Package 'appac'

September 29, 2024

Type Package

Title Atmospheric Pressure Peak Area Correction

Version 0.1.3

Author Ruediger Forster <ruediger.forster@gmx.net> [aut, cre]

Maintainer Ruediger Forster <ruediger.forster@gmx.net>

Description Correct the effect that ambient pressure has on the peak areas of GC/FID and GC/TCD chromatograms.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression bzip2

Imports methods, stats, dplyr, data.table, Rbeast, VGAM, gridExtra, ggplot2, ggpubr, patchwork, mathjaxr, magrittr, utils

Suggests knitr, testthat (>= 3.0.0)

RdMacros mathjaxr

ByteCompile true

BuildManual TRUE

VignetteBuilder knitr

Depends R (>= 3.5.0)

Collate 'helper.R'
'plot.R'
'S4_definitions.R'
'appac_step_1.R'
'appac_step_2.R'
'appac_step_3.R'
'appac.R'

Contents

---

appac *Atmospheric Pressure Peak Area Correction (APPAC)*

Description

APPAC is a method for posterior correction of peak areas obtained from GC/FID or GC/TCD chromatograms. The corrections are obtained from daily repetitive measurements on dedicated control samples. The algorithm takes into account the influences of atmospheric pressure and instrument drift. Change points, such as abrupt changes, in the data are detected and any resulting bias is compensated. For this purpose, the data are divided into episodes at the breakpoints. The episodes are analysed separately and then recombined. Additionally, the detector response is linearized in the output.

Usage

```
appac ( data,
        P.ref = 1013.25,
        appac.colnames = list(sample.col = NA,
                              peak.col = NA,
                              date.col = NA,
                              pressure.col = NA,
                              area.col = NA),
        appac.control  = list(min.data.points = 50)
      )
```

Arguments

`data`	a data frame containing daily repetitive measurements of one or more samples from a single GC channel.
`P.ref`	numeric. The atmospheric pressure (in hPa) to which the instrument response is corrected. Any arbitrary pressure within the üressure range of the data set may be chosen; however, in this case the means of the input and the output areas will differ. It is recommended to choose the median of the pressure measurements of the dataset.
`appac.control`	a list of the control elements which determine how the data are processed. • `min.data.points`: integer(>0). The minimum number of data points belonging to a peak from which an episode is analysed.

`appac.colnames` a list of column names of `df`, which contain the following information:

- `sample.col`: character. The name of the column containing the names of the samples
- `peak.col`: character. The name of the column containing the names of the peaks
- `pressure.col`: character. The name of the column containing the measured atmospheric pressures
- `area.col`: character. The name of the column containing the measured peak areas

Details

Sample and peak names in the input data frame are used as variables in the code and thus should comply to **R** naming conventions. A valid variable name must start with a letter and may contain letters, digits, period(.) and underscore(_). If this is not the case, non-complying names will be converted and a warning will be thrown.

Dates in the input dataframe must comply to the format `%Y-%m-%d`. See `strptime` for further information.

appac makes extensive use of the correlation, which can be found in the peak areas of the chromatograms of identical samples. It is reuired that the chromatograms to be evaluated contain at least two evaluated peaks, however more than two evaluated peaks are highly recommended.

Value

an object of class `Appac`

Warning

....

Examples

```
## Please be patient, the calculation takes a while.

## Not run:
library(appac)

## PLOT_FID is a long dataset of daily measurements of 5 control samples taken on
## a dedicated natural gas analyzer.
data("PLOT_FID")

## truncate the data set in favor of a shorter calculation time
PLOT_FID <- PLOT_FID[1:30000, ]


## define P.ref as the median of the measured ambient pressures at injection time
## data set contains some NA, because of missing peaks
P.ref <- median(PLOT_FID$air.pressure, na.rm = TRUE)

## define the column names of the dataset
appac.colnames <- list(
  sample.col    = "sample.name",
  peak.col      = "peak.name",
  date.col      = "injection.date",
```

```
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 50)

## appac will throw warnings that sample and peak names have been changed
x <- appac(data = PLOT_FID, P.ref = P.ref,
           appac.colnames = appac.colnames,
           appac.control = appac.control)

## view results
print(paste("Air pressure correction coefficient (kappa):",
      sprintf("%.3e", coefficients(x@correction)), "[1/hPa]"))

## plot the fit
colors <- list(highlight_color="black", lowlight_color="darkgrey",
               line_color="darkblue", fill_color="lightblue")
plotGlobalFit(x, colors)
```

Appac-class *Class* Appac

Description

An object of the S4 class Appac holds the results of the appac function. It contains the slots 'correction' and 'drift', which are the data containers for the respective result elements of the appac function.

