¿ 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 volumetric methods

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

September 23, 2019

1 Introduction

Volumetric 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 volumetric methods, biogas volumes are measured under constant pressure conditions from different methods such as liquid displacement and syringe methods [**?**]. In liquid displacement method, the biogas is transferred to an external system measuring the volume. As biogas is produced it is collected in the external liquid displacement system where it displaces an amount of the barrier liquid equivalent to the biogas volume. In the syringe method, lubricated syringes are manually placed in the bottle of interest. Here, the pressure increase resulting from biogas production forces the piston inwards the syringe until the pressure equilibrates the surrounding pressure (most commonly atmospheric). Produced biogas volume is equivalent to the displacement of the piston and can be read directly off the syringe [**?**]. 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.

A newly developed biogas package address issues with time-consuming calculations and lack of reproducible among laboratories for obtaining BMP. The biogas package consists of ten function including cumBg(), which is used to calculate cumulative production of biogas and methane (CH_4) and production rates with either volumetric, manometric, gravimetric or gas density methods. These production values and rates can be furhter used to calculate BMP. cumBg() is a large and rather complicated function, which requires some proficiency in R for use. We developed a function to process volumetric 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 descrip-

tion of the volumetric biogas calculation function (`cumBgVol()`) for new users. We have assumed that readers are familiar with biogas data collection and R.

2 Overview of the function

`cumBgVol()` is a "high-level" function within the biogas package. The purpose of `cumBgVol()` is to convert volume 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 and accepts different data structures such as "wide", "long", and "longcombo". For simple operations (e.g. interpolation and standardization of biogas volume) `cumBgVol()` 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 `cumBgVol()`.

In general, cumBg* functions are compiled of four sections: check arguments, restructure and sorting data, interpolation if needed, and biogas standardization and calculations. Restructuring and sorting 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 `cumBgVol()` 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. If data of concern are mixed (interval and cumulative, `empty.name = TRUE`), these will be sorted and biogas volume standardized within `cumBgDataPrep()` to obtain interval data only. Subsequently, the now restructured and sorted data is standardized in `cumBgVol()` by an external function called `stdVol()`, if not already standardized.

Two methods are commonly used to evaluate volumetric biogas measurements. Method 1 is based on normalized CH_4 concentrations, whereas method 2 accounts for the actual CH_4 in the bottle headspace. Both methods are available through `cumBgVol()` and results is expected to be independent of method. The examples below describe cumulative biogas calculation from three datasets of different structures ("wide", "long", and "longcombo") using volumetric method 1. For method 2 examples, refer to vignette for manometric cumBg function (`cumBgMan_function.Rnw`). All external functions are within the biogas package

2.1 Function arguments

The arguments for the function are:

```
## function (dat, comp = NULL, temp = NULL, pres = NULL,
##           interval = TRUE, data.struct = "longcombo",
##           id.name = "id", time.name = "time",
##           dat.name = "vol", comp.name = "xCH4",
```

```
##            headspace = NULL, vol.hs.name = "vol.hs",
##            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 response variable - here volume measurements), `comp` (data frame with methane concentration), and `temp` (biogas temperature)[1] along with the names of a few columns in the input data frames. If `temp` and/or `pres` arguments are not provided, biogas volumes will not be standardized.

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. When data are mixed interval and cumulative response variables (`empty.name != TRUE`), the interval argument is ignored (refer to Section **??**).

Similarly, there is an `id.name` argument for the bottle identification code (ID) column (default = "id"). The default value is `"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 cumBgVol 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 the CH_4 content (as mole fraction in dry biogas, normalized

[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 argurments.

so the sum of mole fractions of CH_4 and CO_2 sum to unity) (default = xCH4). The `comp` argument may be a single value instead of a data frame or column. When providing a single value for `comp`, this value is applied to all observations, and the `comp.name` argument is not needed. Note that if the `comp` argument is not provided, `cumBgVol()` will return calculations on biogas only and no calculations on CH_4. 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 CO_2 only.

If any CH_4 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 extrapolated (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.

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").

By default biogas is assumed to be saturated with water vapor. If biogas volumes are already standardized to dry conditions (as with AMPTS II data) we need to set `dry = TRUE`.

2.2 Data Structures

Input data may be structured in one of three ways: "longcombo", "long", and "wide." 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 volumetric calculation methods within `cumBgVol()` 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 volumetric measurements

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: `s3lcombo` for both biogas volumes and composition. These data are from an experiment carried out with municipal wastewater sludge. Only three bottles, all with inoculum and substrate, were included: D, E, and F. 300 mL glass serum bottles with butyl rubber septa and screw caps were used. Volume was measured using syringes. Methane and carbon dioxide contents were determined by gas chromatography using a thermal conductivity detector and normalized so methane and carbon dioxide sum to 1.0. This is a small dataset, with versions available for all three different data structures. It cannot be used to calculate BMP, because no inoculum-only bottles were included.

