Package 'chipPCR'

September 29, 2014

Type Package

Title Toolkit of helper functions to pre-process amplification data

Version 0.0.8-3

LazyData true

Date 2014-09-29

Description The chipPCR package is a toolkit of functions to pre-process amplification curve data. Amplification data can be obtained from conventional PCR reactions or isothermal amplification reactions. The package contains functions to normalize and baseline amplification curves,a routine to detect the start of an amplification reaction, several smoothers for amplification data, a function to distinguish positive and negative amplification reactions and a function to determine the amplification efficiency. The smoothers are based on LOWESS, moving average, cubic splines, Savitzky-Golay and others. In addition the first approximate approximate derivative maximum (FDM) and second approximate derivative maximum (SDM) can be calculated by a 5-point-stencil as quantification points from real-time amplification curves. chipPCR contains data sets of experimental nucleic acid amplification systems including the VideoScan HCU and a capillary convective PCR (ccPCR) system. The amplification data were generated by helicase dependent amplification (HDA) or polymerase chain reaction (PCR) under various temperature conditions. As detection system intercalating dyes (EvaGreen, SYBR Green) and hydrolysis probes (TaqMan) were used. The latest source code is available via: https://github.com/michbur/chipPCR

License GPL-3

URL https://github.com/michbur/chipPCR

Depends R (>= 3.0.0), methods

Suggests drc, knitr, markdown, MBmca (>= 0.0.3-4), qpcR, RDML, xtable

VignetteBuilder knitr

Imports lmtest, MASS, outliers, ptw, quantreg, Rfit, robustbase,shiny, signal

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-29 07:26:55

Author Stefan Roediger [cre, aut],Michal Burdukiewicz [aut]

Maintainer Stefan Roediger <stefan.roediger@hs-lausitz.de>

R topics documented:

`chipPCR-package` *Toolkit of functions to pre-process amplification data*

Description

The `chipPCR` package is a toolkit of functions to pre-process amplification curve data. Amplification data can be obtained from conventional PCR reactions or isothermal amplification reactions. The package contains functions to normalize and baseline amplification curves, a routine to detect the start of an amplification reaction, several smoothers for amplification data, a function to distinguish positive and negative amplification reactions and a function to determine the amplification efficiency. The smoothers are based on LOWESS, moving average, cubic splines, Savitzky-Golay and others. In addition the first approximate approximate derivative maximum (FDM) and second approximate derivative maximum (SDM) can be calculated by a 5-point-stencil as quantification points from real-time amplification curves. `chipPCR` contains data sets of experimental nucleic acid amplification systems including the VideoScan HCU and a capillary convective PCR (ccPCR) system. The amplification data were generated by helicase dependent amplification (HDA) or polymerase chain reaction (PCR) under various temperature conditions. As detection system intercalating dyes (EvaGreen, SYBR Green) and hydrolysis probes (TaqMan) were used. The latest source code is available via: https://github.com/michbur/chipPCR

Details

Package:	chipPCR
Type:	Package
Version:	0.0.8-3
Date:	2014-09-29
License:	GPL-3

The function `bg.max` can be used to remove missing values in amplification curve data. The function `amptester` is used to test if an amplification is positive. `fixNA` is used to impute missing values from a data column. `CPP` can be used to normalize curve data, to remove background, to remove outliers and further steps. The package includes further functions to smooth the data by different functions including LOWESS, Moving Average, Friedman's SuperSmoother, Cubic Spline and Savitzky-Golay smoothing.

Author(s)

Stefan Roediger, Michal Burdukiewicz

Maintainer: Stefan Roediger <stefan.roediger@hs-lausitz.de>

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Nucleic acid detection based on the use of microbeads: a review. S. Roediger, C. Liebsch, C. Schmidt, W. Lehmann, U. Resch-Genger, U. Schedler, P. Schierack. *Microchim Acta* 2014:1–18. DOI: 10.1007/s00604-014-1243-4

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

See Also

qpcR.news.

Examples

```
# Example: A simple function to test for a background range.
# Data were taken form the chipPCR C17 data set.
data(C17)
plot(C17[, 2], C17[,  3], xlab = "time [min]", ylab = "Fluorescence",
      pch = 20)
res <- bg.max(C17[, 2], C17[, 3], bg.corr = 1.4, bg.start = 3)
abline(v = c(slot(res, "bg.start"), slot(res, "bg.stop")), col = c(1,2))
abline(h = slot(res, "fluo"), col = "blue")
```

AmpSim *Amplification curve simulator*

Description

This function is a simple simulator of an amplification reaction based on a 5-parameter Richards function. This simple approach was chosen because it is impossible to model the shape of any amplification curve. An implementation of realistic models is ambitious and not conclusively addressed in the literature. First, they have to take "all" random effects of noise into consideration and second, they need to be generic enough to cover all amplification processes. More sophisticated mechanistic models and simulations have been proposed elsewhere *mehra_2005*, *cobbs_2012*. This approach of `AmpSim` is similar to the `pcrsim` function from the *qpcR* package, which offers simulations of sigmoidal qPCR data with goodness-of-fit analysis by Ritz and Spiess 2008.

Usage

```
AmpSim(cyc = 1:35, b.eff = -25, bl = 0.05, ampl = 1, Cq = 20,
noise = FALSE, nnl = 0.025, nnl.method = "constant")
```

Arguments

`cyc`	is a vector containing the cycle values.
`b.eff`	can be used to adjust the amplification efficiency.
`bl`	is used to define the base level (minimum) of the background range.
`ampl`	defines the plateau (maximum) of the amplification reaction.
`Cq`	defines approximately the quantification point (Cq) of the amplification reaction.
`noise`	adds some noise to the amplification reaction.
`nnl`	level of noise during the amplification reaction.
`nnl.method`	trend of noise level during the amplification reaction. "constant" uses same noise of amplification, "decreasing" leads to less noise at the end of the amplification reaction, and "increasing" leads to more noise at the end of the amplification reaction.

Details

The function `AmpSim` is a simple simulator for amplification reaction. The function has several parameters which can be used to simulate the amplification curve. `b.eff` and `Cq` are most connected with another. Thus changing one of them will change both values. `Cq` can be used to define an approximate Cq value. The expression "approximate Cq value" is used here because the actual Cq value is dependent on the users preferred method (e.g., Cy0 method, Second Derivative Maximum (SDM) method, threshold method). The funtion can be used to see how an experimental system compares to a predicted model. Moreover it can be used to simulate data with noise, missing values (NA), signal-to-noise ratios, photo-bleaching and other influences on a PCR reaction.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Ritz, C., Spiess, A.-N.: qpcR: an R package for sigmoidal model selection in quantitative real-time polymerase chain reaction analysis. *Bioinformatics* 24(13), 1549–1551 (2008). doi:10.1093/bioinformatics/btn227. PMID: 18482995.

See also qpcR.news

Examples

```
# Example one
# Simulate a qPCR reaction with AmpSim for 40 cycles.
# Use an in-silico dilution of the template be adjusting
# the Cq parameter. A change of 3.32 cycles corresponds
# approximately to a 10-fold dilution.
par(mfrow = c(2,1))
plot(NA, NA, xlim = c(1,40), ylim = c(0.01,2), xlab = "Cycles",
     ylab = "Fluorescence", main = "In-silco Dilution Experiment")
cycle.dilution <- seq(18, 35, 3.32)
for (i in 1:6) {
  lines(AmpSim(cyc = 1:40, b.eff = -25, bl = 0.01, ampl = 2,
      Cq = cycle.dilution[i]), type = "b", col = i, pch = 20)
}

# Example two
# Simulate a qPCR reaction with AmpSim for 40 cycles and some noise.
plot(NA, NA, xlim = c(1,40), ylim = c(0.01,2.2), xlab = "Cycles",
     ylab = "Fluorescence",
     main = "In-silco Dilution Experiment with Some Noise")
cycle.dilution <- seq(18, 35, 3.32)
for (i in 1:6) {
  lines(AmpSim(cyc = 1:40, b.eff = -25, bl = 0.01, ampl = 2,
      Cq = cycle.dilution[i], noise = TRUE, nnl = 0.05),
      type = "b", col = i, pch = 20)
}
par(mfrow = c(1,1))

# Example three
# Apply constant, increasing, decreasing nose to
# amplification data.

par(mfrow = c(3,1))
method <- c("constant", "increase", "decrease")
for (j in 1:3){
```

```
    plot(NA, NA, xlim = c(1,40), ylim = c(0.02,2.2), xlab = "Cycles",
ylab = "Fluorescence",
main = paste("In-silco Dilution Experiment with noise method: ",
    method[j]))
    cycle.dilution <- seq(18, 35, 3.32)
    for (i in 1:6) {
      lines(AmpSim(cyc = 1:40, b.eff = -25, bl = 0.02, ampl = 2,
   Cq = cycle.dilution[i], noise = TRUE, nnl = 0.08,
   nnl.method = method[j]), type = "b", col = i, pch = 20)
   }
}
par(mfrow = c(1,1))
```

AmpSim.gui *Amplification Curve Simulation Graphical User Interface*

Description

Launches graphical user interface that allows simulating and analyzing amplification reactions. The function will open the GUI in a webpage of the default browser. All parameters of the AmpSim function can be used. In addition shows the GUI some information calculated by the bg.max in a summary field and a plot below the simulated amplification curve.

Usage

```
AmpSim.gui()
```

Warning

Any ad-blocking software may be cause of malfunctions.

Author(s)

Stefan Roediger, Michal Burdukiewicz.

See Also

AmpSim, bg.max

Examples

```
# Run from a R console following commands
require(shiny)

# Invoke the shiny AmpSim app in the default browser
runApp(paste0(find.package("chipPCR")[1],"/AmpSim.gui"))
```

amptest *Class* "amptest"

Description

An S4 class containing the output amptester function.

Slots

.Data: "numeric" is a vector containing the fluorescence values.

decisions: "logical" contains outcomes of various tests. shap.noisy is presence of noise, lrt.test states if data are likely from a amplification curve and both tht.dec and tht.dec defines if the amplification is "positive" or "negative".

noiselevel: "numeric" user-defined threshold for a significant amplification signal.

background: range of the background signal in the amplification curve.

polygon: The pco test determines if the points in an amplification curve (like a polygon) are in a "clockwise" order. The sum over the edges result in a positive value if the amplification curve is "clockwise" and is negative if the curve is counter-clockwise.

slope.ratio: ratio of the slopes at the start and the end of exponential phase..

Methods

summary signature(object = "amptest"): prints summary of the object. Silently returns vector of all calculated parameters.

show signature(object = "amptest"): prints only .Data slot of the object.

plot signature(object = "amptest"): plots input data and graphical interpretation of link{amptester} tests' results.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

amptester

Examples

```
# Compare a positive and a negative amplification reaction.
# First simulate positive reaction (fluo.pos) and than the
# negative reaction (fluo.neg).
# Simulation of an amplifiaction curve with some noise and a high signal.

fluo.pos <- AmpSim(cyc = 1:40, noise = TRUE)[, 2]
ampt.pos <- amptester(fluo.pos, manual = TRUE, background = c(0, 15),
      noiselevel = 0.15)
```

```
# Plot amplification curve and result of amptester
plot(fluo.pos, xlab = "Cycles", ylab = "RFU", pch = 19, ylim = c(0, 1))
lines(ampt.pos, col = 2, pch = 19, type = "b")
legend(5, 1, c("Raw data", "amptester output"),
       col = c(1,2,3), bty = "n", pch = c(19, 19))
# Borders for background calculation
abline(v = c(0,15), col = 2)
# Level for background threshold
abline(h = 0.15, col = 3, lty = 2)
text(5, 0.18, "Noise threshold")
# Summary of amptester results
summary(ampt.pos)

# Simulation of an amplifiaction curve with high noise and a low signal.

fluo.neg <- AmpSim(cyc = 1:40, noise = TRUE, ampl = 0.13, nnl = 0.4)[, 2]
ampt.neg <- amptester(fluo.neg, manual = TRUE, background = c(0, 15),
      noiselevel = 0.15)

# Plot amplification curve and result of amptester
plot(fluo.neg, xlab = "Cycles", ylab = "RFU", pch = 19, ylim = c(0, 1))
lines(ampt.neg, col = 2, pch = 19, type = "b")
legend(5, 1, c("Raw data", "amptester output"),
       col = c(1,2,3), bty = "n", pch = c(19, 19))
# Borders for background calculation
abline(v = c(0,15), col = 2)
# Level for background threshold
abline(h = 0.15, col = 3, lty = 2)
text(5, 0.18, "Noise threshold")
# Summary of amptester results
summary(ampt.neg)
#plot amptester results
plot(ampt.neg)
```

`amptester` *Amplification test*

Description

The function `amptester` can be used to test if an amplification is significant.

Usage

```
amptester(y, manual = FALSE, noiselevel = 0.08, background = NULL)
```

Arguments

y	is a vector containing the fluorescence values.

manual	switches between a statistical test (based on a Wilcoxon rank sum test (wilcox.test)) or manual test for a positive amplification signal.
noiselevel	can be set to a user defined value as threshold for a significant amplification signal.
background	is the range of the background signal in the amplification curve. The values can be added by the user or taken from the bg.max function. Ignored if manual is TRUE.

Details

The function tries to estimate if a amplification process is taking place. Several instances of tests are included. The first involves a semiautomatic test if the range of the background is lower than the range of the assumed signal. To differ between the ranges an instance of bg.max is used. Herein, the function assumes that an amplification takes place in case the signal of the amplification is larger than the median + 5 * mad than the background. The automatic test uses a Wilcoxon rank sum test wilcox.test to compare the first and the last elements of the data. The input values are delivered by head and tail, respectively. For other methods please refer to the references listed below. Instead of assigning a zero to negative amplification reaction uses the current implementation of amptester very small random values. This is because some post function might fail in case all values are set to zero.