Slots

drift: object of class Drift-class

correction: object of class Correction-class

Methods

plotGlobalFit: signature (object = "Appac", colors = "list")

plotLocalFit: signature (object = "Appac", sample = "character", peak = "character", colors = "list")

plotControlChart: signature (object = "Appac", sample = "character", peak = "character", colors = "list")

See Also

appac, Drift-class, Correction-class

Examples

```
## view class
showClass("Appac")
getSlots("Appac")
```

areas — *Raw, pressure corrected and/or drift compensated, expected and residual areas (Matrices)*

Description

The functions `rawAreas`, `correctedAreas`, `compensatedRawAreas`, `compensatedCorrectedAreas`, `expectedAreas`, and `residualAreas` return the respective area values from an `Appac` object.

Usage

```
## S4 method for signature 'Correction,character'
rawAreas(object, sample)
## S4 method for signature 'Correction,character'
correctedAreas(object, sample)
## S4 method for signature 'Drift,character'
compensatedRawAreas(object, sample)
## S4 method for signature 'Correction,character'
compensatedCorrectedAreas(object, sample)
## S4 method for signature 'Correction,character'
expectedAreas(object, sample)
## S4 method for signature 'Correction,character,character'
residualAreas(object, sample,
  type = c("raw.area",
           "corrected.area",
           "compensated.corrected.area"))
```

Arguments

`sample`	a character string representing a sample name
`object`	an `Appac` object
`type`	a character string giving the item to which the residuals refer. This must be (an abbreviation of) one of the strings `"raw.area"`, `"corrected.area"`, or `"compensated.corrected.area"`

Value

a numeric matrix

Note

For a given sample, the number of rows in the output matrix equals the length of the vectors of `dates` and `P`, see `dates`. The columns of the output matrix represent the peaks of a sample.

See Also

`dates`

Examples

```
## missing code
```

`coefficients` *Get the atmospheric pressure correction peak area correction (APPAC) coefficient*

Description

Method `coefficients` provides a means to retrieve the calculated correction coefficient. The standard error of the coefficient can be retrieved by calling the method `stdErrors`, the p-Values of the calculated coefficient by calling `pValues`.

Usage

```
## S4 method for signature 'Correction'
coefficients(object, ...)
## S4 method for signature 'Correction'
stdErrors(object, ...)
## S4 method for signature 'Correction'
pValues(object, ...)
```

Arguments

`object`	a `Correction` object
`...`	currently not in use

Details

The `Correction` object inside an `Appac` object `'appac'` is invoked by `'appac@correction'`.

Value

a numeric value or vector

See Also

`Correction`

Examples

```
## Not run:
library(appac)
library(dplyr)

## PGC_TCD is a short dataset of weekly measurements of 1 control sample taken on
## process analyzer. The data frame needs to be filtered as it contains several
## channels.
data("PGC_TCD")
PGC_TCD <- PGC_TCD %>% filter(channel == "WLD_RL2")

## define P.ref as the median of the measured ambient pressures at injection time
P.ref <- median(PGC_TCD$air.pressure)

## identify the column names of PGC_TCD
```

```
str(PGC_TCD)

## define the column names of interest
appac.colnames <- list(
  sample.col    = "sample.name",
  peak.col      = "peak.name",
  date.col      = "injection.date",
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 5)

## appac will throw warnings that sample and peak names have been changed
x <- suppressWarnings(appac(data = PGC_TCD,
                            P.ref = P.ref,
                            appac.control = appac.control,
                            appac.colnames = appac.colnames))
coefficients(x@correction)
stdErrors(x@correction)
pValues(x@correction)
```

Correction-class *Class* Correction

Description

An object of the S4 class Correction holds the results of the atmosphric pressure correction part of the appac function.

