775 144.40021 1399.2705 618.8761 150.7736  75.60220
6 283.6379 146.13027 1682.9084 765.0063 132.5411  68.28517
```

Note the message about standard temperature and pressure–it is important to make sure these values are correct, therefore users are reminded by a message[3]. Also, note warning concerning pressure ranges. The pressure ranges from 101.3 to 203.4 kPa is reasonable and warning can be ignored. We are aware of this issue and it will be addressed (refer to Section **??**).

The output looks like this:

```
> head(cum.prod.l)

  bottle              date.time time     pres pres.resid
1      1                   <NA> 0.00       NA          NA
2      1 2016-04-28 10:30:00.00 1.90 153.2034   101.6290
3      1 2016-04-29 16:00:00.00 3.12 171.9485   101.8316
4      1 2016-04-30 13:20:00.00 4.01 140.7404   101.8316
5      1 2016-05-02 11:00:00.00 5.92 168.7061   102.1356
6      1 2016-05-04 14:30:00.00 8.06 168.4022   102.0343
       xCH4 headspace temperature pres.abs pres.resid.abs
1        NA        NA          NA       NA             NA
2 0.4189000     491.3          31 254.5034       202.9290
```

[3]Remember that standard conditions can be set in the function call with `temp.std` and `pres.std`, or globally with `options()`.

```
3 0.4363556     491.3          31 273.2485       203.1316
4 0.4413000     491.3          31 242.0404       203.1316
5 0.5014287     491.3          31 270.0061       203.4356
6 0.5152000     491.3          31 269.7022       203.3343
   rh.resid pres.resid.prev rh.resid.prev temp.prev xCH4.prev
1        NA              NA            NA        NA        NA
2 0.7973527        101.3000     1.0000000     21.55 0.0000000
3 0.7433951        202.9290     0.7973527     31.00 0.4189000
4 0.8392467        203.1316     0.7433951     31.00 0.4363556
5 0.7534481        203.1316     0.8392467     31.00 0.4413000
6 0.7539216        203.4356     0.7534481     31.00 0.5014287
       vBg      vCH4      cvBg    cvCH4     rvBg     rvCH4
1   0.0000   0.00000    0.0000   0.0000       NA        NA
2 644.8100 270.11092  644.8100 270.1109 339.3737 142.16364
3 302.1496 131.84467  946.9597 401.9556 247.6636 108.06940
4 164.3333  72.52027 1111.2929 474.4759 184.6441  81.48345
5 287.9775 144.40021 1399.2705 618.8761 150.7736  75.60220
6 283.6379 146.13027 1682.9084 765.0063 132.5411  68.28517

> dim(cum.prod.l)

[1] 84 21
```

The data frame that is returned has been restructured to "longcombo" structure and contains all the original columns in `strawPressure`, plus additional columns from volumetric biogas calculations (refer to Section **??**)

As with the "longcombo" data example, the results may be easier to understand graphically using the `ggplot` function from the `ggplot2` package.

```
> ggplot(cum.prod.l, aes(time, cvCH4, colour = factor(bottle))) +
+   geom_point() +
+   geom_line(aes(group = bottle)) +
+   labs(x = "Time [d]", y = "cvCH4  [mL]", colour = "Bottle id")  +
+   theme_bw()
```

### 3.3 Additional examples

For examples using "wide" structured input data frame and varying settings of other arguments (e.g. `addt0` and `showt0`), refer to vignette for volumetric cumBg function (`cumBgVol_function.Rnw`).

## 4 Continuing with the cumBgMan function

The cumBgMan function is one of several cumBg* functions within the biogas package. Results from the cumBg* functions can be used directly in the summBg function from the biogas package to calculate BMP for the data in question. Though, this operation often requires additional setup information (e.g. inoculum and substrate mass), which is most commonly provided in an external data frame (`setup`). More details can be found in the associated help file

In the current version of `cumBgMan()`, an issue is commonly observed with the warning regarding pressure range during cumulative biogas calculations. The warning is sometimes returned with the results even though range is reasonable. This has been raised as an issue and should be addressed in a future scope.

## 5 Function internals

Table 1: Operations done with the low-level functions in `cumBgMan()`. All functions are vectorized. See help files for more details.

| Operation | Function |
|---|---|
| Standardize gas volume | `stdVol()` |
| Interpolate composition etc. | `interp()` |
| Structurize and sort data | `cumBgDataPrep()` |

## References


[1] J. Filer, H. H. Ding and S. Chang Biochemical Methane Potential (BMP) Assay Method for Anaerobic Digestion Research. *Water*, 11, 921, 2019.

[2] B.K. Richards, R.J. Cummings, T.E. White, and W.J. Jewell. Methods for kinetic-analysis of methane fermentation in high solids biomass digesters. *Biomass & Bioenergy*, 1(2):65–73, 1991.