```
> data("s3lcombo")
> dim(s3lcombo)

[1] 21  4

> s3lcombo

   id  time.d vol.ml   xCH4
1   D  0.9438    103     NA
2   E  0.9451    106     NA
3   F  0.9472    107     NA
4   D  2.9060    192 0.6983
5   E  2.9090    181 0.6817
6   F  2.9100    203 0.6869
7   D  5.8860    141 0.6800
8   E  5.8880    133 0.6800
9   F  5.8900    140 0.6800
10  D 10.0000    112 0.6646
11  E 10.0000    111 0.6644
12  F 10.0100    110 0.6632
13  D 23.1000    200 0.6946
14  E 23.1000    190 0.6871
15  F 23.1000    200 0.6829
16  D 34.0100    109 0.6626
17  E 34.0100    110 0.6556
```

```
18  F 34.0100    112 0.6527
19  D 57.8400    146 0.6651
20  E 57.8400    136 0.6600
21  F 57.8400    138     NA
```

```
> summary(s3lcombo)
```

```
 id        time.d            vol.ml           xCH4
 D:7   Min.   : 0.9438   Min.   :103.0   Min.   :0.6527
 E:7   1st Qu.: 2.9100   1st Qu.:110.0   1st Qu.:0.6632
 F:7   Median :10.0000   Median :136.0   Median :0.6800
       Mean   :19.2421   Mean   :141.9   Mean   :0.6741
       3rd Qu.:34.0100   3rd Qu.:181.0   3rd Qu.:0.6829
       Max.   :57.8400   Max.   :203.0   Max.   :0.6983
                                         NA's   :4
```

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 volumetric data with `s3lcombo` data frame as the only input, can be calculated from `cumBgVol()`.

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 `s3lcombo`), `temp` (biogas temperature), and `pres` (biogas pressure) along with the names of a few columns in our input data frame.

We can use the default values `"longcombo"`, `"id"`, and `"xCH4"` for the `data.struct`, `id.name`, and `comp.name` arguments, respectively. Whereas, the `dat.name` argument needs to be specified as `"vol.ml"`. 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. Furthermore, initial composition (xCH4) values are missing in the `s3lcombo` data frame. Therefore, we set `extrap = TRUE` for extrapolation of these initial CH_4 values

By default biogas is assumed to be saturated with water vapor (`dry = FALSE`). Biogas volumes have not been standardized to dry conditions and hence, default value for the `dry` argument, matches requirements for making an actual evaluation of the dataset.

```
> cum.prod.lc <- cumBgVol(s3lcombo, temp = 25, pres = 1,
+                         time.name = 'time.d',
+                         dat.name = 'vol.ml',
+                         extrap = TRUE)
```

The output looks like this:

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

```
  id  time.d vol.ml   xCH4 temperature pressure       vBg
1  D  0.0000     NA     NA          NA       NA   0.00000
2  D  0.9438    103 0.6983          25        1  91.40334
3  D  2.9060    192 0.6983          25        1 170.38293
4  D  5.8860    141 0.6800          25        1 125.12497
5  D 10.0000    112 0.6646          25        1  99.39004
6  D 23.1000    200 0.6946          25        1 177.48222
       vCH4      cvBg     cvCH4     rvBg     rvCH4
1   0.00000   0.00000   0.00000       NA        NA
2  63.82696  91.40334  63.82696 96.84609 67.627628
3 118.97840 261.78628 182.80536 86.83260 60.635206
4  85.08498 386.91124 267.89033 41.98824 28.552006
5  66.05462 486.30129 333.94496 24.15898 16.056058
6 123.27915 663.78351 457.22411 13.54826  9.410622