Slots

global.fit: object of class 'list'

local.fits: object of class 'list'

samples: object of class 'list'

Methods

coefficients: signature(object = "Correction", ...)

stdErrors: signature(object = "Correction", ...)

pValues: signature(object = "Correction", ...)

dates: signature(object = "Correction", sample = "character")

P: signature(object = "Correction", sample = "character")

rawAreas: signature(object = "Correction", sample = "character")

correctedAreas: signature(object = "Correction", sample = "character")

compensatedRawAreas: signature(object = "Correction", sample = "character")

compensatedCorrectedAreas: signature(object = "Correction", sample = "character")

expectedAreas: signature(object = "Correction", sample = "character")

residualAreas: signature(object = "Correction", sample = "character", type= c("raw.area", "corrected.area", "compensated.corrected.area"))

```
variance: signature( object = "Correction", sample = "character", type= c("raw.area",
    "corrected.area", "compensated.corrected.area") )
covMatrix: signature( object = "Correction", sample = "character", type= c("raw.area",
    "compensated.corrected.area") )
corMatrix: signature( object = "Correction", sample = "character", type= c("raw.area",
    "compensated.corrected.area") )
```

See Also

Appac, Drift

Examples

```
## view class
showClass("Correction")
```

---

dates *Retrieve dates and measured atmospheric pressures of a sample*

Description

Method dates retrieves the dates at which a sample was measured. Method P retreives the measured atmospheric pressures which prevailed during the sample runs.

Usage

```
## S4 method for signature 'Correction,character'
dates(object, sample)
## S4 method for signature 'Correction,character'
P(object, sample)
```

Arguments

object	a Correction object
sample	a character string representing a sample name

Details

To retreive the Correction object inside the Appac object X, type: X@correction.

Value

a vector of class IDate (date) or a numeric vector (P)

Note

The returned vectors of date and P have the same length for a given sample.

See Also

Correction areas

Examples

```
## missing code
```

`Drift-class` *Class* `Drift`

Description

An oject of the S4 class `Drift` holds the results of the drift compensation part of the `appac` function.

Slots

`drift.factors:` object of class `'data.frame'`

`bias:` object of class `'list'`

`samples:` object of class `'list'`

Methods

`driftFactor:` `signature ( object = "Drift", date = "IDate", area = "matrix" )`

See Also

`Appac`, `Correction`

Examples

```
## view class
showClass("Drift")
```

`driftFactor` *Get drift factors*

Description

Method `driftFactor` provides a method to calculate drift factors.

Usage

```
## S4 method for signature 'Drift,IDate,matrix'
driftFactor(object, date, area)
```

Arguments

`area`	a numeric matrix giving the peak areas for which the drift factor shall be returned. `area` and `date` must either both be of the same length, or one of them must have length 1.
`date`	a vector of class `IDate` which has the same length as `area` or length 1.
`object`	a `Drift-class` object

Details

The drift factor is a function of the corrected peak area. To retrieve the correct drift factors, the argument must be of the type `correctedAreas()`. See Examples.

Value

a numeric matrix

See Also

Drift-class

Examples

```
## Not run:
library(appac)
library(dplyr)

## PGC_TCD is a short dataset of weekly measurements of 1 control sample taken on
## process analyzer. The data frame needs to be filtered as it contains several
## channels.
data("PGC_TCD")
PGC_TCD <- PGC_TCD %>% filter(channel == "WLD_RL2")

## define P.ref as the median of the measured ambient pressures at injection time
P.ref <- median(PGC_TCD$air.pressure)

## identify the column names of interest in PGC_TCD
str(PGC_TCD)

## define the column names of interest
appac.colnames <- list(
  sample.col    = "sample.name",
  peak.col      = "peak.name",
  date.col      = "injection.date",
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 5)

x <- suppressWarnings(appac(data = PGC_TCD,
                            P.ref = P.ref,
                            appac.control = appac.control,
                            appac.colnames = appac.colnames))

## define a sample of interest
s <- "X17k"

## get the dates vector, which applies to sample 's'
d <- dates(sample = s, object = x@correction)

## get the areas of sample 's'
a <- correctedAreas(sample = s, object = x@correction)

## get the drift factors for all peaks of s
```

```
f <- driftFactor (area = a, date = d, object = x@drift)
plot(d, a[, 1] / f[, 1], ylim = c(1329000, 1360200), pch = 19, col = "blue",
     xlab = "date", ylab = "drift corrected area")
points(d, a[, 2] / f[, 2], pch = 19, col = "green")
```

---

packed_FID *Some natural gas components measured a packed column with an FID*

Description

The data set was measured on an Agilent 6890N series gas chromatograph on a channel equipped with a switching valve, a packed pre-column and a packed main column, which was connetcted to an FID. The gaseous samples were injected at a constant injection pressure.