Value

An object of amptest class containing result of the test as well as the original data.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Frank, D. N. BARCRAWL and BARTAB: software tools for the design and implementation of barcoded primers for highly multiplexed DNA sequencing *BMC bioinformatics*, 2009, Vol. 10, pp. 362

Peirson, S. N., Butler, J. N. and Foster, R. G. Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis *Nucleic Acids Research*, 2003, Vol. 31(14), pp. e73-e73

Rao, X., Lai, D. and Huang, X. A New Method for Quantitative Real-Time Polymerase Chain Reaction Data Analysis *Journal of Computational Biology*, 2013, Vol. 20(9), pp. 703-711

Ruijter, J. M., Ramakers, C., Hoogaars, W. M. H., Karlen, Y., Bakker, O., Hoff, M. J. B. v. d. and Moorman, A. F. M. Amplification efficiency: linking baseline and bias in the analysis of quantitative PCR data, *Nucleic Acids Research*, 2009, Vol. 37(6), pp. e45-e45

Rutledge, R. G. and Stewart, D. A kinetic-based sigmoidal model for the polymerase chain reaction and its application to high-capacity absolute quantitative real-time PCR *BMC biotechnology*, 2008, Vol. 8, pp. 47

Tichopad, A., Dilger, M., Schwarz, G. and Pfaffl, M. W. Standardized determination of real-time PCR efficiency from a single reaction set-up *Nucleic Acids Research*, 2003, Vol. 31(20), pp. e122

Wilhelm, J., Pingoud, A. and Hahn, M. SoFAR: software for fully automatic evaluation of real-time PCR data *BioTechniques*, 2003, Vol. 34(2), pp. 324-332

Zhao, S. and Fernald, R. D. Comprehensive Algorithm for Quantitative Real-Time Polymerase Chain Reaction *Journal of computational biology: a journal of computational molecular cell biology*, 2005, Vol. 12(8), pp. 1047-1064

Examples

```
# First example
# Arrange graphs in orthogonal matrix and set parameter for the plot.
par(las = 0, bty = "n", cex.axis = 1.5, cex.lab = 1.5,
    font = 2, cex.main = 1.8, oma = c(1,1,1,1))

# Simulation of an amplification curve with 40 cycles and a Cq of
# circa 28. The amplification curve of "pos" (positive) has low
# noise and the amplification curve of "neg" (negative) has high
# noise.

pos <- AmpSim(cyc = 1:40, Cq = 28, noise = TRUE, nnl = 0.03)
neg <- AmpSim(cyc = 1:40, Cq = 28, noise = TRUE, nnl = 0.8)

# Plot the raw data of the simulations.

par(fig = c(0,0.5,0.5,1))
plot(NA, NA, xlim = c(1, 40), ylim = c(0, 2.1), xlab = "Cycles",
     ylab = "Fluorescence", main = "qPCR - Raw data", type = "b")
mtext("A", cex = 2, side = 3, adj = 0, font = 2)
points(pos, col = 1, typ = "b", pch = 19)
points(neg, col = 2, typ = "b", pch = 20)
legend(1, 2, c("Positive", "Negative Control (noise)"),
       pch = c(19,20), col = c(1,2), lwd = 2, bty = "n")

# Plot data again after an analysis by ampteser. "neg" is set to small
# random numbers, while "pos" remains unchanged.

par(fig = c(0,0.5,0,0.5), new = TRUE)
plot(NA, NA, xlim = c(1, 40), ylim = c(0, 2.1), xlab = "Cycles",
     ylab = "Fluorescence", main = "qPCR - amptester", type = "b")
points(amptester(pos[, 2]), col = 1, type = "b", pch = 19)
points(amptester(neg[, 2]), col = 2, type = "b", pch = 20)
legend(1, 2, c("Positive", "Negative Control (noise)"),
       pch = c(19,20), col = c(1,2), lwd = 2, bty = "n")

# Use of amptester for time-dependent measurements. Amplification curves
# from the capillaryPCR data set were processed in a loop. The results of
# amptester are added to the raw data.

par(fig = c(0.5,1,0,1), new = TRUE)
colors <- rainbow(8)
plot(NA, NA, xlim = c(0,80), ylim = c(0,1300), xlab = "Time [min]",
     ylab = "Voltage (micro V)", main = "ccPCR")
mtext("B", cex = 2, side = 3, adj = 0, font = 2)
```

```
sapply(c(1,3,5,7), function(i) {
    xy.tmp <- cbind(capillaryPCR[1:750, i], capillaryPCR[1:750, i + 1])

# Use amptester to analyse the amplification curve.
# Note: The decisions of amptester can be invoked via res.ampt@decisions
# in the present example.

    res.ampt <- amptester(xy.tmp[, 2])

# Use the "decisions" of amptester in a logic to automatically decide if an
# amplification reaction is positive. In this example linear regression test
# (lrt.test) and the threshold test (tht.dec) are used.

    res.ampt <- ifelse(res.ampt@decisions[2] == TRUE &&
       res.ampt@decisions[4] == TRUE, "positve", "negative")

# Plot the amplification curve with the decisions.
    lines(xy.tmp[, 1], xy.tmp[, 2], type = "b", pch = 20, col = colors[i])
    text(75, max(na.omit(xy.tmp[, 2])), res.ampt, cex = 1.3, col = colors[i])
  }
)
# Second Example
# Example to test an amplification reaction.
# Simulate first a positive amplification curve with 45 cycles and than a
# negative amplification curve with 45 cycles. The negative amplification
# curve is created from a normal distribution
#
fluo.neg <- rnorm(45)
fluo.pos <- AmpSim(cyc = 1:45, Cq = 45, ampl = 40, noise = TRUE,
   nnl = 0.03)[, 2]

plot(NA, NA, xlim = c(1, 45), ylim = c(-1, 45), xlab = "Cycles",
     ylab = "Fluorescence",
     main = "Simulation of a qPCR with 45 Cycles", type = "b")
points(amptester(fluo.pos), type = "b", pch = 20)
points(amptester(fluo.neg), type = "b", col = "red", pch = 20)
points(1:45, fluo.neg, col = "red")

legend(1,40, c("Positive", "Negative Control (noise)",
       "noise pattern"), pch = c(20,20,1), col = c(1,2,2), lwd = 2)
```

`amptester.gui` *Amplification Test Graphical User Interface*

Description

`amptester.gui` is a graphical user interface for the `amptester` function. This function can be used for a fast and convenient analysis of amplification curve data. In addition it is possible to analyze the Cq (quantification cycle) and to perform a report generation of the analyzed data.

Usage

```
amptester.gui()
```

Warning

Any ad-blocking software may be cause of malfunctions.

Author(s)

Stefan Roediger, Michal Burdukiewicz.

See Also

AmpSim, bg.max

Examples

```
# Run from a R console following commands
require(shiny)

# Invoke the shiny AmpSim app in the default browser
runApp(paste0(find.package("chipPCR")[1],"/amptester.gui"))
```

---

bg *Class* "bg"

Description

An S4 class containing the output bg.max function.

Slots

.Data: "matrix" which columns represent respectively cycle number, raw fluorescence data, first derivative and second derivative.

bg.start: "numeric" value representing start of the background range.

bg.stop: "numeric" value representing end of the background range.

bg.corr: "numeric" a value which helps to tweak on the suggested background value of bg.max.

fluo: "numeric" a value of fluorescence at the end of amplification.

amp.stop: "numeric" value representing end of the amplification .

Methods

plot signature(x = "bg"): plots background information. See plot.bg

show signature(object = "bg"): prints only .Data slot of the object.

summary signature(object = "bg"): prints information about object prettier than show and allows easy access to some slots. See summary.bg

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

bg.max, plot.bg, summary.bg

Examples

```
res <- AmpSim(cyc = 1:40, Cq = 25)
tmp <- bg.max(res)
summary(tmp)
plot(tmp)
```

---

bg.max *Simple function to detect and correct the background range*

Description

The function bg.max is intended to detect and correct background noise. The detection is made without any assumptions regarding the model of the function.

Usage

```
## S4 method for signature 'numeric,numeric'
bg.max(x, y, bg.corr = 1.3, bg.start = 2,
  inder.approx = TRUE)

## S4 method for signature 'matrix,missing'
bg.max(x, y, bg.corr = 1.3, bg.start = 2,
  inder.approx = TRUE)

## S4 method for signature 'data.frame,missing'
bg.max(x, y, bg.corr = 1.3, bg.start = 2,
  inder.approx = TRUE)
```

Arguments

x	is a vector containing the time or cycle values or data frame/matrix containing cycle in the first column and fluorescence values in the second column.
y	is a vector containing the fluorescence values. Used only if x is also a vector.
bg.corr	a value which helps to tweak on the suggested background value of bg.max.
bg.start	a user defined value for the start of the background range.
inder.approx	a logical value defining if data should be numerically derived by inder function. If FALSE, derivatives are calculated by the predict.smooth.spline.

Details

Background range herein refers to a level of fluorescence measured before any specific amplification is detectable. The raw data (e.g., fluorescence intensity) measured after each step (cycle or time point) follow a non-linear progress. The background is assumed to be constant for the entire measurement. The algorithm of `bg.max` is based on the assumption that the signal difference of successive cycles in the linear ground phase is approximately constant. After transition in the early exponential phase the signal changes drastically. First data are smoothed by Friedman's 'super smoother' (as found in `supsmu`. Thereof the approximate first and second derivative are calculated by a five-point stencil `link[chipPCR]{inder}`.

The difference of cycles at the maxima of the first and second approximate derivative and a correction factor are used to estimate the range before the exponential phase. This simple function finds the background range without modeling the function. the start of the background range is defined be a fixed value. since many signals tend to overshot in the first cycles a default value of 3 is chosen. `bg.max` tries also to estimate the end of an amplification reaction.

– The following paragraphs describe methods to detect the background range of amplification curves. Currently none of them is implemented as R function.

The easiest way to classify them is the extend of assumptions made before applying of a method. The simplest approach is to treat the background fluorescence as a value constant during whole amplification reaction. In this case the noise could be approximated as the mean or median of fluorescence values in lag phase (Frank (2009)) or their standard deviations (Peirson et al. (2003)). The more sophisticated way of approximating constant background fluorescence requires optimizing its value to achieve linearity of the model fit on the semi logarithmic plot in log-linear phase (Frank (2009)). The latter procedure is greatly enhanced by performing further computations only on a subset of consecutive measurements for which calculated efficiencies have the lowest variance. Other methods loosen the assumption that background fluorescence is a constant value and instead describe it as a function of the cycle number. For example the algorithm used in SoFar (Wilhelm et al. (2003)) fits a nonlinear saturation function to measurement points before the start of the exponential growth phase. Parameters of the saturation function are chosen to minimize the sum of squared residuals of the fitted function. Then the value of saturation function is calculated for all data points and subtracted from measured values giving corrected values of fluorescence, which are used in next calculations.

Some approaches make even less assumptions regarding the form of the background noise. The taking-difference linear regression method has a premise that changes of fluorescence between subsequent cycles are exclusively caused by the amplification of the product (Rao et al. (2013)). The corrected values are calculated by simply subtracting the fluorescence value in the former cycle from fluorescence in the latter. Of course in this case the real fluorescence value in first cycle is unknown, so the number of cycles that can be used in following computations is reduced by one.

The Real-Time PCR Miner algorithm is also nearly assumption-free (Zhao (2005)). The main principle is that background fluorescence is similar in the small groups of subsequent measurements. So the first step of the algorithm is division of subsequent measurement points belonging to the exponential phase of amplification in at least four-element groups. For each set of points is calculated a pair of the estimate of the efficiency and the significance of model representing relation between the fluorescence value and the cycle number. The estimates paired with the highest significance are the most influential in the computation of the final efficiency.

To find the beginning of the lag phase and end of plateau phase is important for the goodness-of-fit for both exponential-phase-only and S-shaped models. There are two strategies. The first

narrows the area of the search to the neighborhood of their theoretical values determined by a fitted model of the amplification reaction. To this group belongs SoFar (Wilhelm et al. (2003)). The algorithm looks for the start and the end of the exponential phase near the second derivatives of the function representing the relation between logarithm of the fluorescence and the cycle number. The available correction guarantees that the start of amplification has higher value than background noise. The very similar procedure is implemented in Real-Time PCR Miner (Zhao (2005)), where background noise is also used as parameter in implemented models to calculate theoretical the start of the amplification process. The end of amplification process is detected by calculating the third derivative of implemented S-shaped model. The second approach does not require theoretical values. A very intuitive solution, designated take-off point, by Tichopad et al. (2003) describes the lag phase using a linear function. Random deviations are taken into account as standardized residuals. The method starts with a fitting of a linear function to first three measurement points. If none of residuals is considered an outlier with a statistical test, the algorithm fits a new linear model to the first four measurement points and so on. The procedure stops when two last points are designated as outliers. The first of aforementioned outliers is considered the end of lag phase. It is worth noting that this algorithm is versatile enough to also detect the beginning of the plateau phase.

Value

An object of bg class containing predicted background range as well as other parameters.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

D. N. Frank. BARCRAWL and BARTAB: software tools for the design and implementation of barcoded primers for highly multiplexed DNA sequencing. *BMC Bioinformatics*, 10:362, 2009. ISSN 1471-2105. doi: 10.1186/1471-2105-10-362. PMID: 19874596 PMCID: PMC2777893.

S. N. Peirson, J. N. Butler, and R. G. Foster. Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis. *Nucleic Acids Research*, 31(14):e73, July 2003. ISSN 1362-4962. PMID: 12853650 PMCID: PMC167648.

X. Rao, D. Lai, and X. Huang. A new method for quantitative real-time polymerase chain reaction data analysis. *Journal of computational biology: a journal of computational molecular cell biology*, 20(9):703–711, Sept. 2013. ISSN 1557-8666. doi: 10.1089/cmb.2012.0279. PMID: 23841653 PMCID: PMC3762066.

A. Tichopad, M. Dilger, G. Schwarz, and M. W. Pfaffl. Standardized determination of real-time PCR efficiency from a single reaction set-up. *Nucleic Acids Research*, 31(20):e122, Oct. 2003. ISSN 1362-4962. PMID: 14530455 PMCID: PMC219490.

J. Wilhelm, A. Pingoud, and M. Hahn. Real-time PCR-based method for the estimation of genome sizes. *Nucleic Acids Research*, 31(10):e56, May 2003. ISSN 0305-1048. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC156059/. PMID: 12736322 PMCID: PMC156059.

S. Zhao and R. D. Fernald. Comprehensive algorithm for quantitative real-time polymerase chain reaction. *Journal of computational biology: a journal of computational molecular cell biology*, 12(8): 1047–1064, Oct. 2005. ISSN 1066-5277. doi: 10.1089/cmb.2005.12.1047. PMID: 16241897 PMCID: PMC2716216.

Examples

```
# First example: Test for the background of an amplification reaction.
par(mfrow = c(2,1))
res <- AmpSim(cyc = 1:40, Cq = 25)
background <- bg.max(res)
plot(background, main = "Estimation of the Background Range\n
in Absence of Noise")
res.noise <- AmpSim(cyc = 1:40, Cq = 25, noise = TRUE)
background.noise <- bg.max(res.noise)
plot(background.noise, main = "Estimation of the Background Range\n
in Presence of Noise")
par(mfrow = c(1,1))


# Second example: A simple function to test for a background range.
# Data were taken form the chipPCR C17 data set.
# Note that the not the time but the "cycle number" was
# used to calculate the background range.
data(C17)
plot(C17[, 2], C17[,  3], xlab = "Cycle", ylab = "RFU",
     main = "Estimate the begin of the Amplification\n of a HDA",
     pch = 20)
res <- bg.max(C17[, 2:3], bg.corr = 1.4, bg.start = 1)
abline(v = c(slot(res, "bg.start"), slot(res, "bg.stop")),
       col = c(1,2))
abline(h = slot(res, "fluo"), col = "blue")

# Third example: Test for the background of an amplification reaction.
# Simulate amplification curves with different quantification points
# within 40 cycles.
cyc <- seq(1, 40, 1)

# Use a five parameter model to simulate amplification curves
b <- -15; c <- 0.02; d <- 1
# Define the different quantification points with a difference of
# circa 3.32 cycles
e <- seq(21, 35, 3.32)

# Plot the amplification curves and the estimated background ranges.
plot(NA, NA, xlim = c(1, 40), ylim = c(0, 1), xlab = "Cycles",
     ylab = "Fluorescence")

for (i in 1:length(e)) {
  fluo <- c + (d - c)/(1 + exp(b * (log(cyc) - log(e[i]))))
  points(cyc, fluo, type = "b", col = i, pch = 20)
  res <- bg.max(cyc, fluo, bg.corr = 1.4, bg.start = 1)
  abline(v = slot(res, "bg.stop"), col = i)
  abline(h = slot(res, "fluo"), col = i)
}
```

C126EG595 *qPCR Experiment for the Amplification of HPRT1 Using the Bio-Rad iQ5 thermo cycler*

Description

A Quantitive PCR (qPCR) with the DNA binding dye (EvaGreen) (Mao et al. 2007) was performed in the Bio-Rad iQ5 thermo cycler. The cycle-dependent increase of the fluorescence was quantified at the annealing step (59.5 deg Celsius).

Usage

```
data(C126EG595)
```

Format

A data frame with 40 observations on the following 97 variables. The first column ("Cycle") contains the number of cylces and consecutive columns contain the replicates ("A01" to "H12").

