

Package ‘qcrLscR’

July 22, 2025

Type Package

Title Quality Control–based Robust LOESS Signal Correction

Version 0.1.3

Description An R implementation of quality control–based robust LOESS(local polynomial regression fitting) signal correction for metabolomics data analysis, described in Dunn, W., Broadhurst, D., Begley, P. et al. (2011) <[doi:10.1038/nprot.2011.335](https://doi.org/10.1038/nprot.2011.335)>. The optimisation of LOESS's span parameter using generalized cross-validation (GCV) is provided as an option. In addition to signal correction, 'qcrLscR' includes some utility functions like batch shifting and data filtering.

License GPL (>= 2)

Depends R (>= 3.0.0)

URL <https://github.com/wanchanglin/qcrLscR>

BugReports <https://github.com/wanchanglin/qcrLscR/issues>

Encoding UTF-8

RoxygenNote 7.3.2

LazyLoad yes

LazyData yes

ZipData No

NeedsCompilation no

Author Wanchang Lin [aut, cre],
Warwick Dunn [aut]

Maintainer Wanchang Lin <wanchanglin@hotmail.com>

Repository CRAN

Date/Publication 2025-04-22 13:20:02 UTC

Contents

batch.shift	2
man_qc	3
mv.filter	4
mv.filter.qc	5
mv.perc	6
outl.det.u	6
qc.rlsc	7
qc.rlsc.wrap	8
Index	11

batch.shift	<i>Batch shifting</i>
-------------	-----------------------

Description

Remove batch effect withing each block.

Usage

```
batch.shift(x, y, method = "mean", overall_average = TRUE)
```

Arguments

- x a data matrix.
- y a categorical data for batch/block information.
- method method for shifting.
- overall_average a logical value to indicate whether or not an overall average will be added after shifting.

Value

a shifted data matrix.

References

Silvia Wagner, et.al, Tools in Metabonomics: An Integrated Validation Approach for LC-MS Metabolic Profiling of Mercapturic Acids in Human Urine Anal. Chem., 2007, 79 (7), pp 2918-2926, DOI: 10.1021/ac062153w

Examples

```
names(man_qc)
data <- man_qc$data
meta <- man_qc$meta
## batch shifting
cls.bl <- factor(meta$batch)
res <- batch.shift(data, cls.bl, overall_average = TRUE)
```

man_qc	<i>man_qc: test data for QC-RLSC</i>
--------	--------------------------------------

Description

This HPLC data set includes 4 batches with missing values.

Usage

```
man_qc
```

Format

A list with data matrix and meta data:

data A data frame with 462 replicates (row) and 656 features (column)

meta A data frame with 2 columns:

- batch: 4 batches
- sample_type: QC and Sample

Examples

```
man_qc
t(sapply(man_qc, dim))
## Select data matrix and meta data
data <- man_qc$data
meta <- man_qc$meta
## Select batches and data types
cls.qc <- factor(meta$sample_type)
cls.bl <- factor(meta$batch)
```

`mv.filter`*Filtering variable based on the percentage of missing values*

Description

This function calculates the percentage of missing values and keeps those features with missing values percentage less than the designed threshold.

Usage

```
mv.filter(x, thres = 0.3)
```

Arguments

<code>x</code>	a data matrix. The columns are features.
<code>thres</code>	threshold of missing values. Features less than this threshold will be kept. Value has to be between 0 and 1.

Value

a list of with contents:

- `dat` the filtered data matrix
- `idx` a logical vector of index for keeping features.

See Also

Other missing value processing: [mv.filter.qc\(\)](#), [mv.perc\(\)](#)

Examples

```
names(man_qc)
data <- man_qc$data
meta <- man_qc$meta
## check missing value rates
tail(sort(mv.perc(data)), 20)
## missing values filtering
tmp <- mv.filter(data, thres = 0.15)
data_f <- tmp$dat
## compare
dim(data_f)
dim(data)
```

mv.filter.qc*Data filtering based on "qc" missing values*

Description

Data filtering based on "qc" missing values

Usage

```
mv.filter.qc(x, y, thres = 0.3)
```

Arguments

x	a data matrix.
y	a character string with contents of "sample", "qc" and "blank".
thres	threshold of missing values. Features less than this threshold will be kept.

Value

a list of with contents:

- dat the filtered data matrix
- idx a logical vector of index for keeping features.

See Also

Other missing value processing: [mv.filter\(\)](#), [mv.perc\(\)](#)

Examples

```
names(man_qc)
data <- man_qc$data
meta <- man_qc$meta
## check missing value rates
tail(sort(mv.perc(data)), 20)
## missing values filtering based on QC
cls.qc <- factor(meta$sample_type)
tmp <- mv.filter.qc(data, cls.qc, thres = 0.15)
data_f <- tmp$dat
## compare
dim(data_f)
dim(data)
```

mv.perc	<i>Missing value percentage</i>
---------	---------------------------------

Description

Calculate missing value percentage.

Usage

```
mv.perc(x)
```

Arguments

x an vector, matrix or data frame.

Value

missing value percentage.