Usage

```
data("packed_FID")
```

Format

A data frame with 28155 observations on the following 9 variables.

`sample.name` the names of the samples

`file.name` a unique identifier of each run

`instrument` the name of the instrument

`channel` the name of the channel

`injection.date` the date at which a run was made

`peak.name` the name of the peak

`retention.time` the measured retention time of the peak

`raw.area` the area as obtained from manual peak integration

`air.pressure` a numeric vector of the measured atmospheric pressures which prevailed during the measurement

Details

The packed columns had been salvaged from an analyzer that was built in the 1980ies nd had been on-duty until the mid 1990ies. The columns, however, exhibited some leaks at the the connections over time.

Source

Gas Quality Competence Centre, Open Grid Europe GmbH, Essen, Germany

Examples

```
data(packed_FID)
str(packed_FID)
```

PGC_TCD *Some natural gas components measured in a TCD*

Description

The data set was obtained from measurements taken with a Siemens Maxum process GC. The short data set contains an extract with weekly measurements (average of 5 measurements) of a single control sample.

Usage

```
data("PGC_TCD")
```

Format

A data frame with 250 observations on the following 9 variables.

`sample.name` the names of the samples

`file.name` a unique identifier for the origin of each data point

`instrument` the name of the instrument

`channel` the name of the channel

`injection.date` the date at which the measurements were taken

`peak.name` the name of the peak

`raw.area` the average area of 5 individual runs

`air.pressure` a numeric vector of the measured atmospheric pressures which prevailed during the measurement series

Details

The sample was injected at atmospheric pressure; the sample amount injected is proportional (increasing) with the pressure. This effect roughly compensates for the negative pressure dependence of the detector, so that only a small positive effect of the ambient pressure on peak area remains.

Note

appac will give an overly optimistic estimate of the drift values when only a single sample is supplied. It is recommended to provide 3 or more control samples whose peak areas are distributed over the entire usable dynamic range of the detector.

Source

Bundesanstalt für Materialforschung und -prüfung, Berlin, Germany

Examples

```
library(appac)

data(PGC_TCD)
str(PGC_TCD)
```

plotControlChart *Plot the raw and fitted area vs. time of a peak of a sample*

Description

The control chart is a plot of the raw and the fitted (pressure corrected and drift compensated) areas vs. time. It is a variant of a Shewart control chart.

Usage

```
## S4 method for signature 'Appac,character,character,list'
plotControlChart(object, sample, peak, colors)
```

Arguments

sample	a character value representing the name of the sample
peak	a character value representing the name of the peak
object	an object of class Appac
colors	a list containing the colors of the color scheme to be used in the plot: **highlight_color:** the color in which the datapoints of the corrected areas are represented **lowlight_color:** the color in which the datapoints of the raw areas are represented **line_color:** the color of the fitted function line **fill_color:** the color in which the standard uncertainties of the fitted function line are plotted

Value

a ggplot object

See Also

plotLocalFit

Examples

```
## Please be patient, the calculation takes a while.

## Not run:
library(appac)

data("packed_FID")

## define P.ref as the median of the measured ambient pressures at injection time
## data set contains some NA, because of missing peaks
P.ref <- median(packed_FID$air.pressure, na.rm = TRUE)

## define the column names of the dataset
appac.colnames <- list(
  sample.col    = "sample.name",
```

```
  peak.col      = "peak.name",
  date.col      = "injection.date",
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 50)

## appac will throw warnings that sample and peak names have been changed
x <- appac(data = packed_FID,
           P.ref = P.ref,
           appac.control = appac.control,
           appac.colnames = appac.colnames)

## View results
s <- "KGM.11D.4"
p <- "C3H8"
colors <- list(highlight_color="black", lowlight_color="darkgrey",
               line_color="darkblue", fill_color="lightblue")

plotControlChart ( object = x, sample = s, peak = p, colors = colors)
```

`plotGlobalFit` *Plot the global (APPAC) fit function*

Description

S4 method to plot an APPAC global fit.