Details

HPRT1 was amplified in the Bio-Rad iQ5. The the change of fluorescence was simultaniously monitored for the Hydrolysis probe of HPRT1 and EvaGreen. The primer sequences for HPRT1 were taken from Roediger et al. (2013). A 10 micro L qPCR reaction was composed of 250 nM primer (forward and reverse), qPCR Mix (accordingt to the manufactures recommendations), 1 micro L template (HPRT1 amplification product), 60 nM hydorlysis probe probe for HPRT1. EvaGreen was used at 0.5x final. During the amplification was monitored 59.5 degree Celsius.

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Mao, F., Leung, W.-Y., Xin, X., 2007. Characterization of EvaGreen and the implication of its physicochemical properties for qPCR applications. *BMC Biotechnol*. 7, 76.

Examples

```
data(C126EG595)
tmp  <- C126EG595

plot(NA,NA, xlim = c(1,40), ylim = c(min(tmp[, 2:ncol(tmp)]),
   max(tmp[, 2:ncol(tmp)])), xlab = "Cycle", ylab = "RFU (FAM)",
   main = "Amplification monitored at \n58.5 degree Celsius (annealing step)")
apply(tmp[, 2:ncol(tmp)], 2,
      function(x) lines(tmp[1:nrow(tmp),1],x))
```

C126EG685 *qPCR Experiment for the Amplification of HPRT1 Using the Roche Ligth Cycler 1.5*

Description

A Quantitive PCR (qPCR) with the DNA binding dye (EvaGreen) (Mao et al. 2007) was performed in the Roche Ligth Cycler 1.5 thermo cycler. The cycle-dependent increase of the fluorescence was quantified at the elongation step (68.5 deg Celsius).

Usage

```
data(C126EG685)
```

Format

A data frame with 40 observations on the following 97 variables. The first column ("Cycles") contains the number of cylces and consecutive columns contain the replicates ("A01" to "H12").

Details

MLC-2v was amplified in the Roche Ligth Cycler 1.5. The the change of fluorescence was simultaniously monitored for the Hydrolysis probe of MLC-2v and EvaGreen. The primer sequences for MLC-2v were taken from Roediger et al. (2013). A 10 micro L qPCR reaction was composed of 250 nM primer (forward and reverse), qPCR Mix (accordingt to the manufactures recommendations), 1 micro L template (MLC-2v amplification product), 60 nM hydorlysis probe probe for MLC-2v. EvaGreen was used at 0.5x final. The amplification was monitored at 68.5 degree Celsius (elongation step).

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Mao, F., Leung, W.-Y., Xin, X., 2007. Characterization of EvaGreen and the implication of its physicochemical properties for qPCR applications. *BMC Biotechnol*. 7, 76.

Examples

```
data(C126EG685)
tmp <- C126EG685

plot(NA,NA, xlim = c(1,40), ylim = c(min(tmp[, 2:ncol(tmp)]),
    max(tmp[, 2:ncol(tmp)])), xlab = "Cycle",
    ylab = "RFU (FAM)",
    main = "Amplification monitored at \n68.5 degree Celsius (elongation
step)")

apply(tmp[, 2:ncol(tmp)], 2,
      function(x) lines(tmp[1:nrow(tmp),1],x))
```

C127EGHP *qPCR Experiment for the Amplification of MLC-2v Using the Roche Light Cycler 1.5*

Description

Quantitive PCR (qPCR) with a hydrolysis probe (Cy5/BHQ2) and DNA binding dye (EvaGreen) (Mao et al. 2007) was performed in the Roche Light Cycler 1.5 thermo cycler. The cycle-dependent increase of the fluorescence was quantified at the annealing step.

Usage

```
data(C127EGHP)
```

Format

A data frame with 40 observations on the following 66 variables.

`index` Index of sample

`Cycle` Cycles

`EG1` Eva Green

`EG2` Eva Green

`EG3` Eva Green

EG4 Eva Green
EG5 Eva Green
EG6 Eva Green
EG7 Eva Green
EG8 Eva Green
EG9 Eva Green
EG10 Eva Green
EG11 Eva Green
EG12 Eva Green
EG13 Eva Green
EG14 Eva Green
EG15 Eva Green
EG16 Eva Green
EG17 Eva Green
EG18 Eva Green
EG19 Eva Green
EG20 Eva Green
EG21 Eva Green
EG22 Eva Green
EG23 Eva Green
EG24 Eva Green
EG25 Eva Green
EG26 Eva Green
EG27 Eva Green
EG28 Eva Green
EG29 Eva Green
EG30 Eva Green
EG31 Eva Green
EG32 Eva Green
HP1 Hydrolysis probe
HP2 Hydrolysis probe
HP3 Hydrolysis probe
HP4 Hydrolysis probe
HP5 Hydrolysis probe
HP6 Hydrolysis probe
HP7 Hydrolysis probe
HP8 Hydrolysis probe

HP9 Hydrolysis probe

HP10 Hydrolysis probe

HP11 Hydrolysis probe

HP12 Hydrolysis probe

HP13 Hydrolysis probe

HP14 Hydrolysis probe

HP15 Hydrolysis probe

HP16 Hydrolysis probe

HP17 Hydrolysis probe

HP18 Hydrolysis probe

HP19 Hydrolysis probe

HP20 Hydrolysis probe

HP21 Hydrolysis probe

HP22 Hydrolysis probe

HP23 Hydrolysis probe

HP24 Hydrolysis probe

HP25 Hydrolysis probe

HP26 Hydrolysis probe

HP27 Hydrolysis probe

HP28 Hydrolysis probe

HP29 Hydrolysis probe

HP30 Hydrolysis probe

HP31 Hydrolysis probe

HP32 Hydrolysis probe

Details

MLC-2v was amplified in the Roche Ligth Cycler 1.5. The the change of fluorescence was simultaniously monitored for the Hydrolysis probe of MLC-2v and EvaGreen. The primer sequences for MLC-2v were taken from Roediger et al. (2013). A 10 micro L qPCR reaction was composed of 250 nM primer (forward and reverse), qPCR Mix (accordingt to the manufactures recommendations), 1 micro L template (MLC-2v amplification product), 60 nM hydorlysis probe probe for MLC-2v. EvaGreen was used at 0.5x final. During the amplification was monitored 59.5 degree Celsius.

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Mao, F., Leung, W.-Y., Xin, X., 2007. Characterization of EvaGreen and the implication of its physicochemical properties for qPCR applications. *BMC Biotechnol.* 7, 76.

Examples

```
str(C127EGHP)
data(C127EGHP)
tmp <- C127EGHP

par(mfrow = c(2,1))
plot(NA, NA, xlim = c(1,40), ylim = c(0,10), xlab = "Cycle",
      ylab = "Fluorescence", main = "MLC-2v qPCR - EvaGreen")
  for (i in 3:34) {
    points(tmp[, 2], tmp[, i], type = "l", col = i)
  }

plot(NA, NA, xlim = c(1,40), ylim = c(0,10), xlab = "Cycle",
      ylab = "Fluorescence", main = "MLC-2v qPCR - Hydrolysis probe")
  for (i in 35:66) {
    points(tmp[, 2], tmp[, i], type = "l", col = i)
  }
par(mfrow = c(1,1))
```

C17 *Helicase Dependent Amplification of HPRT1 at Different Temperatures using the VideoScan Platform 2.0*

Description

A Helicase Dependent Amplification (HDA) of HPRT1 (Homo sapiens hypoxanthine phosphoribosyltransferase 1) was performed at different temperatures in the VideoScan Platform 2.0 (similar to Roediger et al. (2013)). The HDA was performed at 55, 60 and 65 degree Celsius. The optimal temperature for a HDA is circa 65 degree Celsius. Lower temperatures will affect the slope and plateau of the HDA amplification curve.

Usage

```
data(C17)
```

Format

A data frame with 125 observations on the following 5 variables.

`C17.t` Elapsed time during HDA in seconds.

`C17.cycle` a numeric vector

`C17.T55` Time-dependent fluorescence at 55 degree Celsius

`C17.T60` Time-dependent fluorescence at 60 degree Celsius

`C17.T65` Time-dependent fluorescence at 65 degree Celsius

Details

To perform an isothermal amplification in VideoScan 2.0, standard conditions for the IsoAmp(R) III Universal tHDA Kit (Biohelix) were used. The reaction was composed of 12.5 micro L buffer A containing 1.25 micro L 10x reaction buffer, 150 nM primer (forward and reverse), 0.75 micro L template (synthetic) and A. bidest which was covered with 50 micro L mineral oil. The primer sequences for HPRT1 were taken from Roediger et al. (2013). Preincubation: This mixture was incubated for 2 min at 95 degree. Celsius and immediately placed on ice. 12.5 micro L of reaction buffer B which was composed of 1.25 micro L 10x buffer, 40 mM NaCl, 5 mM MgSO4, 1.75 micro L dNTPs, 0.2 x Evagreen, 1 micro L Enzyme mix and A. bidest. The fluorescence measurement in VideoScan HCU started directly after adding buffer B at 55, 60 or 65 degree Celsius and revealed optimal conditions for the amplification when using 60 or 65 degree Celsius. Temperature profile (after Preincubation): - 60 seconds at 65 degree Celsius - 11 seconds at 55 degree Celsius && Measurement

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(C17)
plot(NA, NA, xlim = c(0,5000), ylim = c(0,1.2), xlab = "Time [sec]",
     ylab = "Fluorescence",
     main = "Temperature dependency of HDA amplification reactions")
  points(C17[, 1], C17[, 3], type = "b", col = 1, pch = 20)
  points(C17[, 1], C17[, 4], type = "b", col = 2, pch = 20)
  points(C17[, 1], C17[, 5], type = "b", col = 3, pch = 20)
legend(2000, 0.4, c("55 deg Celsius", "60 deg Celsius", "65 deg Celsius"),
col = c(1,2,3), pch = rep(20,3))
```

C54 *qPCR Experiment for the amplification of MLC-2v using the VideoScan heating/cooling-unit*

Description

qPCR Experiment for the amplification of MLC-2v using the VideoScan heating/cooling-unit

Usage

```
data(C54)
```

Format

A data frame with 56 observations on the following 4 variables.

`Cycle` Cycle number

`D1` Stock concentration of input cDNA

`D2` 1/10 diluted stock cDNA

`D3` 1/100 diluted stock cDNA

Details

The aim was to amplify MLC-2v in the VideoScan and to monitor with a hydrolysis probe for MLC-2v. The primer sequences for MLC-2v were taken from Roediger et al. (2013). The amplification was detected in solution of the 1 HCU (see Roediger et al. 2013 for details). A 20 micro L PCR reaction was composed of 250 nM primer (forward and reverse), 1x Maxima Probe qPCR Master Mix (Fermentas), 1 micro L template (MLC-2v amplification product in different dilutions), 50 nM hydorlysis probe probe for MLC-2v and A. bidest. During the amplification, fluorescence was measured at 59.5 degree Celsius. The Cy5 channel was used to monitor the MLC-2v specific hydrolysis probe. Input stock cDNA was used undiluted (D1). D2 was 1/1000 and D3 1/1000000 diluted in A. bidest. The D1, D2, and D3 have different numbers measure points and D2 contains a missing value at cycle 37.

Source

Stefan Roediger, Claudia Deutschmann

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(C54)
str(C54)
plot(NA, NA, xlim = c(0,56), ylim = c(0, 0.7), xlab = "Cycle",
     ylab = "refMFI")
apply(C54[, c(2:4)], 2, function(x) lines(C54[, 1], x))
```

`C60.amp` *qPCR Experiment for the Amplification of MLC-2v and Vimentin (as decadic dilutions) Using the Roche Light Cycler 1.5*

Description

Dilution experiment and metling curve (`C60.melt`) for the human genes MLC-2v and Vimentin (see Roediger et al. 2013) using the Roch Light Cycler 1.5.

Usage

```
data(C60.amp)
```

Format

A data frame with 45 observations on the following 33 variables.

`Index` Index of Cycles

`Vim.0.1` Vimentin water control

`Vim.0.2` Vimentin water control

`Vim.1.1` Vimentin 10^-3 diluted

`Vim.1.2` Vimentin 10^-3 diluted

`Vim.2.1` Vimentin 10^-4 diluted

`Vim.2.2` Vimentin 10^-4 diluted

`Vim.3.1` Vimentin 10^-5 diluted

`Vim.3.2` Vimentin 10^-5 diluted

`Vim.4.1` Vimentin 10^-6 diluted

`Vim.4.2` Vimentin 10^-6 diluted

`Vim.5.1` Vimentin 10^-7 diluted

`Vim.5.2` Vimentin 10^-7 diluted

`Vim.6.1` Vimentin 10^-8 diluted

`Vim.6.2` Vimentin 10^-8 diluted

`Vim.7.1` Vimentin 10^-9 diluted

`Vim.7.2` Vimentin 10^-9 diluted

`MLC2v.1.1` MLC-2v 10^-3 diluted

`MLC2v.1.2` MLC-2v 10^-3 diluted

`MLC2v.2.1` MLC-2v 10^-4 diluted

`MLC2v.2.2` MLC-2v 10^-4 diluted

`MLC2v.3.1` MLC-2v 10^-5 diluted

`MLC2v.3.2` MLC-2v 10^-5 diluted

`MLC2v.4.1` MLC-2v 10^-6 diluted

`MLC2v.4.2` MLC-2v 10^-6 diluted

`MLC2v.5.1` MLC-2v 10^-7 diluted

`MLC2v.5.2` MLC-2v 10^-7 diluted

`MLC2v.6.1` MLC-2v 10^-8 diluted

`MLC2v.6.2` MLC-2v 10^-8 diluted

`MLC2v.7.1` MLC-2v 10^-9 diluted

`MLC2v.7.2` MLC-2v 10^-9 diluted

`MLC2v.0.1` MLC-2v water control

`MLC2v.0.2` MLC-2v water control

Details

MLC-2v and Vimentin were amplified in the Roche Ligth Cycler 1.5. Decadic dilutions of the input cDNA were prepared. The change of fluorescence was simultaniously monitored with EvaGreen. The primer sequences for MLC-2v were taken from Roediger et al. (2013). A 10 micro L qPCR reaction was composed of 250 nM primer (forward and reverse), Roche qPCR Master-Mix (accordingt to the manufactures recommendations) and 1 micro L input DNA. EvaGreen was used at 1x final. During the amplification was monitored 58 degree Celsius. Temperature profile:

95 deg C for 8 minutes 40 x 95 deg C for 10 sec 58 deg C for 15 sec 69 deg C for 25 sec

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Mao, F., Leung, W.-Y., Xin, X., 2007. Characterization of EvaGreen and the implication of its physicochemical properties for qPCR applications. *BMC Biotechnol*. 7, 76.

Examples

```
## data(C60.amp)
str(C60.amp)
```

C60.melt *Melt Curves MLC-2v and Vimentin for the qPCR Experiment C60.amp Using the Roche Light Cycler 1.5*

Description

Melting curves were continously monitored with the Light Cycler 1.5 (Roche). For details on sample preparation refere to C60.amp.

Usage

```
data(C60.melt)
```

Format

A data frame with 128 observations on the following 65 variables. Refere to C60.amp for details of the specific samples.

