Package 'RDML'

September 10, 2014

Type Package

Title Importing real-time thermo cycler (qPCR) data from RDML format files

Version 0.4-2

LazyData true

Date 2014-09-10

Description Imports real-time thermo cycler (qPCR) data from RDML format files and transforms to the appropriate formats of the 'qpcR' and 'chipPCR' packages.

License GPL (>= 2)

URL https://github.com/kablag/RDML

Depends XML (>= 3.98-1.1)

Imports chipPCR (>= 0.0.8-1)

Suggests qpcR (>= 1.3-9)

Author Konstantin A. Blagodatskikh [cre, aut],Stefan Roediger [aut],Michal Burdukiewicz [aut]

Maintainer Konstantin A. Blagodatskikh <k.blag@yandex.ru>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-10 11:38:16

R topics documented:

RDML *qPCR data import function*

Description

Imports qPCR data from RDML v1.1 format files (Lefever et al. 2009) and transforms it to the appropriate format of the qpcR (Ritz et al. 2008, Spiess et al. 2008) and chiPCR packages. RDML is the recommended file format element in the Minimum Information for Publication of Quantitative Real-Time PCR Experiments (MIQE) guidelines (Bustin et al. 2009). Data can be imported to the one table format (flat) or list of tables spited by PCR targets and samples types (see 'Details'). Tables consist of 'Cycles' in the first column and fluorescence data of the samples in the remaining columns for qPCR data. For melting data tables consist of 'Temperature' in the first column and fluorescence data of the samples in the remaining columns. Names of the samples can be formed by patterns (see 'Details').

Usage

```
RDML(rdmlfile = NA, name.pattern = "%NAME%__%TUBE%", flat.table = FALSE,
     omit.ntp = TRUE)
```

Arguments

rdmlfile	RDML file with qPCR data.
name.pattern	sample name pattern (see 'Details').
flat.table	logical. If TRUE, a result table is not spited by PCR targets and samples types.
omit.ntp	logical. If TRUE, samples with 'ntp' type (mark for empty sample in data files generated by *LightCycler 96* software) are omitted.

Details

ATTENTION: Although the format RDML claimed as data exchange format, the specific implementation of the format at devices from different manufacturers differ significantly. Currently this function is checked against RDML data from devices: *Bio-Rad CFX96*, *Roche LightCycler 96* and *Applied Biosystems StepOne*.

This function has been designed to import qPCR data from instruments that support RDML v1.1 format export. Output from this function can be easily used with qpcR package.

There are two general variants of output for this function:

- simplified (flat.table = TRUE). Represents three elements list: vector of named dilutions – 'Dilutions', data.frame with qPCR fluorescence data of **all** samples – 'qPCR' and data.frame with melting fluorescence data of **all** samples – 'Melt'.
- splitted (flat.table = FALSE). Represents three elements list: vector of named dilutions – 'Dilutions', list of data.frames which contain qPCR fluorescence data of samples spited by targets and then by types – 'qPCR' and list of data.frames which contain melting fluorescence data of samples spited by targets and then by types – 'Melt'

Names of the samples can be generated by rules described in `name.pattern`. For example, if `name.pattern = "%NAME%__%TUBE%"`, sample name (inputted in the qPCR software) is "Sample_1", and position on the plate is "A4", then generated sample name will be: "Sample_1__A4".

Possible keys in `name.pattern`

- `%NAME%` - name of the sample inputted in the qPCR software (ex.: "Sample 1")
- `%ID%` - tube ID (ex.: "23")
- `%TUBE%` - tube name (ex.: "B11")
- `%TARGET%` - PCR target (ex.: "GAPDH")
- `%TYPE%` - type of the sample (ex.: "unkn")

Value

A list of `data.frame` (or one `data.frame`), suitable for analysis with `qpcR` package. Also output list contains information about dilutions (quantity or concentration) if available.

Author(s)

Konstantin A. Blagodatskikh <k.blag@yandex.ru>, Stefan Roediger <stefan.roediger@hs-lausitz.de>, Michal Burdukiewicz <michalburdukiewicz@gmail.com>

References

RDML format http://www.rdml.org/
`qpcR` package http://cran.r-project.org/web/packages/qpcR/index.html

`chipPCR` package: http://cran.r-project.org/web/packages/chipPCR/index.html

Ritz, C., Spiess, A.-N., 2008. qpcR: an R package for sigmoidal model selection in quantitative real-time polymerase chain reaction analysis. *Bioinformatics* 24, 1549–1551. doi:10.1093/bioinformatics/btn227

Spiess, A.-N., Feig, C., Ritz, C., 2008. Highly accurate sigmoidal fitting of real-time PCR data by introducing a parameter for asymmetry. *BMC Bioinformatics* 9, 221. doi:10.1186/1471-2105-9-221

Bustin, S.A., Benes, V., Garson, J.A., Hellemans, J., Huggett, J., Kubista, M., Mueller, R., Nolan, T., Pfaffl, M.W., Shipley, G.L., Vandesompele, J., Wittwer, C.T., 2009. The MIQE guidelines: minimum information for publication of quantitative real-time PCR experiments. *Clin. Chem.* 55, 611–622. doi:10.1373/clinchem.2008.112797