Usage

```
## S4 method for signature 'Appac,list'
plotGlobalFit(object, colors)
```

Arguments

`object`	an object of class `Appac`
`colors`	a list containing the colors of the color scheme to be used in the plot: **highlight_color:** the color in which the datapoints are represented **lowlight_color:** the color in which the errorbars are represented **line_color:** the color of the fitted function line **fill_color:** the color in which the standard uncertainties of the fitted function line are plotted

Details

APPAC fits a function of the form $\hat{Y} = Y_{ref} \times (1 + \kappa(P - P_{ref}))$ to a given data set. P_{ref} is set by the user; the fitted area at the reference pressure Y_{ref} is the expected value of the area of a component in a sample at P_{ref}. Y_{ref} is obtained by local fits of area vs. P. `appac.plot` plots the slope vs the area.ref of the local fits; κ is given by the slope of the fitted line.

Value

a ggplot object

See Also

appac

Examples

```
## Please be patient, the calculation takes a while.

## Not run:
library(appac)

data("packed_FID")

## define P.ref as the median of the measured ambient pressures at injection time
## data set contains some NA, because of missing peaks
P.ref <- median(packed_FID$air.pressure, na.rm = TRUE)

## define the column names of the dataset
appac.colnames <- list(
  sample.col    = "sample.name",
  peak.col      = "peak.name",
  date.col      = "injection.date",
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 50)

## appac will throw warnings that sample and peak names have been changed
x <- suppressWarnings(
       appac(data = packed_FID,
       P.ref = P.ref,
       appac.control = appac.control,
       appac.colnames = appac.colnames)
     )

## View results
colors <- list(highlight_color="black", lowlight_color="darkgrey",
               line_color="darkblue", fill_color="lightblue")
plotGlobalFit(x, colors)
```

---

plotLocalFit *Plot the fit of area vs. pressure of a peak of a sample*

Description

View a plot of the fit of (raw and fitted) area vs. atmospheric pressure of a peak of a sample. appac fits area vs. atmospheric pressure for each peac of each sample. Further, the slopes and the expected values of the areas at the refererence pressure are deployed to obtain the global fit (APPAC) function.

Usage

```
## S4 method for signature 'Appac,character,character,list'
plotLocalFit(object, sample, peak, colors)
```

Arguments

sample	a character value representing the name of the sample
peak	a character value representing the name of the peak
object	an object of class Appac
colors	a list containing the colors of the color scheme to be used in the plot:

highlight_color: the color in which the datapoints of the corrected areas are represented

lowlight_color: the color in which the datapoints of the raw areas are represented

line_color: the color of the fitted function line

fill_color: the color in which the standard uncertainties of the fitted function line are plotted

Value

a ggplot object

See Also

plotControlChart

Examples

```
## Please be patient, the calculation takes a while.

## Not run:
library(appac)

data("packed_FID")

## define P.ref as the median of the measured ambient pressures at injection time
## data set contains some NA, because of missing peaks
P.ref <- median(packed_FID$air.pressure, na.rm = TRUE)

## define the column names of the dataset
appac.colnames <- list(
  sample.col    = "sample.name",
  peak.col      = "peak.name",
  date.col      = "injection.date",
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 50)

## appac will throw warnings that sample and peak names have been changed
x <- appac(data = packed_FID,
           P.ref = P.ref,
```

```
            appac.control = appac.control,
            appac.colnames = appac.colnames)

## View results
s <- "KGM.11D.4"
p <- "C3H8"
colors <- list(highlight_color="black", lowlight_color="darkgrey",
               line_color="darkblue", fill_color="lightblue")

plotLocalFit ( object = x, sample = s, peak = p, colors = colors)
```

PLOT_FID *Some natural gas components measured on a porous layer open tubular (PLOT) column with an FID*

Description

The data set was measured on an Agilent 6890N series gas chromatograph on a channel equipped with a switching valve, a PLOT pre-column and a PLOT main column, which was connetcted to an FID. The gaseous samples were injected at a constant injection pressure.