> dim(cum.prod.lc)

[1] 24 12
```

The data frame that is returned has maintained the "longcombo" structure with all the original columns in `s3lcombo` and contains additional columns from volumetric 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'll use the `ggplot` function from the `ggplot2` package to plot it.

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

3.2 ”wide” data structure

In this example, we will use a ”wide” structured example dataset included in the biogas package, having the data frame `feedVol` for biogas volumes and `feedSetup` for bottle identification and substrate and inoculum masses. The `feedsetup` data frame is not relevant for `cumBgVol()`, but is required subsequently for `summBg()` to calculate BMP. BMP measurement data are from a batch test carried out on animal feed ingredients along with cellulose as a control. The experiment included 12 batch bottles:

- Three bottles with inoculum only (BK)
- Three bottles with cellulose and inoculum (CEL)
- Three bottles with animal feed ingredient 1 and inoculum (SC)
- Three bottles with animal feed ingredient 2 and inoculum (SD)

A typical automated volumetric method called AMPTS II was used to measure biogas production: an online, standardized lab-measurement platform for BMP tests. Applying AMPTS II, the measured volumes are standardized, and the composition is 100% methane. Therefore, the `comp` argument is set to 1 when calling the cumBgVol function. Furthermore, pressure is set to a fixed value (1 atm or 101.325 kPa) and temperature to 0°C.

```
> data("feedVol")
> dim(feedVol)

[1] 44 13

> head(feedVol)

  time.d     1     2     3      4      5      6      7      8
1      0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0
2      1  70.6  68.5  69.7   78.6   78.1   85.9  383.8  390.7
3      2 100.5  97.6 100.3  357.0  369.0  384.4  902.4  927.4
4      3 125.0 120.8 125.9  846.0  895.2  902.4 1517.3 1563.0
5      4 144.0 138.3 144.5 1291.5 1323.7 1373.0 1911.5 1940.0
6      5 158.6 152.1 158.9 1513.2 1542.1 1600.1 2044.9 2080.7
       9     10     11     12
1    0.0    0.0    0.0    0.0
2  399.3  249.5  277.8  280.6
3  927.0  521.1  561.8  562.9
4 1557.6  830.6  876.9  876.0
5 1955.3 1112.7 1160.3 1150.0
6 2082.0 1348.9 1413.0 1388.9