`Index` Index of elements

`Vim.0.1.T` Temperature

`Vim.0.1.F` Fluorescence

`Vim.0.2.T` Temperature

`Vim.0.2.F` Fluorescence

`Vim.1.1.T` Temperature

`Vim.1.1.F` Fluorescence

`Vim.1.2.T` Temperature

`Vim.1.2.F` Fluorescence

`Vim.2.1.T` Temperature

`Vim.2.1.F` Fluorescence

`Vim.2.2.T` Temperature

`Vim.2.2.F` Fluorescence

`Vim.3.1.T` Temperature

`Vim.3.1.F` Fluorescence

`Vim.3.2.T` Temperature

`Vim.3.2.F` Fluorescence

`Vim.4.1.T` Temperature

`Vim.4.1.F` Fluorescence

`Vim.4.2.T` Temperature

`Vim.4.2.F` Fluorescence

`Vim.5.1.T` Temperature

`Vim.5.1.F` Fluorescence
`Vim.5.2.T` Temperature
`Vim.5.2.F` Fluorescence
`Vim.6.1.T` Temperature
`Vim.6.1.F` Fluorescence
`Vim.6.2.T` Temperature
`Vim.6.2.F` Fluorescence
`Vim.7.1.T` Temperature
`Vim.7.1.F` Fluorescence
`Vim.7.2.T` Temperature
`Vim.7.2.F` Fluorescence
`MLC2v.1.1.T` Temperature
`MLC2v.1.1.F` Fluorescence
`MLC2v.1.2.T` Temperature
`MLC2v.1.2.F` Fluorescence
`MLC2v.2.1.T` Temperature
`MLC2v.2.1.F` Fluorescence
`MLC2v.2.2.T` Temperature
`MLC2v.2.2.F` Fluorescence
`MLC2v.3.1.T` Temperature
`MLC2v.3.1.F` Fluorescence
`MLC2v.3.2.T` Temperature
`MLC2v.3.2.F` Fluorescence
`MLC2v.4.1.T` Temperature
`MLC2v.4.1.F` Fluorescence
`MLC2v.4.2.T` Temperature
`MLC2v.4.2.F` Fluorescence
`MLC2v.5.1.T` Temperature
`MLC2v.5.1.F` Fluorescence
`MLC2v.5.2.T` Temperature
`MLC2v.5.2.F` Fluorescence
`MLC2v.6.1.T` Temperature
`MLC2v.6.1.F` Fluorescence
`MLC2v.6.2.T` Temperature
`MLC2v.6.2.F` Fluorescence
`MLC2v.7.1.T` Temperature
`MLC2v.7.1.F` Fluorescence

`MLC2v.7.2.T` Temperature

`MLC2v.7.2.F` Fluorescence

`MLC2v.0.1.T` Temperature

`MLC2v.0.1.F` Fluorescence

`MLC2v.0.2.T` Temperature

`MLC2v.0.2.F` Fluorescence

Details

Melting curves were continously monitored with the Light Cycler 1.5 (Roche).

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Mao, F., Leung, W.-Y., Xin, X., 2007. Characterization of EvaGreen and the implication of its physicochemical properties for qPCR applications. *BMC Biotechnol*. 7, 76.

Examples

```
## data(C60.melt)
str(C60.melt)
```

C67 *Helicase Dependent Amplification of HPRT1 with different input DNA quantities using the Bio-Rad iQ5 thermo cycler*

Description

A Helicase Dependent Amplification (HDA) of HPRT1 (Homo sapiens hypoxanthine phosphoribosyltransferase 1) was performed at three different input DNA quantities using the Bio-Rad iQ5 thermo cycler. The HDA was performed at 65 degree Celsius. The optimal temperature for a HDA is circa 65 degree Celsius. Lower temperatures will affect the slope and plateau of the HDA amplification curve.

Usage

```
data(C67)
```

Format

A data frame with 43 observations on the following 6 variables.

`Cycles.C67` a numeric vector containing the cycle numbers

`t.C67` a numeric vector containing the time elapsed between the cycles. The time was calculated by the cycle duration of one iQ5 thermocycler step (71 seconds / step).

`D1` Dilution 1.

`D2` Dilution 2.

`D3` Dilution 3.

`D4` Dilution 4.

Details

To perform an isothermal amplification in VideoScan, standard conditions for the IsoAmp(R) III Universal tHDA Kit (Biohelix) were used. The reaction was composed of 12.5 micro L buffer A containing 1.25 micro L 10x reaction buffer, 150 nM primer (forward and reverse), 0.75 micro L template (synthetic) and A. bidest which was covered with 50 micro L mineral oil. The primer sequences for HPRT1 were taken from Roediger et al. (2013). Preincubation: This mixture was incubated for 2 min at 95 degree. Celsius and immediately placed on ice. 12.5 micro L of reaction buffer B which was composed of 1.25 micro L 10x buffer, 40 mM NaCl, 5 mM MgSO4, 1.75 micro L dNTPs, 0.2 x Evagreen, 1 micro L Enzyme mix and A. bidest. The fluorescence measurement started directly after adding buffer B and the preincubation step. Temperature profile if the iQ5 thermo cycler (after Preincubation): - 60 seconds at 65 degree Celsius - 11 seconds at 55 degree Celsius && Measurement

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(C67)
## maybe str(C67) ; plot(C67) ...
```

C81 *Helicase Dependent Amplification of pCNG1 using the VideoScan Platform*

Description

A Helicase Dependent Amplification (HDA) of pCNG1 was performed. The VideoScan Platform (Roediger et al. (2013)) was used to monitor the amplification. The HDA was performed at 65 degree Celsius. Two concentrations of input DNA were used.

Usage

```
data(C81)
```

Format

A data frame with 351 observations on the following 5 variables.

`Cycle` Cycles HDA measurements.

`t.D1` Dilution 1, elapsed time during HDA in seconds.

`MFI.D1` Dilution 1, fluorescence.

`t.D2` Dilution 2, elapsed time during HDA in seconds.

`MFI.D2` Dilution 2, fluorescence.

Details

To perform an isothermal amplification in VideoScan, standard conditions for the IsoAmp(R) III Universal tHDA Kit (Biohelix) were used. The reaction was composed of reaction mix A)10 micro L A. bidest, 1.25 micro L 10xbuffer, 0.75 micro L primer(150nM final), 0.5 micro L template plasmid. Preincubation: This mixture was incubated for 2 min at 95 degree. Celsius and immediately placed on ice. Reaction mix B) 5 micro L A. bidest., 1.25 micro L 10x buffer, 2 micro L NaCl, 1.25 micro L MgSO4, 1,75 micro L dNTPs, 0.25 micro L EvaGreen, 1 micro L enzyme mix. The mix was covered with 50 micro L mineral oil. The fluorescence measurement in VideoScan HCU started directly after adding buffer B at 65 degree Celsius. A 1x (D1) and a 1:10 dilution (D2) were tested. Temperature profile (after Preincubation): - 60 seconds at 65 degree Celsius - 11 seconds at 55 degree Celsius && Measurement

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(C81)
# First example
# Comparison of Lowess, Moving average and splines to smooth amplification curve
# data of
# HDA for pCNG1.

plot(NA, NA, xlim = c(0, 120), ylim = c(0, 1.2), xlab = "Time [min]",
     ylab = "Fluorescence", main = "VideScan HCU HDA amplification - Raw data")
  points(C81[, 2]/60, C81[, 3], type = "b", col = 1, pch = 20)
  points(C81[, 4]/60, C81[, 5], type = "b", col = 2, pch = 20)
legend(2000, 0.4, c("D1", "D2"), col = c(1,2), pch = rep(20,2))
```

C85 *Helicase Dependent Amplification of Vimentin using the VideoScan Platform*

Description

A Helicase Dependent Amplification (HDA) of Vimentin (Vim) was performed. The VideoScan Platform (Roediger et al. (2013)) was used to monitor the amplification. The HDA was performed at 65 degree Celsius. Three concentrations of input DNA (D1, D2, D3) were used.

Usage

```
data(C85)
```

Format

A data frame with 301 observations on the following 5 variables.

`Cycle` Cycles HDA measurements.

`t.D1` Dilution 1, elapsed time during HDA in seconds.

`MFI.D1` Dilution 1, fluorescence.

`t.D2` Dilution 2, elapsed time during HDA in seconds.

`MFI.D2` Dilution 2, fluorescence.

`t.D3` Dilution 3, elapsed time during HDA in seconds.

`MFI.D3` Dilution 3, fluorescence.

Details

To perform an isothermal amplification in VideoScan, standard conditions for the IsoAmp(R) III Universal tHDA Kit (Biohelix) were used. Primers and templates are described in Roediger et al. (2013). The reaction was composed of reaction mix A)10 micro L A. bidest, 1.25 micro L 10xbuffer, 0.75 micro L primer(150nM final), 0.5 micro L template plasmid. Preincubation: This mixture was incubated for 2 min at 95 degree. Celsius and immediately placed on ice. Reaction mix B) 5 micro L A. bidest., 1.25 micro L 10x buffer, 2 micro L NaCl, 1.25 micro L MgSO4, 1.75 micro L dNTPs, 0.25 micro L EvaGreen, 1 micro L enzyme mix. The mix was covered with 50 micro L mineral oil. The fluorescence measurement in VideoScan HCU started directly after adding buffer B at 65 degree Celsius. A 1x (D1), a 1:10 dilution (D2) and a 1:100 (D3) dilution were tested. Temperature profile (after Preincubation): - 60 seconds at 65 degree Celsius - 11 seconds at 55 degree Celsius && Measurement

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(C85)
# First example
plot(NA, NA, xlim = c(0,85), ylim = c(0,1), xlab = "Time [min]",
     ylab = "Fluorescence", main = "HDA amplification")
  points(C85[, 2]/60, C85[, 3], type = "b", col = 1, pch = 20)
  points(C85[, 4]/60, C85[, 5], type = "b", col = 2, pch = 20)
  points(C85[, 6]/60, C85[, 7], type = "b", col = 3, pch = 20)
legend(40, 0.5, c("D1, 1x", "D2, 1:10", "D3, 1:100"), col = c(1:3),
pch = rep(20,3))

# Second example
plot(NA, NA, xlim = c(0,30), ylim = c(0,0.8), xlab = "Time [min]",
     ylab = "Fluorescence", main = "HDA amplification")
points(C85[, 2]/60, C85[, 3], type = "b", col = 1, pch = 20)
points(C85[, 2]/60, smoother(C85[, 2]/60, C85[, 3],
     method = list("savgol")), type = "b", col = 2, pch = 20)
points(C85[, 2]/60, smoother(C85[, 2]/60, C85[, 3],
     method = list("smooth")), type = "b", col = 3, pch = 20)
points(C85[, 2]/60, smoother(C85[, 2]/60, C85[, 3],
     method = list("mova")), type = "b", col = 4, pch = 20)

legend(1, 0.8, c("D1, raw", "D1, savgol", "D1, smooth", "D1, mova"),
col = c(1:4), pch = rep(20,4))
```

```
# Third example
# Comparison of Lowess, Moving average and splines to smooth amplification
# curve data of
# a HDA using the VideoScan HCU for amplification and monitoring.

xrange <- 2:2400
plot(NA, NA, xlim = c(0,85), ylim = c(0.4,0.8), xlab = "Time [min]",
     ylab = "RFI", main = "Raw data")
  points(C85[, 2]/60, C85[, 3], type = "b", col = 1, pch = 20)
  points(C85[, 4]/60, C85[, 5], type = "b", col = 2, pch = 20)
  points(C85[, 6]/60, C85[, 7], type = "b", col = 3, pch = 20)
legend(40, 0.5, c("D1, 1x", "D2, 1:10", "D3, 1:100"), col = c(1:3),
        pch = rep(20,3))

mtext("A", cex = 2, side = 3, adj = 0, font = 2)

plot(NA, NA, xlim = c(0,40), ylim = c(0.4,0.8), xlab = "Time [min]",
     ylab = "RFI", main = "Moving average")
movaww <- seq(1,17,4)
for (i in 1:length(movaww)) {
  for (j in c(2,4,6)) {
    tmp <- data.frame(na.omit(C85[xrange, j])/60, na.omit(C85[xrange, j+1]))
   tmp.out <- smoother(tmp[, 1], tmp[, 2], method = list(mova = list(movaww = movaww[i])),
                       bg.outliers = TRUE)
    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20, cex = 0.5,
          col = i)
  }
}
mtext("B", cex = 2, side = 3, adj = 0, font = 2)
legend(2, 0.8, c("D1, 1x", "D2, 1:10", "D3, 1:100"), col = c(1:3),
        pch = rep(20,3))
legend(25,0.6, paste("movaww : ", movaww), pch = 20, lwd = 2,
        col = 1:length(movaww))

plot(NA, NA, xlim = c(0,40), ylim = c(0.4,0.8), xlab = "Time [min]",
     ylab = "RFI", main = "Cubic Spline")
df.fact <- seq(0.5,0.9,0.1)
for (i in 1:length(df.fact)) {
  for (j in c(2,4,6)) {
    tmp <- data.frame(na.omit(C85[xrange, j])/60, na.omit(C85[xrange, j+1]))
    tmp.out <- smoother(tmp[, 1], tmp[, 2],
                        method = list(smooth = list(df.fact = df.fact[i])),
                        bg.outliers = TRUE)

            lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20,
                  cex = 0.5, col = i)
  }
}

mtext("C", cex = 2, side = 3, adj = 0, font = 2)
legend(2, 0.8, c("D1, 1x", "D2, 1:10", "D3, 1:100"), col = c(1:3),
        pch = rep(20,3))
legend(25,0.6, paste("df.fact : ", df.fact), pch = 20, lwd = 2,
```

```
        col = 1:length(df.fact))

plot(NA, NA, xlim = c(0,40), ylim = c(0.4,0.8), xlab = "Time [min]",
     ylab = "RFI", main = "Lowess")
f <- seq(0.01,0.2,0.04)
for (i in 1:length(f)) {
  for (j in c(2,4,6)) {
      tmp <- data.frame(na.omit(C85[xrange, j])/60, na.omit(C85[xrange, j+1]))
      tmp.out <- smoother(tmp[, 1], tmp[, 2], method = list(lowess = list(f = f[i])),
                          bg.outliers = TRUE)

      lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20, cex = 0.5,
            col = i)
  }
}

mtext("D", cex = 2, side = 3, adj = 0, font = 2)
legend(2, 0.8, c("D1, 1x", "D2, 1:10", "D3, 1:100"), col = c(1:3),
        pch = rep(20,3))
legend(25,0.6, paste("f : ", f), pch = 20, lwd = 2, col = 1:length(f))


plot(NA, NA, xlim = c(0,40), ylim = c(0.4,0.8), xlab = "Time [min]",
     ylab = "RFI", main = "Friedman's\n''super smoother''")
span <- seq(0.01,0.05,0.01)
for (i in 1:length(span)) {
  for (j in c(2,4,6)) {
    tmp <- data.frame(na.omit(C85[xrange, j])/60, na.omit(C85[xrange, j+1]))
    tmp.out <- smoother(tmp[, 1], tmp[, 2], method = list(supsmu = list(span = span[i])),
                        bg.outliers = TRUE)

    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20, cex = 0.5,
          col = i)
  }
}

mtext("E", cex = 2, side = 3, adj = 0, font = 2)
legend(2, 0.8, c("D1, 1x", "D2, 1:10", "D3, 1:100"), col = c(1:3),
        pch = rep(20,3))
legend(25,0.6, paste("span : ", span), pch = 20, lwd = 2, col = 1:length(span))

plot(NA, NA, xlim = c(0,40), ylim = c(0.4,0.8), xlab = "Time [min]",
     ylab = "RFI", main = "Savitzky-Golay")

for (j in c(2,4,6)) {
    tmp <- data.frame(na.omit(C85[xrange, j])/60, na.omit(C85[xrange, j+1]))
    tmp.out <- smoother(tmp[, 1], tmp[, 2], method = list("savgol"),
 bg.outliers = TRUE)

    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20, cex = 0.5,
  col = 1)
  }
```

```
mtext("F", cex = 2, side = 3, adj = 0, font = 2)
legend(25,0.6, paste("/ : ", NULL), pch = 20, lwd = 2, col = 1:length(span))
```

---

capillaryPCR *capillary convective PCR*

Description

The capillary convective PCR (ccPCR) is a modified device of the ccPCR system proposed by Chou et al. 2011.

Usage