See Also

Other missing value processing: [mv.filter\(\)](#), [mv.filter.qc\(\)](#)

Examples

```
names(man_qc)
data <- man_qc$data
meta <- man_qc$meta
## check missing value rates
tail(sort(mv.perc(data)), 20)
```

outl.det.u	<i>Univariate outlier detection</i>
------------	-------------------------------------

Description

Perform outlier detection using univariate method.

Usage

```
outl.det.u(x, method = c("percentile", "median"))
```

Arguments

x a numeric vector.

method method for univariate outlier detection. Only percentile and median are supported.

Details

- median: the absolute difference between the observation and the sample median is larger than 2 times of the Median Absolute Deviation divided by 0.6745.
- percentile: either smaller than the 1st quartile minus 1.5 times of IQR, or larger than the 3rd quartile plus 1.5 times of IQR.

Value

a logical vector.

References

Wilcox R R, Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy, Springer 2010 (2nd edition), pages 31-35.

Examples

```
x <- c(2, 3, 4, 5, 6, 7, NA, 9, 50, 50)
outl.det.u(x, "percentile")
```

qc.rlsc

QC based robust LOESS signal correction (QC-RLSC)

Description

QC based robust LOESS (locally estimated scatterplot smoothing) signal correction (QC-RLSC)

Usage

```
qc.rlsc(x, y, method = c("subtract", "divide"), opti = TRUE, ...)
```

Arguments

x	A data frame with samples (row) and variables (column).
y	A vector with string of "qc" and "sample".
method	Data scaling method.
opti	A logical value indicating whether or not optimise 'span'
...	Other parameter for 'loess'.

Details

This function includes only information of sample types (QC or Sample) for signal correction. It does not require batch information. User may use batch elimination routine such as `batch.shift()` in this package or others to remove batch effects after signal correction.

If data matrix has missing values, user should filter the data based on missing values percentage. No missing values imputation is needed.

An option is also provided to optimise LOESS's span in a range between 0.05 to 0.95. The R codes are modified from <https://bit.ly/3zBo3Qn>.

Value

A corrected data frame.

References

Dunn et al. Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. Nature Protocols 6, 1060–1083 (2011)

See Also

Other QC-RLSC function: [qc.rlsc.wrap\(\)](#)

Examples

```
names(man_qc)
data <- man_qc$data
meta <- man_qc$meta

cls.qc <- factor(meta$sample_type)
cls.bl <- factor(meta$batch)

## apply QC-RLSC with optimisation of 'span'
res_1 <- qc.rlsc(data, cls.qc, method = "subtract", opti = TRUE)

## apply QC-RLSC without optimisation of 'span'
res_2 <- qc.rlsc(data, cls.qc, method = "subtract", opti = FALSE)
```

qc.rlsc.wrap

Wrapper function for QC-RLSC

Description

Wrapper function for QC-RLSC

Usage

```
qc.rlsc.wrap(
  dat,
  cls.qc,
  cls.bl,
  method = c("subtract", "divide"),
  intra = FALSE,
  opti = TRUE,
  log10 = TRUE,
  outl = TRUE,
```

```

    shift = TRUE,
    ...
)

```

Arguments

<code>dat</code>	A data frame with samples (row) and variables (column).
<code>cls.qc</code>	A vector with string of "qc" and "sample".
<code>cls.bl</code>	A vector with string of batch indicators.
<code>method</code>	Data scaling method. Support "subtract" and "divide"
<code>intra</code>	A logical value indicating whether signal correction is performed inside each batch ("intra-batch") or not ("inter-batch").
<code>opti</code>	A logical value indicating whether or not 'span' parameters are optimised.
<code>log10</code>	A logical value indicating whether log10 transformation for the data set or not. If the transformation is applied, the reverse procedure will be performed.
<code>outl</code>	A logical value indicating whether or not QC outlier detection is employed. If TRUE, the QC outlier will be assigned as the median of QC.
<code>shift</code>	A logical value indicating whether or not batch shift is applied after signal correction.
<code>...</code>	Other parameter for 'loess'.

Value

A corrected data frame.

See Also

Other QC-RLSC function: [qc.rlsc\(\)](#)

Examples

```

names(man_qc)
data <- man_qc$data
meta <- man_qc$meta

cls.qc <- factor(meta$sample_type)
cls.bl <- factor(meta$batch)

## apply QC-RLSC wrapper function
method <- "divide"      # "subtract"
intra <- TRUE
opti <- TRUE
log10 <- TRUE
outl <- TRUE
shift <- TRUE

res <- qc.rlsc.wrap(data, cls.qc, cls.bl, method, intra, opti, log10,

```

out1, shift)

Index

- * **QC-RLSC function**

- qc.rlsc, [7](#)

- qc.rlsc.wrap, [8](#)

- * **datasets**

- man_qc, [3](#)

- * **missing value processing**

- mv.filter, [4](#)

- mv.filter.qc, [5](#)

- mv.perc, [6](#)

batch.shift, [2](#)

man_qc, [3](#)

mv.filter, [4](#), [5](#), [6](#)

mv.filter.qc, [4](#), [5](#), [6](#)

mv.perc, [4](#), [5](#), [6](#)

outl.det.u, [6](#)

qc.rlsc, [7](#), [9](#)

qc.rlsc.wrap, [8](#), [8](#)