> summary(feedVol)
```

```
    time.d            1               2               3
Min.   : 0.00   Min.   :  0.0   Min.   :  0.0   Min.   :  0.0
1st Qu.:10.75   1st Qu.:214.2   1st Qu.:200.7   1st Qu.:212.2
Median :21.50   Median :256.4   Median :239.7   Median :251.3
Mean   :21.50   Mean   :228.7   Mean   :213.5   Mean   :225.1
3rd Qu.:32.25   3rd Qu.:269.3   3rd Qu.:249.8   3rd Qu.:264.5
Max.   :43.00   Max.   :269.3   Max.   :249.8   Max.   :264.5
      4               5               6               7
Min.   :   0   Min.   :   0   Min.   :   0   Min.   :   0
1st Qu.:1877   1st Qu.:1892   1st Qu.:1915   1st Qu.:2328
Median :1975   Median :1995   Median :1995   Median :2558
Mean   :1784   Mean   :1807   Mean   :1807   Mean   :2339
3rd Qu.:2011   3rd Qu.:2047   3rd Qu.:2022   3rd Qu.:2645
Max.   :2025   Max.   :2047   Max.   :2022   Max.   :2675
      8               9              10              11
Min.   :   0   Min.   :   0   Min.   :   0   Min.   :   0
1st Qu.:2376   1st Qu.:2367   1st Qu.:2247   1st Qu.:2335
Median :2603   Median :2603   Median :2818   Median :2866
Mean   :2373   Mean   :2377   Mean   :2455   Mean   :2504
3rd Qu.:2674   3rd Qu.:2685   3rd Qu.:3015   3rd Qu.:3055
Max.   :2681   Max.   :2699   Max.   :3124   Max.   :3141
      12
Min.   :   0
1st Qu.:2298
Median :2854
Mean   :2487
3rd Qu.:3044
Max.   :3120
```

As with the "longcombo" data, cumulative production of CH_4 is needed in order to calculate BMP. Again, we can calculate these with the `cumBgVol()` function, using `feedVol` data frame and `comp = 1` argument as input. For this particular example, standardized, cumulative CH_4 volume is present in the input data frame (as with all AMPTS II datasets) and the function only restructures the data and calculates interval volumes and rates. In other cases (see `s3volw` for example), volume standardization and other steps (e.g., interpolation or summing interval values to get cumulative) are needed.

To calculate CH_4 production from these "wide" structured data, we must provide values for at least `dat`, `comp`, `temp`, and `pres` along with the names of a few columns in our input data frame. The `dat` argument is set as the `feedVol` data frame, whereas `comp`, `temp`, and `pres` are set as single values of 1 (mole fraction), 0 (degree C), and 1 (atm), respectively.

For `data.struct != "longcombo"` the data structure needs to be specified. Here we set `data.struct = "wide"` which is internally passed to `cumBgDataPrep()` and restructured to `"longcombo"` structure prior to entering volumetric calculation methods. Furthermore, we need to specify the name of the time

column in `feedVol` using the `time.name` argument. We have provided a single value for `comp`, meaning that the `comp.name` argument is not required.

As mentioned, there are no ID columns for `data.struct = "wide"`. Instead column name of first column with response variables is used as ID code. In this example observations are numbered 1 to 12 and hence, the `dat.name` argument is set to 1. All following columns are assumed to also contain response variables (volume measurements). By default `cmethod = "removed"` evaluating CH_4 concentration based on normalized CH_4 and CO_2 values. Similar, default values can be used for `imethod = "linear"`, resulting in internal linear interpolation of xCH4 by calling the `interp` function, and `extrap = FALSE` as all composition data are provided (set to 1 for all observations). By default biogas is assumed to be saturated with water vapor. For AMPTS II data, biogas volume are already standardized to dry conditions. Therefore, we set `dry = TRUE`. Additionally, the response variables are cumulative data and hence, we set `interval = FALSE`.

```
> cum.prod.w <- cumBgVol(feedVol, comp = 1, temp = 0, pres = 1,
+                        data.struct = "wide",
+                        time.name = "time.d", dat.name = "1",
+                        dry = TRUE,
+                        interval = FALSE)
```

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 the message about applying single composition value to all observations. When defining pressure as a single value, pressure is assumed to be constant and the same for all observations.

The output looks like this:

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

	id	time.d	vol	xCH4	temperature	pressure	vBg	vCH4	cvBg	cvCH4	rvBg	rvCH4
1	1	0	0.0	1	0	1	0.0	0.0	0.0	0.0	NA	NA
2	1	1	70.6	1	0	1	70.6	70.6	70.6	70.6	70.6	70.6
3	1	2	100.5	1	0	1	29.9	29.9	100.5	100.5	29.9	29.9
4	1	3	125.0	1	0	1	24.5	24.5	125.0	125.0	24.5	24.5
5	1	4	144.0	1	0	1	19.0	19.0	144.0	144.0	19.0	19.0
6	1	5	158.6	1	0	1	14.6	14.6	158.6	158.6	14.6	14.6

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

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

[1] 528  12
```