```
data(capillaryPCR)
```

Format

A data frame with 1844 observations on the following 10 variables.

`t.121205` Elapsed time during amplification

`ED.121205` a numeric vector

`t.121128` Elapsed time during amplification

`ED.121128` a numeric vector

`t.121130.1` Elapsed time during amplification

`ED.121130.1` a numeric vector

`t.121130.2` Elapsed time during amplification

`ED.121130.2` a numeric vector

`t.121130.3` Elapsed time during amplification

`ED.121130.3` a numeric vector

Details

Modified version of the capillary convective tube isothermal heater heater by Chou et al. 2011. As heating system a conventional block heat was used. On the top of the heating block, we placed for the uptake of the capillaries an aluminum block (8 mm height) in which four holes (3.2 mm diameter and 3.0 mm depth with round shaped bottom) were drilled. The capillaries are regular 100 micro L Roche LightCycler(R). These glass capillaries have a round shaped closed bottom (2.3 mm inner diameter and 3.2 mm outer diameter). An "ESE-Log" detector (QIAGEN Lake Constance) was used for the real time fluorescent measurements, which was mounted in a distance of 5-10 mm next to the capillary. The PCR was performed with SYBR(R) Green fluorescent intercalating dye. Thereof the ESE-Log has in one channel the excitation at 470 nm and the detection at 520 nm. The data was recorded by the FL Digital Software (QIAGEN Lake Constance) and the exported text based raw data.

Source

Ralf Himmelreich, IMM, Mainz, Germany

References

Chou, W., Chen, P., Miao Jr, M., Kuo, L., Yeh, S. and Chen, P. (2011). Rapid DNA amplification in a capillary tube by natural convection with a single isothermal heater. Biotech. 50, 52-57.

Examples

```
# First example
data(capillaryPCR)
plot(NA, NA, xlim = c(0,80), ylim = c(0,1300), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR - Raw Data")

for (i in c(1,3,5,7)) {
  lines(capillaryPCR[, i], capillaryPCR[, i+1], type = "b", pch = 20)
}

abline(h = 290, v = c(18, 23, 35))
legend(60,800, c("Run 1", "Run 2", "Run 3", "Control"), pch = 20, lwd = 2)

# Second example
par(mfrow = c(2,1))

type <- c("mova", "spline", "savgol")
plot(NA, NA, xlim = c(0,80), ylim = c(0,1100), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR with mova,
     spline and savgol")
for (i in 1:3) {
  for (j in c(1,3,5,7)) {
      tmp <- data.frame(na.omit(capillaryPCR[, j]),
na.omit(capillaryPCR[, j+1]))
      tmp.sm <- smoother(tmp[, 1], tmp[, 2], method = list(type[i]))
      lines(data.frame(tmp[, 1], tmp.sm), type = "b", pch = 20, cex = 0.5,
    col = i)
  }
}

abline(h = 200, v = c(17.5, 21.3, 32.9))
legend(0, 1000, c("mova", "spline", "savgol"), pch = 20, lwd = 2,
col = c(1:3))

plot(NA, NA, xlim = c(10,40), ylim = c(50,300), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR with mova,
     spline and savgol")
for (i in 1:3) {
  for (j in c(1,3,5,7)) {
      tmp <- data.frame(na.omit(capillaryPCR[, j]),
na.omit(capillaryPCR[, j+1]))
      tmp.sm <- smoother(tmp[, 1], tmp[, 2], method = list(type[i]))
      lines(data.frame(tmp[, 1], tmp.sm), type = "b", pch = 20, cex = 0.5,
```

```
    col = i)
  }
}
abline(h = 200, v = c(17.5, 21.3, 32.9))
legend(10, 300, c("mova", "spline", "savgol"), pch = 20, lwd = 2,
col = c(1:3))

par(mfrow = c(1,1))

# Third example
method <- c("lowess","mova","savgol","smooth","spline", "supsmu")

plot(NA, NA, xlim = c(0,100), ylim = c(-50,1100), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "capillary convective PCR")
for (i in 1:length(method)) {
  for (j in c(1,3,5,7)) {
      tmp <- data.frame(na.omit(capillaryPCR[, j]),
na.omit(capillaryPCR[, j+1]))
      tmp.sm <- smoother(tmp[, 1], tmp[, 2], method = list(method[i]))
      lines(data.frame(tmp[, 1], tmp.sm), type = "l", pch = 20, cex = 0.5,
    col = i)
  }
}
legend(0,1000, method, pch = 20, lwd = 2, col = 1:length(method))


par(fig = c(0.5,1,0.25,0.8), new = TRUE)
plot(NA, NA, xlim = c(10,40), ylim = c(50,300), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "")
for (i in 1:length(method)) {
  for (j in c(1,3,5,7)) {
      tmp <- data.frame(na.omit(capillaryPCR[, j]),
na.omit(capillaryPCR[, j+1]))
      tmp.sm <- smoother(tmp[, 1], tmp[, 2], method = list(method[i]))
      lines(data.frame(tmp[, 1], tmp.sm), type = "l", pch = 20, cex = 0.5,
    col = i)
  }
}
legend(0,1000, method, pch = 20, lwd = 2, col = 1:length(method))

# Fourth example
# Comparison of Lowess, Moving average and splines to smooth amplification
# curve data of
# a capillary convective PCR.

plot(NA, NA, xlim = c(10,40), ylim = c(50, 300), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR - Moving average")
movaww <- seq(1,17,4)
for (i in 1:length(movaww)) {
  for (j in c(1,3,5,7)) {
    tmp <- data.frame(na.omit(capillaryPCR[, j]),
      na.omit(capillaryPCR[, j+1]))
    tmp.out <- smoother(tmp[, 1], tmp[, 2],
```

```
                method = list(mova = list(movaww = movaww[i])))

    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20,
  cex = 0.5, col = i)
  }
}
text(10,300, "A)", cex = 3)
legend(25,200, paste("movaww : ", movaww), pch = 20, lwd = 2,
col = 1:length(movaww))

plot(NA, NA, xlim = c(10,40), ylim = c(50, 300), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR - Cubic Spline")
df.fact <- seq(0.5,0.9,0.1)
for (i in 1:length(df.fact)) {
  for (j in c(1,3,5,7)) {
    tmp <- data.frame(na.omit(capillaryPCR[, j]),
      na.omit(capillaryPCR[, j+1]))
    tmp.out <- smoother(tmp[, 1], tmp[, 2], method = list(smooth =
    list(df.fact = df.fact[i])))

    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20,
  cex = 0.5, col = i)
  }
}
text(10,300, "B)", cex = 3)
legend(30,200, paste("df.fact : ", df.fact), pch = 20, lwd = 2,
col = 1:length(df.fact))


plot(NA, NA, xlim = c(10,40), ylim = c(50, 300), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR - Lowess")
f <- seq(0.01,0.2,0.04)
for (i in 1:length(f)) {
  for (j in c(1,3,5,7)) {
    tmp <- data.frame(na.omit(capillaryPCR[, j]),
      na.omit(capillaryPCR[, j+1]))
    tmp.out <- smoother(tmp[, 1], tmp[, 2], method = list(lowess = list(f = f[i])))

    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20,
  cex = 0.5, col = i)
  }
}
text(10,300, "C)", cex = 3)
legend(30,200,  paste("f : ", f), pch = 20, lwd = 2, col = 1:length(f))

plot(NA, NA, xlim = c(10,40), ylim = c(50, 300), xlab = "Time [min]",
     ylab = "Voltage [micro V]",
     main = "ccPCR - Friedman's ''super smoother''")
span <- seq(0.01,0.05,0.01)
for (i in 1:length(span)) {
  for (j in c(1,3,5,7)) {
    tmp <- data.frame(na.omit(capillaryPCR[, j]),
      na.omit(capillaryPCR[, j+1]))
```

```
    tmp.out <- smoother(tmp[, 1], tmp[, 2],
                method = list(supsmu = list(span = span[i])))

    lines(data.frame(tmp[, 1], tmp.out), type = "l", pch = 20,
  cex = 0.5, col = i)
  }
}
text(10,300, "D)", cex = 3)
legend(25,200,  paste("span : ", f), pch = 20, lwd = 2, col = 1:length(span))

par(mfrow = c(1,1), cex = 1)
```

CD74 *Quantitive PCR with a hydrolysis probe and DNA binding dye*

Description

Quantitive PCR (qPCR) with a hydrolysis probe and DNA binding dye (EvaGreen) (Mao et al. 2007) was performed in the VideoScan heating cooling unit. The cycle-dependent increase of the fluorescence was quantified at three different temperatures in order to estimate temperature dependent effects.

Usage

```
data(CD74)
```

Format

A data frame with 60 observations on the following 19 variables. refMFI (referenced Mean Fluorescence Intensity), fluorescence. Dilution A, 1x; Dilution B, 1:10; Dilution B, 1:100.

`Cycle` PCR cycles

`EG.30.A` refMFI at 30 deg Celsius, EvaGreen, Dilution A

`EG.59.5.A` refMFI at 59.5 deg Celsius, EvaGreen, Dilution A

`EG.68.5.A` refMFI at 68.5 deg Celsius, EvaGreen, Dilution A

`HP.30.A` refMFI at 30 deg Celsius, hydrolysis probe, Dilution A

`HP.59.5.A` refMFI at 59.5 deg Celsius, hydrolysis probe, Dilution A

`HP.68.5.A` refMFI at 68.5 deg Celsius, hydrolysis probe, Dilution A

`EG.30.B` refMFI at 30 deg Celsius, EvaGreen, Dilution B

`EG.59.5.B` refMFI at 59.5 deg Celsius, EvaGreen, Dilution B

`EG.68.5.B` refMFI at 68.5 deg Celsius, EvaGreen, Dilution B

`HP.30.B` refMFI at 30 deg Celsius, hydrolysis probe, Dilution B

`HP.59.5.B` refMFI at 59.5 deg Celsius, hydrolysis probe, Dilution B

`HP.68.5.B` refMFI at 68.5 deg Celsius, hydrolysis probe, Dilution B

`EG.30.C` refMFI at 30 deg Celsius, EvaGreen, Dilution C

`EG.59.5.C` refMFI at 59.5 deg Celsius, EvaGreen, Dilution C

`EG.68.5.C` refMFI at 68.5 deg Celsius, EvaGreen, Dilution C

`HP.30.C` refMFI at 30 deg Celsius, hydrolysis probe, Dilution C

`HP.59.5.C` refMFI at 59.5 deg Celsius, hydrolysis probe, Dilution C

`HP.68.5.C` refMFI at 68.5 deg Celsius, hydrolysis probe, Dilution C

Details

The aim was to amplify MLC-2v in the VideoScan platform while the intercalating dye EvaGreen and a hydrolysis probe for MLC-2v were used simultaneously. The primer sequences for MLC-2v were taken from Roediger et al. (2013). The amplification was detected in solution of the 1 HCU (see Roediger et al. 2013 for details). A 20 micro L PCR reaction was composed of 500 nM primer (forward and reverse), 1x Maxima Probe qPCR Master Mix (Fermentas), 1 micro L template (MLC-2v amplification product in different dilutions), 50 nM hydorlysis probe probe for MLC-2v, 0.5x EvaGreen and A. bidest. During the amplification, fluorescence was measured at 3 different temperatures, at 59.5 degree Celsius the annealing temperature, at 68.5 degree Celsius the elongation temperature and at 30 degree Celsius. The FAM channel was used to monitor EvaGreen and the Cy5 channel to monitor the MLC-2v specific hydrolysis probe.

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Mao, F., Leung, W.-Y., Xin, X., 2007. Characterization of EvaGreen and the implication of its physicochemical properties for qPCR applications. *BMC Biotechnol*. 7, 76.

Examples

```
# First example
# Comparison of smoothers and filter on amplification curves
# Amplification curves were measured at three temperature (30,
# 59.5, 68.5 degree Celsius) using the VideoScan HCU (see
# Roediger et al. 2013 for details). MLC-2v was amplified.
# The change of fluorescence was monitored by the intercalating
# dye EvaGreen and hydrolsis probes.

data(CD74)
par(mfrow = c(1,2))
plot(NA, NA, xlim = c(1,30), ylim = c(0,2), xlab = "Cycle",
     ylab = "MFI", main = "VideoScan HCU\nRaw Data")
lim <- 1:30
```

```
for (j in c(2:4)) {
  for (i in seq(j,19,6)) {
    lines(CD74[lim, 1], CD74[lim, i], col = 1)
  }
}
for (j in c(5:7)) {
  for (i in seq(j,19,6)) {
    lines(CD74[lim, 1], CD74[lim, i], col = 2)
  }
}

plot(NA, NA, xlim = c(1,30), ylim = c(0,1.8), xlab = "Cycle",
     ylab = "MFI", main = "VideoScan HCU\nSmoothed Data")
lim <- 1:30
for (j in c(2:4)) {
  for (i in seq(j,19,6)) {
    lines(CD74[lim, 1], smoother(CD74[lim, 1], CD74[lim, i], trans = TRUE),
  col = 1)
  }
}
for (j in c(5:7)) {
  for (i in seq(j,19,6)) {
    lines(CD74[lim, 1], smoother(CD74[lim, 1], CD74[lim, i], trans = TRUE),
  col = 2)
  }
}
par(mfrow = c(1,1))
```

CD75 *Helicase Dependent Amplification in the VideoScan HCU*

Description

Helicase Dependent Amplification in the VideoScan HCU of HPRT1 (Homo sapiens hypoxanthine phosphoribosyltransferase 1)

Usage

```
data(CD75)
```

Format

A data frame with 93 observations on the following 6 variables. The data frame contains three replicates of a HDA for HPRT1.

`CD75.t1` Elapsed time during HDA in seconds.

`CD75.F1` Time-dependent fluorescence during HDA.

`CD75.t2` Elapsed time during HDA in seconds.a numeric vector

`CD75.F2` Time-dependent fluorescence during HDA

`CD75.t3` Elapsed time during HDA in seconds.

`CD75.F3` Time-dependent fluorescence during HDA.

Details

To perform an isothermal amplification in VideoScan, standard conditions for the IsoAmp(R) III Universal tHDA Kit (Biohelix) were used. The reaction was composed of 12.5 micro L buffer A containing 1.25 micro L 10x reaction buffer, 150 nM primer (forward and reverse), 0.75 micro L template (synthetic) and A. bidest which was covered with 50 micro L mineral oil. The primer sequences for HPRT1 were taken from Roediger et al. (2013). Preincubation: This mixture was incubated for 2 min at 95 degree. Celsius and immediately placed on ice. 12.5 micro L of reaction buffer B which was composed of 1.25 micro L 10x buffer, 40 mM NaCl, 5 mM MgSO4, 1.75 micro L dNTPs, 0.2 x Evagreen, 1 micro L Enzyme mix and A. bidest. The fluorescence measurement in VideoScan HCU started directly after adding buffer B at 55, 60 or 65 degree Celsius and revealed optimal conditions for the amplification when using 60 or 65 degree Celsius. Temperature profile (after Preincubation): - 60 seconds at 65 degree Celsius - 11 seconds at 55 degree Celsius && Measurement

Source

Claudia Deutschmann & Stefan Roediger, BTU Cottbus - Senftenberg, Senftenberg, Germany

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(CD75)
## maybe str(CD75) ; plot(CD75) ...
```

`chipPCR.datasets` *Overview for data sets of the* `chipPCR` *package*

Description

The `chipPCR` package contains numerous data sets from commercial and experimental technologies for the amplifiaction of nucleic acids. The data sets include results from qPCR experiments with the VideoScan HCU (Roediger et al. 2013), Bio-Rad (iQ5, CFX96), capilary convectiive PCR (ccPCR) and Roche Light Cycler 1.5. Real-time monitored amplification reactions were performed using standard amplifiaction methods (qPCR, based on Taq polymerase) and quantitive isothermal amplifiaction (qIA). In selected data sets are melting curves and dilution series available. Most of the data sets have equidistant measure points. However, some datasets have none homogenous measure points as indicated below.

capillary convective PCR (ccPCR)

capillaryPCR: The capillary convective PCR (ccPCR) is a modified device of the ccPCR system proposed by Chou et al. 2013.

standard qPCR - commercial thermo cyclers

C60.amp: qPCR Experiment for the Amplification of MLC-2v and Vimentin (as decadic dilutions) Using the Roche Light Cycler 1.5.