Lefever, S., Hellemans, J., Pattyn, F., Przybylski, D.R., Taylor, C., Geurts, R., Untergasser, A., Vandesompele, J., RDML consortium, 2009. RDML: structured language and reporting guidelines for real-time quantitative PCR data. *Nucleic Acids Res.* 37, 2065–2069. doi:10.1093/nar/gkp056

Examples

```
## EXAMPLE 1:
## internal dataset lc96_bACTXY.rdml (in 'data' directory)
## generated by Roche LightCycler 96. Contains qPCR data
## with four targets and two types.
## Import with default settings.
PATH <- path.package("RDML")
```

```
filename <- paste(PATH, "/extdata/", "lc96_bACTXY.rdml", sep ="")
lc96 <- RDML(filename)

## Show targets names
names(lc96$qPCR)
## Show types of the samples for target 'FAM@bACT'
names(lc96$qPCR[["FAM@bACT"]])

## Show dilutions for dye - FAM
lc96$Dilutions$FAM
## Not run:
COPIES <- unique(lc96$Dilutions$FAM["quant",])
## Define calibration curves (type of the samples - 'std').
## No replicates.
library(qpcR)
CAL <- modlist(lc96$qPCR[["FAM@bACT"]]$std,
               fluo = c(2, 4, 6, 8, 10))
## Define samples to predict (first two samples with the type - 'unkn').
PRED <- modlist(lc96$qPCR[["FAM@bACT"]]$unkn[1:5],
                fluo = grep("^S", names(lc96$qPCR[["FAM@bACT"]]$unkn)[1:2]))
## Conduct quantification.
calib(refcurve = CAL, predcurve = PRED, thresh = "cpD2",
      dil = COPIES)

## End(Not run)
## Not run:
## EXAMPLE 2:
## internal dataset lc96_bACTXY.rdml (in 'data' directory)
## generated by Roche LightCycler 96. Contains qPCR data
## with four targets and two types.
## Import with default settings.
library(chipPCR)
PATH <- path.package("RDML")
filename <- paste(PATH, "/extdata/", "lc96_bACTXY.rdml", sep ="")
lc96 <- RDML(filename)

## Compactly display the structure of the lc96 object
str(lc96)
## Fetch cycle dependent fluorescence for HEX chanel
tmp <- lc96[["qPCR"]][["Hex@X"]][["std"]]
## Fetch vector of dillutions for HEX chanel
dilution <- as.vector(lc96[["Dilutions"]][["Hex"]])

## Use plotCurves function from the chiPCR package to
## get an overview of the amplification curves
plotCurves(tmp[, 1], tmp[, -1])
par(mfrow = c(1,1))
## Use inder function from the chiPCR package to
## calculate the Cq (second derivative maximum, SDM)
SDMout <- sapply(2L:ncol(tmp), function(i) {
  SDM <- summary(inder(tmp[, 1], tmp[, i]), print = FALSE)[2]
})
```

```
## Use the effcalc function from the chipPCR package and
## plot the results for the calculation of the amplification
## efficiency analysis.
plot(effcalc(dilution, SDMout), CI = TRUE)

## End(Not run)
## Not run:
## EXAMPLE 3:
## internal dataset BioRad_qPCR_melt.rdml (in 'data' directory)
## generated by Bio-Rad CFX96. Contains qPCR and melting data.
## Import without splitting by targets/types and with
## custom name pattern.
PATH <- path.package("RDML")
filename <- paste(PATH, "/extdata/", "BioRad_qPCR_melt.rdml", sep ="")
cfx96 <- RDML(filename, flat.table=TRUE,
                        name.pattern = "%TUBE%_%NAME%_%TYPE%_%TARGET%")
## Use plotCurves function from the chiPCR package to
## get an overview of the amplification curves
library(chipPCR)
plotCurves(cfx96$qPCR[, 1], cfx96$qPCR[, -1], type = "l")

## Show some generated names for samples.
names(cfx96$Melt[2L:5])
## Select index numbers of the columns that contain
## samples with dye 'EvaGreen' and have type 'pos'.
cols <- grep("pos_EvaGreen$", names(cfx96$Melt))
## Conduct melting curve analysis.
library(qpcR)
invisible(meltcurve(cfx96$Melt, fluos = cols,
          temps = rep(1, length(cols))))

## End(Not run)
```

`summary-RDML_object` *Summary RDML_objects*

Description

Prints and silently returns summary statistics of `RDML_objects`.

Usage

```
## S3 method for class 'RDML_object'
summary(object, print = TRUE, ...)
```

Arguments

`object`	an object of class `RDML_object`.
`print`	logical, if `TRUE` summary is printed to console.
`...`	currently ignored.

Value

A list of length at most 3 (less if `object` does not contain information about dilutions, qPCR or melting data):

`dilTable`	A list of dilutions used in experiments.
`meltTable`	Melting data - currently not implemented.
`expTable`	Summary statistics describing experiments: mean, median, standard deviation, median absolute deviation, interquartile range, medcouple, skewness, SNR, VRM, number of NAs (missing values), intercept, slope, R squared, Breusch-Pagan test p-value.

Author(s)

Michal Burdukiewicz, Konstantin A. Blagodatskikh, Stefan Roediger

See Also

Summary statistics are calculated using `MFIaggr`.

Examples

```
lc96 <- RDML(system.file("extdata/lc96_bACTXY.rdml", package = "RDML"))
summary(lc96)
#store raw results of summary without printing
res <- summary(lc96, print = FALSE)
```

Index