The data frame that is returned has been sorted and restructured to "longcombo" structure and has all the original columns in `feedVol`, plus additional columns from the volumetric calculation method (ref. "longcombo" data structure 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.w, aes(time.d, cvCH4, colour = factor(id))) +
+        geom_point() +
+        geom_line(aes(group = id)) +
+        labs(x = "Time [d]", y = "Cumulative methane production  [mL]",
+             colour = "Bottle ID")  +
+        theme_bw()
```

3.3 "long" data structure

In this example, we will use the example data sets included with the biogas package: `vol` for biogas volumes and `comp` for composition. These data are from a BMP test carried out on two different substrates A and B, and cellulose included as a "control". The experiment included 12 batch bottles:

- 3 bottles with substrate A and inoculum
- 3 bottles with substrate B and inoculum
- 3 bottles with cellulose and inoculum
- 3 bottles with inoculum only

Reactors consisted of 500 mL or 1.0 L glass bottles, and were sealed with a butyl rubber septum and a screw cap. Initial substrate and inoculum masses were determined. A typical volumetric method was used to measure biogas production: accumulated biogas was measured and removed intermittently using syringes, and composition was measured for some of these samples.

```
> data("vol")
> dim(vol)

[1] 288   4

> head(vol)

   id           date.time days vol
1 2_1 2014-06-07 13:00:00 1.98 393
2 2_1 2014-06-08 13:00:00 2.98 260
3 2_1 2014-06-09 13:00:00 3.98 245
4 2_1 2014-06-10 13:00:00 4.98 225
5 2_1 2014-06-11 13:00:00 5.98 200
6 2_1 2014-06-12 14:00:00 7.02 175

> summary(vol)

       id        date.time                       days
 2_1    : 24   Min.   :2014-06-07 13:00:00   Min.   :  1.98
 2_2    : 24   1st Qu.:2014-06-14 02:00:00   1st Qu.:  8.52
 2_3    : 24   Median :2014-06-28 12:00:00   Median : 22.94
 2_4    : 24   Mean   :2014-07-16 21:29:22   Mean   : 41.33
 2_5    : 24   3rd Qu.:2014-07-26 04:45:00   3rd Qu.: 50.63
 2_6    : 24   Max.   :2014-12-19 10:30:00   Max.   :196.92
 (Other):144
      vol
 Min.   : 98.0
 1st Qu.:171.5
```

```
 Median :225.0
 Mean   :271.7
 3rd Qu.:300.0
 Max.   :840.0

> data("comp")
> dim(comp)

[1] 132   4

> head(comp)

     id           date.time  days      xCH4
516 2_1 2014-06-12 14:00:00  7.02 0.7104731
519 2_1 2014-06-19 14:00:00 14.02 0.7024937
522 2_1 2014-06-26 11:00:00 20.90 0.6659919
524 2_1 2014-07-03 10:00:00 27.85 0.6789466
525 2_1 2014-07-10 09:00:00 34.81 0.6951429
528 2_1 2014-07-24 10:00:00 48.85 0.6693053

> summary(comp)

       id       date.time                      days
 2_1    :11   Min.   :2014-06-12 14:00:00   Min.   :  7.02
 2_2    :11   1st Qu.:2014-06-26 11:00:00   1st Qu.: 20.90
 2_3    :11   Median :2014-07-24 10:00:00   Median : 48.85
 2_4    :11   Mean   :2014-07-31 13:47:43   Mean   : 56.01
 2_5    :11   3rd Qu.:2014-08-28 10:00:00   3rd Qu.: 83.85
 2_6    :11   Max.   :2014-10-13 13:00:00   Max.   :129.98
 (Other):66
      xCH4
 Min.   :0.5647
 1st Qu.:0.6393
 Median :0.6598
 Mean   :0.6587
 3rd Qu.:0.6786
 Max.   :0.7115
```

To calculate CH_4 production from these "long" structured data, we must provide values for at least `dat` (we will use `vol`), `comp` (we will use `comp`), `temp`, and `pres`. 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 volumetric calculation methods. Furthermore, we need to specify the name of the time column in `feedVol` using the `time.name` argument.

We can use the default values `"id"`, `"vol"`, and `"xCH4"` for the `id.name`, `dat.name` and `comp.name` arguments, 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 xCH4 by calling the `interp` function, and for `dry = TRUE` assuming biogas to be saturated with water vapor. 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 the `vol` and `comp` data frames) so we will need to extrapolate, in addition to interpolation. Therefore, we set `extrap = TRUE`.