C60.melt: Melt Curves MLC-2v and Vimentin for the qPCR experiment C60.amp using the Roche Light Cycler 1.5

C126EG595: A Quantitive PCR (qPCR) with the DNA binding dye (EvaGreen) (Mao et al. 2007) was performed in the Bio-Rad iQ5 thermo cycler. The cycle-dependent increase of the fluorescence was quantified at the elongation step (59.5 deg Celsius).

C126EG685: A Quantitive PCR (qPCR) with the DNA binding dye (EvaGreen) (Mao et al. 2007) was performed in the Bio-Rad iQ5 thermo cycler. The cycle-dependent increase of the fluorescence was quantified at the elongation step (68.5 deg Celsius).

C127EGHP: Quantitive PCR (qPCR) with a hydrolysis probe (Cy5/BHQ2) and DNA binding dye (EvaGreen) (Mao et al. 2007) performed in the Roche Light Cycler 1.5 thermo cycler.

VIMCFX96_60: Human vimentin amplification curve data (measured during annealing phase at 60 deg Celsius) for 96 replicate samples in a Bio-Rad CFX96 thermo cycler.

VIMCFX96_69: Human vimentin amplification curve data (measured during elongatio phase at 69 deg Celsius) for 96 replicate samples in a Bio-Rad CFX96 thermo cycler.

VIMCFX96_meltcurve: Human vimentin melting curve data for 96 replicate samples in a Bio-Rad CFX96 thermo cycler.

VIMiQ5_595: Human vimentin amplification curve data (measured during annealing phase at 59.5 deg Celsius) for 96 replicate samples in a Bio-Rad iQ5 thermo cycler.

VIMiQ5_685: Human vimentin amplification curve data (measured during elongatio phase at 68.5 deg Celsius) for 96 replicate samples in a Bio-Rad iQ5 thermo cycler.

VIMiQ5_melt: Human vimentin melting curve data for 96 replicate samples in a Bio-Rad iQ5 thermo cycler.

standard qPCR - experimental thermo cyclers

C54: qPCR Experiment in the VideoScan heating/cooling-unit for the amplification using different concentrations of MLC-2v input cDNA quantities.

CD74: Quantitive PCR with a hydrolysis probe and DNA binding dye (EvaGreen) for MLC-2v measured at 59.5 degree Celsius (annealing temperature), 68.5 degree Celsius (elongation temperature) and at 30 degree Celsius.

Simulations

Eff625: Highly replicate number amplification curves with an approximate amplification efficiency of 62.5 percent at cycle number 18. The data were derived from a simulation such as the AmpSim function.

Eff750: Highly replicate number amplification curves with an approximate amplification efficiency of 75 percent at cycle number 18. The data were derived from a simulation such as the AmpSim function.

Eff875: Highly replicate number amplification curves with an approximate amplification efficiency of 87.5 percent at cycle number 18. The data were derived from a simulation such as the AmpSim function.

Eff1000: Highly replicate number amplification curves with an approximate amplification efficiency of 100 percent at cycle number 18. The data were derived from a simulation such as the AmpSim function.

Isothermal Amplifcation - Helicase Dependent Amplification

C67: A Helicase Dependent Amplification (HDA) of HPRT1 (Homo sapiens hypoxanthine phosphoribosyltransferase 1), performed at different input DNA quantities using the Bio-Rad iQ5 thermo cycler.

CD75: Helicase Dependent Amplification in the VideoScan HCU of HPRT1 (Homo sapiens hypoxanthine phosphoribosyltransferase 1) measured at at 55, 60 or 65 degree Celsius.

C81: Helicase Dependent Amplification (HDA) of pCNG1 using the VideoScan Platform (Roediger et al. (2013)). The HDA was performed at 65 degree Celsius. Two concentrations of input DNA were used.

C85: Helicase Dependent Amplification (HDA) of Vimentin (Vim) in the VideoScan Platform (Roediger et al. (2013)). The HDA was performed at 65 degree Celsius with three dilutions of input DNA.

Source

Stefan Roediger

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(VIMiQ5_melt)

tmp <- VIMiQ5_melt

plot(NA, NA, xlim = c(55,95), ylim = c(0, 40000),
    xlab = "Temperature (deg Celsius)",ylab = "RFU",
    main = "Melting curve in iQ5 (Bio-Rad)")
apply(tmp[, 2:ncol(tmp)], 2,
      function(x) lines(tmp[1:nrow(tmp),1],x))

Fmean <- rowMeans(tmp[, 2:ncol(tmp)])
lines(tmp[1:nrow(tmp),1], Fmean, col = "red", lwd = 3)

legend(55, 4000, c("Raw", "Mean"), pch = c(19,19), col = c(1,2))
```

chipPCR.sp *Scientific plots of the* chipPCR *package*

Description

This Rd-file contains sophisticated plots and example calculations for the chipPCR package. Selected plots and calculations require additional packages and functions (e.g., drm).

Source

Stefan Roediger, Michal Burdukiewicz

Examples

```
## Not run:
##############################################################################
# Figure 1
#################
pdf(file = "problems.pdf", width = 12, height = 8)
# Use AmpSim to generate an amplification curve with 40 cycles
# and a different Cq
res.pos <- AmpSim(cyc = 1:40, noise = TRUE, nnl = 0.05)
res.pos[5, 2] <- res.pos[5, 2] * 6

res.low <- AmpSim(cyc = 1:40, noise = TRUE, bl = 0.5, ampl = 0.58, Cq = 33)
res.low[c(31), 2] <- NA

res.neg <- AmpSim(cyc = 1:40, b.eff = -0.1, bl = 0.05, ampl = 0.4, Cq = 1,
    noise = FALSE, nnl = 0.5)

res.pos.CPP <- cbind(1:40, CPP(res.pos[, 1], res.pos[, 2],
      bg.outliers = TRUE, smoother = TRUE, method =
"smooth",
      norm = "luqn", rob.reg = "lmrob")$y)

res.low.NA <- cbind(1:40, CPP(res.low[, 1], res.low[, 2], smoother = TRUE,
    method = "smooth", bg.outliers = TRUE, norm = "luqn",
    rob.reg = "lmrob")$y)

res.neg.exc <- cbind(1:40, amptester(res.neg[, 2]))

par(mfrow = c(1,2), las = 0, bty = "n", cex.axis = 1.5, cex.lab = 1.5,
    font = 2, cex.main = 1.8, oma = c(1,1,1,1))
plot(NA, NA, xlim = c(1,40), ylim = c(0, 1.5), xlab = "Cycle",
     ylab = "Raw fluorescence")
mtext("A", cex = 2, side = 3, adj = 0, font = 2)
# abline(lm(res.pos[1:10, 2] ~  res.pos[1:10, 1]))
lines(res.pos)
lines(res.low, col = 2)
arrows(38, min(res.low[, 2], na.rm = TRUE), 38, max(res.low[, 2], na.rm =
```

```
TRUE),
code=3, lwd=3, angle=90, col="grey")
text(38, max(res.low[, 2], na.rm = TRUE) * 0.7,"SNR", cex=1.2)

arrows(29,0.42,31,0.51, lwd=2)
text(29,0.38, "NA", cex=1.2)

points(res.pos[5, 1], res.pos[5, 2], pch=21, cex=4, lwd=5, col="orange")
text(res.pos[5, 1], res.pos[5, 2] * 1.2, "Outlier", cex=1.2)

lines(res.neg, col = 4)
text(20, mean(res.neg[, 2]) *0.9, "No amplification", cex=1.2, col =
"blue")


plot(NA, NA, xlim = c(1,40), ylim = c(0, 1.5), xlab = "Cycle",
     ylab = "Pre-processed fluorescence")
mtext("B", cex = 2, side = 3, adj = 0, font = 2)

lines(res.pos.CPP)
lines(res.low.NA, col = 2)
lines(res.neg.exc, col = 4)

dev.off()

################################################################################
# Figure 2
#################
# Load the shiny package (chipPCR should already be loaded).
# Run from a R console following commands.
require(shiny)

# Invoke the shiny AmpSim app in the default browser.
runApp(paste0(find.package("chipPCR")[1],"/AmpSim.gui"))

################################################################################
# Figure 3
#################
pdf(file = "AmpSim_random.pdf", width = 12, height = 6)
# Draw an empty plot for 50 cycles
par(las = 0, bty = "n", cex.axis = 1.2, cex.lab = 1.2,
    font = 2, cex.main = 1.2, oma = c(1,1,1,1))
plot(NA, NA, xlim = c(1,50), ylim = c(0,1.1), xlab = "Cycle",
     ylab = "RFU")
colors <- rainbow(8)
# Create a loop for 8 amplification curves
# The approximate Cqs are synthesize the temporary Cqs
# by adding a random value to a starting Cq of 25
# Note: ``noise'' is set TRUE with a level of 0.03. This
# adds some noise to the plot.
for (i in 1:8) {
  Cq.tmp <- 25 + rnorm(1) * 5
```

```
  tmp <- AmpSim(1:50, Cq = Cq.tmp, noise = TRUE, nnl = 0.03)
  lines(tmp, col = colors[i])

  # Add the approximate Cq values to the plot
  text(1, 1 - i / 10, paste("Cq ", round(Cq.tmp, 2)), col = colors[i])
}
dev.off()


################################################################################
# Figure 4
#################
pdf(file = "SDM.pdf", width = 12, height = 8)
# Use AmpSim to generate an amplification curve with 40 cycles
# and an approximate Cq of 20 and assign it to the object isPCR.
# isPCR will an object of the class "data.frame".
isPCR <- AmpSim(cyc = 1:40, Cq = 20)

# Invoke the inder function for the object isPCR to interpolate
# the derivatives of the simulated data as object res. The Nip
# parameter was set to 5. This leads to smoother curves. res is
# an object of the class "der".
res <- inder(isPCR, smooth.method = "spline", Nip = 5)

# Plot the the object res and add description to the elements.
par(las = 0, bty = "n", cex.axis = 1.5, cex.lab = 1.5,
    font = 2, cex.main = 1.8, oma = c(1,1,1,1))

plot(isPCR, xlab = "Cycle", ylab = "RFU", ylim = c(-0.15,1),
     main = "", type = "b", pch = 20)

# Add graphical elements for the dervatives and the calculated
# Cq values FDM, SDM, SDm and SDC
  lines(res[, "x"], res[, "d1y"], col = "blue")
  lines(res[, "x"], res[, "d2y"], col = "red")
  summ <- summary(res, print = FALSE)
  abline(v = c(summ["SDM"], summ["SDm"], summ["SDC"]), col = c(3,4,5))
  text(summ["SDM"], 0.5, paste("SDM ~ ", round(summ["SDM"], 2)),
       cex = 1.5, col = 3)
  text(summ["SDC"], 0.7, paste("SDC ~ ", round(summ["SDC"], 2)),
       cex = 1.5, col = 5)
  text(summ["SDm"], 0.9, paste("SDm ~ ", round(summ["SDm"], 2)),
       cex = 1.5, col = 4)
  text(summ["FDM"] + 10, 0.65, paste("FDM ~ ", round(summ["FDM"], 2)),
       cex = 1.5, col = 1)

  legend(1.1, 0.9, c("raw", "first derivative", "second derivative"),
         col = c(1,4,2), lty = c(2,1,1), cex = 1.2, bty = "n")

# Summary of the object res.
summ
#     FDM      SDM      SDm      SDC
#19.81407 19.03015 20.98995 19.98604
```

```
dev.off()
##############################################################################


## End(Not run)
```

CPP *Curve Pre-processor*

Description

The function CPP encompasses a set of functions to pre-process an amplification curve. The pre-processing includes options to normalize curve data, to remove background, to remove outliers in the background range and to test if an amplification is significant.

Usage

```
## S4 method for signature 'numeric,numeric'
CPP(x, y, smoother = TRUE, method = "savgol",
trans = FALSE, method.reg = "lmrob",
bg.outliers = FALSE, median = FALSE,
method.norm = "none", qnL = 0.03, amptest = FALSE,
manual = FALSE, nl = 0.08, bg.range = NULL, ...)

## S4 method for signature 'matrix,missing'
CPP(x, y, smoother = TRUE, method = "savgol",
       trans = FALSE, method.reg = "lmrob",
       bg.outliers = FALSE, median = FALSE,
       method.norm = "none", qnL = 0.03, amptest = FALSE,
       manual = FALSE, nl = 0.08, bg.range = NULL, ...)

## S4 method for signature 'data.frame,missing'
CPP(x, y, smoother = TRUE,
   method = "savgol", trans = FALSE,
   method.reg = "lmrob", bg.outliers = FALSE,
   median = FALSE, method.norm = "none",
   qnL = 0.03, amptest = FALSE,
   manual = FALSE, nl = 0.08, bg.range = NULL, ...)
```

Arguments

x	is a vector containing the time or cycle values or a matrix or data frame containing both time or cycle values and fluorescence.
y	is a vector containing the fluorescence values. Omitted if x is a data frame or matrix.
smoother	logical parameter which indicates if smoother should be used.

method	a vector of names defining which smoothing method should be used. The Savitzky-Golay smoothing filter is the default smoother. Use "lowess" for LOWESS smoother (locally-weighted polynomial regression), "mova" for moving average, "savgol" for Savitzky-Golay smoothing filter, "smooth" for cubic spline smooth, "spline" for standard cubic spline smooth, "supsmu" for Friedman's SuperSmoother, "whit1" for weighted Whittaker smoothing with a first order finite difference penalty, "whit2" for weighted Whittaker smoothing with a second order finite difference penalty or "all" for all implemented smoothing algorithms.
trans	defines if the slope of the background range in a curve should be corrected by a linear regression.
method.reg	defines the method ("rfit", "lmrob", "rq") for the robust linear regression. If equal to "least", CPP uses linear regression.
bg.outliers	is a logical argument which to remove outliers in the background range.
median	If set to TRUE, median is used instead of mean in outlier replacement. The mean is used by default.
method.norm	is a argument to use a "none", "minm", "max", "lugn", or "zscore" normalization.
qnL	is the quantile to be used for the quantile normalization.
amptest	is a logical operator which is used to set a test for a positive amplification.
manual	is used to test for a fixed threshold value of the background.
nl	is a value used as fixed threshold value for the background.
bg.range	is a numeric vector of length 2 pointing the background range. If NULL, the background range is calculated by bg.max function.
...	dot operator for diverse arguments of smoother for details).

Details

The function CPP uses the function bg.max to estimate automatically the start of the amplification process. In the background range there is often noise which makes it harder to determine a meaningful background value. Therefore CPP can optionally remove outliers by finding the value with largest difference from the mean as provided by the rm.outlier function. The functions also tries to prevent calculations of non amplified signals.

The slope of the background range is often unequal to zero. By setting the parameter trans it is possible to apply a simple correction of the slope. Thereby either a robust linear regression by computing MM-type regression estimators, a nonparametric rank-based estimator or a standard linear regression model. Care is needed when using trans with time series (see lm for details).

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

Normalization: normalizer

Smoothing: smoother

Robust linear regression: lm.coefs

Examples