Usage

```
data("PLOT_FID")
```

Format

A data frame with 47400 observations on the following 9 variables.

`sample.name` the names of the samples

`file.name` a unique identifier of each run

`instrument` the name of the instrument

`channel` the name of the channel

`injection.date` the date at which a run was made

`peak.name` the name of the peak

`retention.time` the measured retention time of the peak

`raw.area` the area as obtained from manual peak integration

`air.pressure` a numeric vector of the measured atmospheric pressures which prevailed during the measurement

Details

GC Method details

oven program: temperature ramp

sample introduction: sample loop at constant sample pressure

column flow: constant flow with pre-column backflush

Source

Gas Quality Competence Centre, Open Grid Europe GmbH, Essen, Germany

Examples

```
library(appac)
data("PLOT_FID")
str(PLOT_FID)
```

PLOT_TCD *Some natural gas components measured on a porous layer open tubular (PLOT) column wwith a TCD*

Description

The data set was measured on an Agilent 6890N series gas chromatograph on a channel equipped with a switching valve, a PLOT pre-column and a PLOT main column, which was connetcted to a TCD. The gaseous samples were injected at a constant injection pressure.

Usage

```
data("PLOT_TCD")
```

Format

A data frame with 77365 observations on the following 9 variables.

`sample.name` the names of the samples

`file.name` a unique identifier of each run

`instrument` the name of the instrument

`channel` the name of the channel

`injection.date` the date at which a run was made

`peak.name` the name of the peak

`retention.time` the measured retention time of the peak

`raw.area` the area as obtained from manual peak integration

`air.pressure` a numeric vector of the measured atmospheric pressures which prevailed during the measurement

Details

GC Method details

oven program: temperature ramp

sample introduction: sample loop at constant sample pressure

column flow: constant flow with pre-column backflush

Source

Gas Quality Competence Centre, Open Grid Europe GmbH, Essen, Germany

Examples

```
data(PLOT_TCD)
str(PLOT_TCD)
```

variances *Variance, variance-covariance and correlation matrices of the peak areas of a sample*

Description

Calculate the variances, the variance-covariance matrix or the correlation matrix of the raw (`type = "raw.area"`) or fitted (`type = "compensated.corrected.area"`) peak areas.

Usage

```
## S4 method for signature 'Correction,character,character'
variance(object, sample, type)
## S4 method for signature 'Correction,character,character'
covMatrix(object, sample, type)
## S4 method for signature 'Correction,character,character'
corMatrix(object, sample, type)
```

Arguments

`sample`	a character string representing a sample name
`object`	a `Correction-class` object
`type`	a character string giving the item to which the residuals refer. This must be (an abbreviation of) one of the strings `"raw.area"`, or `"compensated.corrected.area"`

Details

...

Value

a numeric vector or matrix

Examples

```
## Not run:
library(appac)
library(dplyr)

## PGC_TCD is a short dataset of weekly measurements of 1 control sample taken on
## process analyzer. The data frame needs to be filtered as it contains several
## channels.
data("PGC_TCD")
PGC_TCD <- PGC_TCD %>% filter(channel == "WLD_RL2")

## define P.ref as the median of the measured ambient pressures at injection time
P.ref <- median(PGC_TCD$air.pressure)

## identify the column names of PGC_TCD
str(PGC_TCD)

## define the column names of interest
```

```
appac.colnames <- list(
  sample.col    = "sample.name",
  peak.col      = "peak.name",
  date.col      = "injection.date",
  pressure.col  = "air.pressure",
  area.col      = "raw.area"
)

appac.control <- list(min.data.points = 5)

## appac will throw warnings that sample and peak names have been changed
x <- suppressWarnings(appac(data = PGC_TCD,
                            P.ref = P.ref,
                            appac.control = appac.control,
                            appac.colnames = appac.colnames))

## define the sample name

s <- "X17k"
p <- "n.Butan"

variance(x@correction, sample = s, type = "raw")
variance(x@correction, sample = s, type = "comp")

corMatrix(x@correction, sample = s, type = "raw")
corMatrix(x@correction, sample = s, type = "comp")

covMatrix(x@correction, sample = s, type = "raw")
covMatrix(x@correction, sample = s, type = "comp")
```

Index