```
> cum.prod.l <- cumBgVol(vol, comp = comp, temp = 35, pres = 1,
+                        data.struct = "long",
+                        time.name = "days",
+                        extrap = TRUE)
```

The output looks like this:

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

```
   id           date.time days vol      xCH4 temperature
1 2_1                <NA> 0.00  NA        NA          NA
2 2_1 2014-06-07 13:00:00 1.98 393 0.7104731          35
3 2_1 2014-06-08 13:00:00 2.98 260 0.7104731          35
4 2_1 2014-06-09 13:00:00 3.98 245 0.7104731          35
5 2_1 2014-06-10 13:00:00 4.98 225 0.7104731          35
6 2_1 2014-06-11 13:00:00 5.98 200 0.7104731          35
  pressure      vBg     vCH4      cvBg    cvCH4     rvBg
1       NA   0.0000   0.0000    0.0000   0.0000       NA
2        1 328.9470 233.7080  328.9470 233.7080 166.1348
3        1 217.6240 154.6160  546.5710 388.3240 217.6240
4        1 205.0687 145.6958  751.6397 534.0198 205.0687
5        1 188.3284 133.8023  939.9681 667.8221 188.3284
6        1 167.4031 118.9354 1107.3712 786.7574 167.4031
     rvCH4
1       NA
2 118.0343
3 154.6160
4 145.6958
5 133.8023
6 118.9354
```

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

```
[1] 300  13
```

The data frame that is returned has been restructured to "longcombo" structure and contains all the original columns in `vol`, plus additional columns from volumetric biogas calculations (ref. section "longcombo" data structure)

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(days, cvCH4, colour = factor(id))) +
+       geom_point() +
+       geom_line(aes(group = id)) +
+       labs(x = "Time [d]", y = "Cumulative methane production  [mL]",
+            colour = "Bottle ID")  +
+       theme_bw()
```

3.4 Other arguments

Instead of setting fixed values for temperature and pressure arguments, these can be taken from columns within `dat`. Fixed data were made adding temperature and pressure to the `vol` data frame from the "long" structure example above. Temperature is set to a fixed value for all observations, whereas pressure is set to vary between observations using `rnorm()` having a mean value of 1 atm and a standard deviation of 0.001. It is possible to have temperature and pressure values that varies between measurements. If this was the case, the approach (function call) would be the same as shown below.

```
> data("vol")
> dim(vol)

[1] 288   4

> head(vol)

   id           date.time days vol
1 2_1 2014-06-07 13:00:00 1.98 393
2 2_1 2014-06-08 13:00:00 2.98 260
3 2_1 2014-06-09 13:00:00 3.98 245
4 2_1 2014-06-10 13:00:00 4.98 225
5 2_1 2014-06-11 13:00:00 5.98 200
6 2_1 2014-06-12 14:00:00 7.02 175

> summary(vol)

       id        date.time                      days
 2_1    : 24   Min.   :2014-06-07 13:00:00   Min.   :  1.98
 2_2    : 24   1st Qu.:2014-06-14 02:00:00   1st Qu.:  8.52
 2_3    : 24   Median :2014-06-28 12:00:00   Median : 22.94
 2_4    : 24   Mean   :2014-07-16 21:29:22   Mean   : 41.33
 2_5    : 24   3rd Qu.:2014-07-26 04:45:00   3rd Qu.: 50.63
 2_6    : 24   Max.   :2014-12-19 10:30:00   Max.   :196.92
 (Other):144
      vol
 Min.   : 98.0
```

```
 1st Qu.:171.5
 Median :225.0
 Mean   :271.7
 3rd Qu.:300.0
 Max.   :840.0
```

```
> data("comp")
> dim(comp)
```

```
[1] 132   4
```

```
> head(comp)
```

```
     id           date.time  days      xCH4
516 2_1 2014-06-12 14:00:00  7.02 0.7104731
519 2_1 2014-06-19 14:00:00 14.02 0.7024937
522 2_1 2014-06-26 11:00:00 20.90 0.6659919
524 2_1 2014-07-03 10:00:00 27.85 0.6789466
525 2_1 2014-07-10 09:00:00 34.81 0.6951429
528 2_1 2014-07-24 10:00:00 48.85 0.6693053
```

```
> summary(comp)
```

```
       id       date.time                       days
 2_1    :11   Min.   :2014-06-12 14:00:00   Min.   :  7.02
 2_2    :11   1st Qu.:2014-06-26 11:00:00   1st Qu.: 20.90
 2_3    :11   Median :2014-07-24 10:00:00   Median : 48.85
 2_4    :11   Mean   :2014-07-31 13:47:43   Mean   : 56.01
 2_5    :11   3rd Qu.:2014-08-28 10:00:00   3rd Qu.: 83.85
 2_6    :11   Max.   :2014-10-13 13:00:00   Max.   :129.98
 (Other):66
      xCH4
 Min.   :0.5647
 1st Qu.:0.6393
 Median :0.6598
 Mean   :0.6587
 3rd Qu.:0.6786
 Max.   :0.7115
```