```
# Function to pre-process an amplification curve.
# Take a subset of the C17 data frame.
data(C17)
par(mfrow = c(2,1))
plot(NA, NA, xlab = "Time [sec]", ylab = "refMFI",
     main = "HDA Raw Data",
     xlim = c(0, 2500), ylim = c(0,1.1), pch = 20)
for (i in 3:5) {
  lines(C17[1:50, 1], C17[1:50, i], col = i - 2,
type = "b", pch = 20)
}

legend(50, 0.5, c("55 deg Celsius", "60 deg Celsius", "65 deg Celsius"),
col = c(1,2,3), pch = rep(20,3))

# Use CPP to pre-process the data by removing the missing value and
# normalization of the data
plot(NA, NA, xlab = "Time [sec]", ylab = "refMFI",
     main = "Curve Pre-processor Applied to HDA Data",
     xlim = c(0, 2500), ylim = c(0,1.1), pch = 20)
for (i in 3:5) {
  y.cpp <- CPP(C17[2:50, 1], C17[2:50, i], method.norm = "minm",
      bg.outliers = TRUE)$y.norm
  lines(C17[2:50, 1], y.cpp, col = i - 2,
type = "b", pch = 20)
}
legend(50, 1, c("55 deg Celsius", "60 deg Celsius", "65 deg Celsius"),
col = c(1,2,3), pch = rep(20,3))
par(mfrow = c(1,1))

# Example for dB analysis
# qPCR analysis using the VIMCFX96_60 and the VIMCFX96_69 data sets from a
# 96-well plate cycler (Bio-Rad CFX96, EvaGreen detection) experiment.
# The dB values during the annealing (60 Celsius) and elongation (69 Celsius)
# phase were determined by calling dB from CPP. The annealing phase values
# were plotted against the elongation phase values and analyzed by
# linear regression (lm, stats) and correlation test (cor.test, stats). Next, we
# computed the interquartile range using IQR (stats) with the default settings.
# The IQR at 60 Celsius was ~ 0.28 dB and ~ 0.21 dB at 69 Celsius.

par(mfrow = c(1,2), las = 0, bty = "n", cex.axis = 1.5, cex.lab = 1.5,
     font = 2, cex.main = 1.1, oma = c(1,1,1,1))

pointer <- function (x, pos = 1, w = 5){
    xx <- pos + rep(seq(-0.1, 0.1, length.out = w),
  ceiling(length(x)/w))
    yy <- sort(x)
    points(xx[1:length(yy)], yy, pch = 19)
    points(pos, mean(x, na.rm = TRUE), col = 2, pch = 15, cex = 1.5)
  }
```

```
T60 <- cbind(apply(VIMCFX96_60[, c(2L:97)], 2,
    function(x) CPP(VIMCFX96_60[, 1], x, qnL = 0.03)$dB))
T69 <- cbind(apply(VIMCFX96_69[, c(2L:97)], 2,
    function(x) CPP(VIMCFX96_69[, 1], x, qnL = 0.03)$dB))

res <- list(T60 = T60, T69 = T69)

plot(T60, T69, xlab = "dB @ 60 deg. Celsius",
      ylab = "dB @ 69 deg. Celsius")
mtext("A", cex = 2, side = 3, adj = 0, font = 2)
res.lm <- lm(T69 ~ T60)
abline(res.lm)

summary(res.lm)
cor.test(T60, T69)

boxplot(res, ylab = "dB", ylim = c(1.45, 3.5))
  pointer(T60, 1, 6)
  pointer(T69, 2, 6)

mtext("B", cex = 2, side = 3, adj = 0, font = 2)

wcx <- wilcox.test(T60, T69)
lines(c(1,2), c(3.2,3.2))
text(1.5, 3.3, paste("p :", wcx$p.value))

IQR(T60)
IQR(T69)
par(mfrow = c(1,1))
```

`der` *Class* `"der"`

Description

An S4 class containing the output `inder` function.

Slots

`.Data`**:** `"matrix"` is a matrix containing smoothed data as well as the first and second derivative.

`method`**:** `"character"` used method of smoothing.

Methods

summary `signature(object = "der")`: calculates and prints approximate first derivative maximum, second derivative maximum, second derivative minimum and second derivative center. See `summary.der`.

show `signature(object = "der")`: prints only `.Data` slot of the object.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

inder

Examples

```
pcr <- AmpSim(cyc = 1:40)
res <- inder(pcr[, 1], pcr[, 2])
sums <- summary(res)
print(sums)
```

---

eff *Class* "eff"

Description

An S4 class containing the output effcalc function.

Slots

.Data: "matrix" containing the "Concentration", "Location" (mean, median), "Deviation" (standard deviation, median absolute deviation), "Coefficient of Variance" (CV, RSD) sequential in the columns.

amplification.efficiency: "numeric" value representing amplification efficiency.

regression: "lm" the results of the linear regression and .

correlation.test: "htest". the correlation test (Pearson) results.

Methods

plot signature(x = "eff"): plots calculated efficiency. See plot.eff

show signature(object = "eff"): prints only .Data slot of the object.

summary signature(object = "eff"): prints information about object prettier than show.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

effcalc, plot.eff,

Eff1000 *Highly Replicate Number Amplification Curves*

Description

Highly replicate number amplification curves with an approximate amplification efficiency of 100 percent at cycle number 18. The data were derivd from a simulation such as the AmpSim function.

Usage

```
data(Eff1000)
```

Format

A data frame with 40 (Cycles) observations on the following 1000 (amplification curves) variables. The columns are all replicates.

Examples

```
data(Eff1000)

plot(NA, NA, xlim = c(1,40), ylim = c(0,max(Eff1000)), xlab = "Cycle",
    ylab = "RFU",
    main = "Amplification Curves with 100 Percent Efficiency")
apply(Eff1000[, 1:ncol(Eff1000)], 2, function(x) lines(1:40,x))

Fmean <- rowMeans(Eff1000[, 1:ncol(Eff1000)])
lines(1:40, Fmean, col = "red", lwd = 3)

legend(1, quantile(unlist(Eff1000), 0.9), c("Raw", "Mean"),
pch = c(19,19), col = c(1,2))
```

Eff625 *Highly Replicate Number Amplification Curves*

Description

Highly replicate number amplification curves with an approximate amplification efficiency of 62.5 percent at cycle number 18. The data were derivd from a simulation such as the AmpSim function.

Usage

```
data(Eff625)
```

Format

A data frame with 40 (Cycles) observations on the following 1000 (amplification curves) variables. The columns are all replicates.

Examples

```
data(Eff625)

plot(NA, NA, xlim = c(1,40), ylim = c(0,max(Eff625)), xlab = "Cycle",
    ylab = "RFU",
    main = "Amplification Curves with 62.5 Percent Efficiency")
apply(Eff625[, 1:ncol(Eff625)], 2, function(x) lines(1:40,x))

Fmean <- rowMeans(Eff625[, 1:ncol(Eff625)])
lines(1:40, Fmean, col = "red", lwd = 3)

legend(1, quantile(unlist(Eff625), 0.9), c("Raw", "Mean"),
      pch = c(19,19), col = c(1,2))
```

Eff750 *Highly Replicate Number Amplification Curves*

Description

Highly replicate number amplification curves with an approximate amplification efficiency of 75 percent at cycle number 18. The data were derivd from a simulation such as the AmpSim function.

Usage

```
data(Eff750)
```

Format

A data frame with 40 (Cycles) observations on the following 1000 (amplification curves) variables. The columns are all replicates.

Examples

```
data(Eff750)

plot(NA, NA, xlim = c(1,40), ylim = c(0,max(Eff750)), xlab = "Cycle",
    ylab = "RFU",
    main = "Amplification Curves with 75 Percent Efficiency")
apply(Eff750[, 1:ncol(Eff750)], 2, function(x) lines(1:40,x))

Fmean <- rowMeans(Eff750[, 1:ncol(Eff750)])
lines(1:40, Fmean, col = "red", lwd = 3)

legend(1, quantile(unlist(Eff750), 0.9), c("Raw", "Mean"),
pch = c(19,19), col = c(1,2))
```

Eff875 *Highly Replicate Number Amplification Curves*

Description

Highly replicate number amplification curves with an approximate amplification efficiency of 87.5 percent at cycle number 18. The data were derivd from a simulation such as the AmpSim function.

Usage

```
data(Eff875)
```

Format

A data frame with 40 (Cycles) observations on the following 1000 (amplification curves) variables. The columns are all replicates.

Examples

```
data(Eff875)

plot(NA, NA, xlim = c(1,40), ylim = c(0,max(Eff875)), xlab = "Cycle",
    ylab = "RFU",
    main = "Amplification Curves with 87.5 Percent Efficiency")
apply(Eff875[, 1:ncol(Eff875)], 2, function(x) lines(1:40,x))

Fmean <- rowMeans(Eff875[, 1:ncol(Eff875)])
lines(1:40, Fmean, col = "red", lwd = 3)

legend(1, quantile(unlist(Eff875), 0.9), c("Raw", "Mean"),
pch = c(19,19), col = c(1,2))
```

effcalc *Analysis of the amplification efficiency*

Description

effcalc is used for a fast calculation of the amplification efficiency of a dilution. `effcalc` returns an object of the class list.

Usage

```
## S4 method for signature 'numeric,numeric'
effcalc(x, y, logx = TRUE, RSD = FALSE, rob = FALSE,
                                    level = 0.95)
## S4 method for signature 'matrix,missing'
effcalc(x, y, logx = TRUE, RSD = FALSE, rob = FALSE,
                                   level = 0.95)
## S4 method for signature 'matrix,missing'
effcalc(x, y, logx = TRUE, RSD = FALSE, rob = FALSE,
                                   level = 0.95)
```

Arguments

x	is the column of a data frame for the concentration (dilution).
y	are multiple columns of fluorescence values from a data.frame (e.g., [, c(1:n)]).
logx	is a logical parameter used to convert the concentration into a decadic logarithm.
RSD	Setting the option RSD = TRUE shows the relative standard deviation (RSD) in percent.
rob	Using the option rob as TRUE the median and the median absolute deviation (MAD) is plotted instead of the mean and standard deviation.
level	Tolerance/confidence level.

Value

eff object.

Author(s)

Stefan Roediger

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

S. Mehra and W.-S. Hu. A kinetic model of quantitative real-time polymerase chain reaction. *Biotechnology and bioengineering*, 91(7):848–860, Sept. 2005. ISSN 0006-3592. doi: 10.1002/bit.20555. PMID: 15986490.

M. Guescini, D. Sisti, M. B. Rocchi, L. Stocchi, and V. Stocchi. A new real-time PCR method to overcome significant quantitative inaccuracy due to slight amplification inhibition. *BMC Bioinformatics*, 9(1):326, July 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-326. http://www.biomedcentral.com/1471-2105/9/326/abstract. PMID: 18667053.

A. Tichopad, M. Dilger, G. Schwarz, and M. W. Pfaffl. Standardized determination of real-time PCR efficiency from a single reaction set-up. *Nucleic Acids Research*, 31(20):e122, Oct. 2003. ISSN 1362-4962. PMID: 14530455 PMCID: PMC219490.

A. Staalberg, P. Aman, B. Ridell, P. Mostad, and M. Kubista. Quantitative real-time PCR method for detection of b-lymphocyte monoclonality by comparison of kappa and lambda immunoglobulin light chain expression. *Clinical Chemistry*, 49(1):51–59, Jan. 2003. ISSN 0009-9147. PMID: 12507960.

W. Liu and D. A. Saint. A new quantitative method of real time reverse transcription polymerase chain reaction assay based on simulation of polymerase chain reaction kinetics. *Analytical Biochemistry*, 302(1):52–59, Mar. 2002. ISSN 0003-2697. doi: 10.1006/abio.2001.5530. PMID: 11846375.

See Also

eff

plot.eff

Examples

```
## Not run:
# First Example
# Amplification efficiency plot

require(MBmca)
par(mfrow = c(1,2), las = 0, bty = "n", cex.axis = 1.2, cex.lab = 1.2,
    font = 2, cex.main = 1, oma = c(0.1, 0.1, 0.5, 0.1))

# Simulate a qPCR reaction with AmpSim for 45 cycles and some noise.
# Define number of cycles
cyc.no <- 45

# Create an empty plot
plot(NA, NA, xlim = c(1,cyc.no), ylim = c(0.01,1.1), xlab = "Cycles",
     ylab = "Fluorescence", main = "")
mtext("A", cex = 2, side = 3, adj = 0, font = 2)

# Create a sequence of "whised" Cq value for the simulation
cycle <- rep(seq(15, 34, 3.5), 3)

# Define the levels for the decadic dilution with concentrations
# from 100 to 0.001 (six steps).
# The in-slico experiment is designed to have three replicates at
# six dilution steps.
dilution <-rep(c(100, 10, 1, 0.1, 0.01, 0.001), 3)

# Create an empty matrix for the results of the concentration
# dependent Cq values
ma.out <- matrix(data = NA, nrow = cyc.no,
 ncol = length(cycle))

# Use AmpSim to simulate amplification curves at different
# concentrations. The simulation is performed with the addition
# of some noise. This will do a generation of unique amplification
# curves, even under idential paramter settings. Calculate the
```

```
# pseudo Second Derivative Maximum (SDM) (Cq) by using the the
# diffQ2 function from the MBmca package.