Adding temperature and pressure to the `dat` data frame.

```
> vol$temp <- 35
> vol$pres <- NA
> vol$pres <- rnorm(vol$pres, mean = 1, sd = 0.001)
> head(vol)
```

```
   id           date.time days vol temp      pres
1 2_1 2014-06-07 13:00:00 1.98 393   35 0.9992407
```

```
2 2_1 2014-06-08 13:00:00 2.98 260   35 1.0016996
3 2_1 2014-06-09 13:00:00 3.98 245   35 1.0010700
4 2_1 2014-06-10 13:00:00 4.98 225   35 0.9987502
5 2_1 2014-06-11 13:00:00 5.98 200   35 0.9982398
6 2_1 2014-06-12 14:00:00 7.02 175   35 0.9997723
```

The arguments required to process these data using `cumBgVol()` are the same as in the "long" data structure example above, besides for the `temp` and `pres` arguments. Instead of fixed values, we now call a column name from `vol`. Therefore, we set `temp = "temp"` and `pres = "pres"`

```
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE)
```

Similar to cum.prod.l, the data frame that is returned has been restructured to "longcombo" structure and contains all the original columns in `vol`, plus additional columns from volumetric biogas calculations (ref. section "longcombo" data structure)

By default, time zero rows are added from the `addt0` argument. Time zero rows can be omitted by setting `showt0 = FALSE`.

```
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE, showt0 = FALSE)
> head(cum.prod)

   id           date.time days vol temp      pres      xCH4
1 2_1 2014-06-07 13:00:00 1.98 393   35 0.9992407 0.7104731
2 2_1 2014-06-08 13:00:00 2.98 260   35 1.0016996 0.7104731
3 2_1 2014-06-09 13:00:00 3.98 245   35 1.0010700 0.7104731
4 2_1 2014-06-10 13:00:00 4.98 225   35 0.9987502 0.7104731
5 2_1 2014-06-11 13:00:00 5.98 200   35 0.9982398 0.7104731
6 2_1 2014-06-12 14:00:00 7.02 175   35 0.9997723 0.7104731
       vBg     vCH4      cvBg    cvCH4     rvBg    rvCH4
1 328.6825 233.5201  328.6825 233.5201 166.0013 117.9394
2 218.0156 154.8942  546.6981 388.4143 218.0156 154.8942
3 205.3011 145.8609  751.9992 534.2752 205.3011 145.8609
4 188.0792 133.6252  940.0784 667.9004 188.0792 133.6252
5 167.0911 118.7137 1107.1695 786.6141 167.0911 118.7137
6 146.4424 104.0434 1253.6119 890.6575 140.8100 100.0417
```

The results are shown graphically using the `ggplot` function from the `ggplot2` package.

```
> ggplot(cum.prod, aes(days, cvCH4, colour = factor(id))) +
+       geom_point() +
+       geom_line(aes(group = id)) +
+       labs(x = "Time [d]", y = "Cumulative methane production  [mL]",
+            colour = "Bottle ID")  +
+       theme_bw()
```

Note the lack of observations in time zero in the plot.

Furthermore, we can avoid adding the time zero values in the first place by setting `addt0 = FALSE`.

```
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE, addt0 = FALSE)
> head(cum.prod)
```

```
   id           date.time days vol temp      pres      xCH4
1 2_1 2014-06-07 13:00:00 1.98 393   35 0.9992407 0.7104731
2 2_1 2014-06-08 13:00:00 2.98 260   35 1.0016996 0.7104731
3 2_1 2014-06-09 13:00:00 3.98 245   35 1.0010700 0.7104731
4 2_1 2014-06-10 13:00:00 4.98 225   35 0.9987502 0.7104731
5 2_1 2014-06-11 13:00:00 5.98 200   35 0.9982398 0.7104731
6 2_1 2014-06-12 14:00:00 7.02 175   35 0.9997723 0.7104731
       vBg     vCH4      cvBg    cvCH4     rvBg    rvCH4
1 328.6825 233.5201  328.6825 233.5201       NA       NA
2 218.0156 154.8942  546.6981 388.4143 218.0156 154.8942
3 205.3011 145.8609  751.9992 534.2752 205.3011 145.8609
4 188.0792 133.6252  940.0784 667.9004 188.0792 133.6252
5 167.0911 118.7137 1107.1695 786.6141 167.0911 118.7137
6 146.4424 104.0434 1253.6119 890.6575 140.8100 100.0417
```

Note how rates are not calculated for the first observations when `addt0 = FALSE`, in contrary to `showt0 = FALSE` where initial rates are calculated. In both cases, cumulative production values for the first observations are omitted from the plot.

```
> ggplot(cum.prod, aes(days, cvCH4, colour = factor(id))) +
+       geom_point() +
+       geom_line(aes(group = id)) +
+       labs(x = "Time [d]", y = "Cumulative methane production  [mL]",
+            colour = "Bottle ID")  +
+       theme_bw()
```

By default, input data are checked for unreasonable values using the `check` argument. If data seems unreasonable, a warning is returned. In the example below, has been set to... for the purpose of demonstrating the function of the `check` argument.

```
> #vol[10,"days"] <- NA
> #vol$temp <- "temperature"
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE)
```

Note the warning about....

If we instead set `check = FALSE`, this warning will not be returned and it is not as obvious why the function call does not work.

```
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE, check = FALSE)
```

By default, the unit of temperature and pressure values are set as degree celsius and atm, respectively. As stated above, these can be set using the `unit.temp` and `unit.pres` arguments. For the purpose of demonstrating the ussage of the two unit arguments, the already fixed pressure and temperature values in `vol` have been manipulated to be given in kelvin `K` and `kPa`, respectively.

```
> vol$temp <- 35 + 273.15
> vol$pres <- NA
> vol$pres <- rnorm(vol$pres, mean = 101.325, sd = 0.101325)
> head(vol)
```

```
   id           date.time days vol   temp     pres
1 2_1 2014-06-07 13:00:00 1.98 393 308.15 101.1627
2 2_1 2014-06-08 13:00:00 2.98 260 308.15 101.3679
3 2_1 2014-06-09 13:00:00 3.98 245 308.15 101.3893
4 2_1 2014-06-10 13:00:00 4.98 225 308.15 101.3274
5 2_1 2014-06-11 13:00:00 5.98 200 308.15 101.2316
6 2_1 2014-06-12 14:00:00 7.02 175 308.15 101.3864
```

```
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE)
```

Note the warnings about temperature and pressure range. This warning is a result of the default settings for pressure and temperature values. If we manually set the correct units, these warnings are avoided.

```
> cum.prod <- cumBgVol(vol, comp = comp, temp = "temp", pres = "pres",
+                         data.struct = "long",
+                         time.name = "days",
+                         extrap = TRUE,
+                         unit.pres = "kPa", unit.temp = "K")
```

4 Continuing with the cumBgVol function

The cumBgVol function is one of several cumBg* within the biogas package. The 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`).

5 Function internals

Table 1: Operations done with the low-level functions in `cumBgVol()`. 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] 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.

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

[3] B.K. Richards, R.J. Cummings, T.E. White, and W.J. Jewell. Methods of assessing microbial activity and inhibition under anaerobic conditions: a literature review. *Rev Environ Sci Biotechnol*, 3: 93-115, 2004.