Cq.out <- vector()

for (i in 1:18) {
  ma.out[1:cyc.no, i] <- AmpSim(cyc = c(1:cyc.no), b.eff = -50, bl = 0.001,
ampl = 1, Cq = cycle[i], noise = TRUE,
nnl = 0.02)[, 2]
  lines(1:cyc.no, ma.out[, i])
  tmpP <- mcaSmoother(1:cyc.no, ma.out[, i])
  #TURNED OFF INDER - NEW COMPATIBILITY TO MBmca STILL PENDING
  Cq.tmp <- diffQ2(tmpP, inder = FALSE)$xTm1.2.D2[1]
  abline(v = Cq.tmp)
  Cq.out <- c(Cq.out, Cq.tmp)
}

# Assign the calculated Cqs to the corresponding concentrations
tmp <- data.frame(dilution[1:6],
  Cq.out[1:6],
  Cq.out[7:12],
  Cq.out[13:18])

# Determine the amplification efficiency by using the effcalc function
plot(effcalc(tmp[, 1], tmp[, 2:4]), CI = TRUE)
mtext("B", cex = 2, side = 3, adj = 0, font = 2)
par(mfrow = c(1,1))
## End(Not run)
```

---

fixNA *Impute missing values into a column of amplification data*

Description

The function fixNA imputes missing values in a single column of data. The imputation is based on a linear approximation by default. However, the data can also be estimated from an approximation by splines.

Usage

```
## S4 method for signature 'numeric,numeric'
fixNA(x, y, spline = TRUE, verbose = FALSE)
## S4 method for signature 'matrix,missing'
fixNA(x, y, spline = TRUE, verbose = FALSE)
## S4 method for signature 'data.frame,missing'
fixNA(x, y, spline = TRUE, verbose = FALSE)
```

Arguments

x	x is a vector containing the time or cycle values or data frame/matrix where second column contains missing data.
y	y is a vector containing missing values. Used only if x is also a vector.
spline	spline is a logical operator which defines of missing values should be imputed by spline approximation as per spline. The default is linear a approximation as per approx.
verbose	verbose defines if the number of missing values should be reported.

Details

Amplification data of experimental systems may contain missing values (NA). The NAs may be caused by detector problems, acquisition error or other assorted problems. There are different ways to handle missing values. One approach is to ignore NAs which is generally acceptable. However, in case of further calculation it is often necessary to handle cases of missing values in a way that the next calculation steps can be performed. Missing values can be eliminated by a imputation. Imputation encompasses various approaches. This includes to calculate a location parameter (e.g., mean, median) or other significant values (e.g., minimum, maximum, modus) of a data column. However, in non-linear processes such as amplification processes its is better to estimate the missing values from a trend. The function fixNA was empirically tested and relies on a linear trend estimation based on the approx function. This approach is useful but may be problematic on the phases other then background or plateau phases of an amplification reaction. The parameter spline on fixNA enables a trend estimation on splines and may be more appropriate in most scenarios. Other smoothing functions such as the Savitzky-Golay smoothing filter have the intrinsic capability to remove missing values [Savitzky and Golay 1964, Eilers 2003].

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Eilers, P. H. C. Anal. Chem. 2003, 75, 3631–3636. Savitzky, A.; Golay, M. J. E. Anal. Chem. 1964, 36, 1627–1639

See Also

approx, spline.

Examples

```
# Simulate a qPCR reaction with AmpSim with for 40 cycles
res <- AmpSim(cyc = c(1:40))
# Introduce a missing value in the transition between
# the background and the exponential phase
res.NA <-res
res.NA[18, 2] <- NA
par(mfrow = c(2,2))
plot(res, xlab = "Cycles", ylab = "Fluorescence", type = "b", pch = 20,
```

```
     main = "Simulation without missing value")
     abline(v = c(17.5, 18.5), col = "grey")
     abline(h = c(0.09, 0.14), col = "grey")

res.NA.linear <- fixNA(res.NA[, 1], res.NA[, 2], spline = FALSE,
       verbose = FALSE)
plot(res.NA.linear, xlab = "Cycles", ylab = "Fluorescence", type = "b",
     pch = 20, main = "Simulation with linear imputed\n NA
     value during transition")
     abline(v = c(17.5, 18.5), col = "grey")
     abline(h = c(0.09, 0.14), col = "grey")

plot(res.NA, xlab = "Cycles", ylab = "Fluorescence", type = "b", pch = 20,
     main = "Simulation with missing\n value during transition")
     abline(v = c(17.5, 18.5), col = "grey")
     abline(h = c(0.09, 0.14), col = "grey")
res.NA.spline <- fixNA(res.NA[, 1], res.NA[, 2], spline = TRUE,
       verbose = FALSE)
plot(res.NA.spline, xlab = "Cycles", ylab = "Fluorescence", type = "b",
      pch = 20, main = "Simulation with spline imputed\n NA value
      during transition")
     abline(v = c(17.5, 18.5), col = "grey")
     abline(h = c(0.09, 0.14), col = "grey")
par(mfrow = c(1,1))
```

`humanrater` *humanrater, a graphical interface to rate curves*

Description

The function `humanrater` is an interactive function, which can be used to rate a curve for a certain characteristic. `humanrater` draws individual graphs of a curve and prompts an input field for the user. This function can be used to compare the human rating and the rating of a machine.

Usage

```
humanrater(x, cyc = 1, repeats = 1,
           designations = list(y = "yes", a = "ambiguous", n = "not"),
           shuffle = TRUE, ...)
```

Arguments

`x`	is the input data (matrix or data.frame).
`cyc`	is the index of column containing the cycle data.
`repeats`	number of repeats to rate the samples.
`designations`	a named list of length at least 2. See Details.
`shuffle`	logical, if `TRUE` sequence of curves is shuffled for purpose of rating.
`...`	additional arguments to `plot` function.

Details

A user can specify the list of designations characterizing the curve, where the names of elements specify short designations used during rating. Defaults are y for "yes", a for "ambiguous" and n for "no".

It is possible to supply longer or shorter designations lists.

Author(s)

Michal Burdukiewicz, Stefan Roediger

Examples

```
testdata <- data.frame(1:35,
       AmpSim(Cq = 15, noise = TRUE)[, 2],
       AmpSim(Cq = 25, noise = TRUE)[, 2],
       rnorm(35),
       AmpSim(Cq = 35, noise = TRUE)[, 2],
       rnorm(35),
       AmpSim(Cq = 45, noise = TRUE)[, 2])

#check testdata for significance of amplification in two repeats
human.test1 <- humanrater(testdata, repeats = 2)

#check testdata for significance of amplification in one repeat and declare more
#finger friendly (but less obvious) designations
human.test2 <- humanrater(testdata, repeats = 1, list(q = "yes", w = "no"))
```

---

inder *Interpolate derivatives*

Description

A function inder ("in" + "der" = interpolate derivatives)) for interpolating first and second derivatives using the five-point stencil. Therefore this function can be used to estimate the Cq (cycle of quantification) of an amplification curve. First positive derivative also known as First Derivative Maximum (FDM) and the Second Derivative Maximum (SDM) are calculated this way (Ruijter et al. 2013). However, from the mathematical point of view it can also be used to calculate the melting point for melting curve analysis (compare Roediger et al. 2013) provided that the sign of the derivative is changed.

Usage

```
inder(x, y, Nip = 4, logy = FALSE, smooth.method = "spline")
```

Arguments

x	is a numeric vector of independent variable or data frame/matrix containing abscissa and ordinate values.
y	is a vector of dependent variable. Omitted if x is data frame or matrix.
Nip	is a value which defines how often an interpolation takes place at n equally spaced points spanning the interval (default 4). Nip is a such a resolution. A high Nip may improve the precision. Nips less than 2 and higher than 20 are not meaningful for conventional qPCR with up to 50 cycles. However, higher Nips might be appropriate for isothermal amplification reactions.
logy	If logy is TRUE than a semi-decadic log scale graph (corresponds to the linear phase) to illustrate the exponential dynamic of the qPCR amplification is used.
smooth.method	a character vector of length 1 or NULL defining smoothing method used. Currently accepted character values: "spline", "supsmu". If NULL, smoothing is not performed.

Details

The function y = f(x) is numerically derived by five-point stencil. This method do not require any assumptions regarding the function f.

A smoothing procedure greatly enhances calculating derivative calculation. inder uses two smoothing algorithms best suited for this approach. A smoothing can be ommited by setting smooth.method to NULL, which is advisable in case of the data already smoothed.

Value

An object of der class containing smoothed original data.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Ruijter JM, Pfaffl MW, Zhao S, et al. (2013) Evaluation of qPCR curve analysis methods for reliable biomarker discovery: bias, resolution, precision, and implications. *Methods* San Diego Calif 59:32–46.

Surface Melting Curve Analysis with R. S. Roediger, A. Boehm and I. Schimke. *The R Journal*. 5(2):37–52, 2013. http://journal.r-project.org

Examples

```
# First example
# Derive sinus
x <- 1:100/10
y <- sin(x)
ders <- inder(x, y)
plot(x, y, type = "l")
lines(ders[, "x"], ders[, "d1y"], col = "red")
```

```
lines(ders[, "x"], ders[, "d2y"], col = "green")
legend("topright", c("f(x)", "f'(x)", "f''(x)"), lty = 1, col = c("black",
"red", "green"))

# Second example
# Determine the approximate second derivative maximum
# for a qPCR experiment. SDM,
isPCR <- AmpSim(cyc = 1:40)

res <- inder(isPCR)

plot(isPCR, xlab = "Cycle", ylab = "RFU", ylim = c(-0.15,1),
     main = "Approximate Second Derivative Maximum (SDM)",
     type = "b", pch = 20)

  lines(res[, "x"], res[, "d1y"], col = "blue")
  lines(res[, "x"], res[, "d2y"], col = "red")
  summ <- summary(res, print = FALSE)
  abline(v = c(summ["SDM"], summ["SDm"], summ["SDC"]), col = c(3,4,5))
  text(summ["SDM"], 0.5, paste0("SDM ~ ", round(summ["SDM"], 2)),
       cex = 1.5, col = 3)
  text(summ["SDC"], 0.7, paste0("SDC ~ ", round(summ["SDC"], 2)),
       cex = 1.5, col = 5)
  text(summ["SDm"], 0.9, paste0("SDm ~ ", round(summ["SDm"], 2)),
       cex = 1.5, col = 4)
  text(summ["FDM"] + 10, 0.65, paste("FDM ~ ", round(summ["FDM"], 2)),
       cex = 1.5, col = 1)

  legend(1, 1, c("raw", "first derivative", "second derivative"),
 col = c(1,4,2), lty = c(2,1,1), cex = 1.2)
```

`lm.coefs` *Compute linear model coefficients*

Description

Computes linear model using the robust linear regression.

Usage

```
lm.coefs(x, y, method.reg)
```

Arguments

`x`	a vector of ordinate values.
`y`	a vector of abscissa values.
`method.reg`	defines the method ("rfit", "lmrob", "rq", "least") for the linear regression.

Details

lm.coefs is convenient wrapper around few functions performing normal (least squares) and robust linear regression. If the robust linear regression is impossible, lm.coefs will give a warning and perform linear regression using the least squares method. This function can be used to calculate the background of an amplification curve. The coefficients of the analysis can be used for a trend based correction of the entire data set.

Value

Data frame with one column and two rows representing coefficients of the linear model.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

rq, rfit, lm, lmrob

Examples

```
plot(VIMCFX96_69[, 1], VIMCFX96_69[, 2], type = "l", xlab = "Cycle",
     ylab = "Fluorescence")
rect(1,0,10,5000)
method <- c("lmrob", "rq", "least", "rfit")
for (i in 1:4) {
  tmp <- lm.coefs(VIMCFX96_69[1:10, 1], VIMCFX96_69[1:10, 2],
  method.reg = method[i])
  abline(a = tmp[1, 1], b = tmp[2, 1], col = i + 1, lwd = 1.5)
}
legend(2, 3000, c("Data", "lmrob", "rq", "least", "rfit"), lty = 1, col = 1:5,
       cex = 1.5)
```

MFIaggr *Multiple comparison of the cycle dependent variance of the fluorescence*

Description

MFIaggr is used for a fast multiple comparison of the cycle dependent variance of the fluorescence. A similar tool with different scope is interated in the MFIerror function (Roediger et al. 2013).

Usage

```
## S4 method for signature 'numeric,numeric'
MFIaggr(x, y, cyc = 1, fluo = 2:ncol(x),
   RSD = FALSE, rob = FALSE, llul = c(1,10))
## S4 method for signature 'matrix,missing'
MFIaggr(x, y, cyc = 1, fluo = 2:ncol(x),
   RSD = FALSE, rob = FALSE, llul = c(1,10))
## S4 method for signature 'data.frame,missing'
MFIaggr(x, y, cyc = 1, fluo = 2:ncol(x),
   RSD = FALSE, rob = FALSE, llul = c(1,10))
```

Arguments

x	is the column of a data frame for the cycle or data.frame/matrix with whole data.
y	are multiple columns of fluorescence values from a data.frame (e.g., [, c(1:n)]).
cyc	is the index of column containing the cycle data. Used only if y is missing.
fluo	are the columns containing the fluorescence data. Used only if y is missing.
RSD	Setting the option RSD = TRUE shows the relative standard deviation (RSD) in percent.
rob	Using the option rob as TRUE the median and the median absolute deviation (MAD) are calculated instead of the mean and standard deviation.
llul	is a parameter to define the lower and upper data limit (cycle), aka region of interest (ROI) used for the density and quantile plot.

Value

An object of the class refMFI. refMFI means referenced Mean Fluorescence Intensity (Roediger et al. 2013).

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

Examples

```
# First Example
# Cycle dependent variance of the refMFI using standard measures
# (Mean, Standard Deviation (SD)).
# Use Standard Deviation (SD) in the plot

data(VIMCFX96_60)

MFIaggr(VIMCFX96_60[, 1], VIMCFX96_60[, 2:ncol(VIMCFX96_60)])
```

```
#alternative usage
MFIaggr(VIMCFX96_60)

#only second and forth column
plot(MFIaggr(VIMCFX96_60, fluo = c(2, 4)))

# Example
# Use of MFIaggr to test for heteroskedasticity using the Breusch-Pagan
# test. The data were aggregated with the MFIaggr function and assigned to
# the object res. The standard deviation was transformed to the variance.
# The plot shows the cycle dependent variance.
# First cycles 1 to 10 of 96 qPCR replicate amplification curves were
# analyzed. Next the cycles 1 to 40 of the same amplification curve data
# were analyzed. The Breusch-Pagan confirmed the heteroskedasticity in the
# amplification curve data.

par(mfrow = c(1,2), bty = "n")
res <- MFIaggr(VIMCFX96_60[, 1], VIMCFX96_60[, 2:ncol(VIMCFX96_60)],
       llul = c(1,10))
head(res)
plot(res[, 1], res[, 3]^2, xlab = "Cycle", ylab = "Variance of refMFI",
     xlim = c(1,10), main = "ROI from Cycle 1 to 10", pch = 19, type = "b")
abline(v = c(1,10), col = "grey", lty = 2, lwd = 2)
legend("top", paste0("Breusch-Pagan test p-value: \n", format(summary(res)[5],
       digits = 2)), bty = "n")

res <- MFIaggr(VIMCFX96_60[, 1], VIMCFX96_60[, 2:ncol(VIMCFX96_60)],
       llul = c(1,40))
head(res)
plot(res[, 1], res[, 3]^2, xlab = "Cycle", ylab = "Variance of refMFI",
     main = "ROI from Cycle 1 to 40", pch = 19, type = "b")
abline(v = c(1,40), col = "grey", lty = 2, lwd = 2)
legend("top", paste0("Breusch-Pagan test p-value: \n", format(summary(res)[5],
       digits = 2)), bty = "n")
```

`MFIaggr.gui` *Multiple Comparison of the Cycle Dependent Variance - Graphical User Interface*

Description

`MFIaggr.gui` is a Graphical User Interface based on the `shiny` package. The core if this GUI is based on the Comparison of the `MFIaggr` function. The cycle dependent variance can be analyzed intuitively. A report generation is possible from this GUI.

Usage

```
MFIaggr.gui()
```

Warning

Any ad-blocking software may be cause of malfunctions.

Author(s)

Stefan Roediger, Michal Burdukiewicz.

See Also

MFIaggr

Examples

```
# Run from a R console following commands
require(shiny)

# Invoke the shiny AmpSim app in the default browser
runApp(paste0(find.package("chipPCR")[1],"/MFIaggr.gui"))
```

---

normalizer *Normalize data*

Description

`normalizer` is a function to normalize any data set. It is possible to chose from different methods (see Details). The function is useful if the data from an experiment have considerable variation regarding the background and plateau signal.

Usage

```
normalizer(y, method.norm = "none", qnL = 0.03)
```

Arguments

`y`	is a vector containing the fluorescence values.
`method.norm`	is a argument to use a "none", "minm", "max", "lugn", or "zscore" normalization.
`qnL`	is the quantile to be used for the quantile normalization. Ignored if `method.norm` is not euqal to "luqn".

Details

The parameter qnL is a user defined quantile which is used for the quantile normalization. A quantile normalization herein refers to an approach which is less prone to outliers than a normalization based on the minimum and the maximum of an amplification curve. minm does a min-max normalization between 0 and 1 (see Roediger et al. 2013 for explanation). max does a normalization to the maximum value (MFI/max(MFI)). lugn does a quantile normalization based on a symmetric proportion as defined by the qnL parameter (e.g., qnL = 0.03 equals 3 and 97 percent quantiles). zscore performs a z-score normalization with a mean of 0 and a standard deviation of 1.

Value

A vector of normalized fluorescence values.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Surface Melting Curve Analysis with R. S. Roediger, A. Boehm and I. Schimke. *The R Journal*. 5(2):37–52, 2013. http://journal.r-project.org

See Also

CPP

Examples

```
normalizer(C17[2L:50, 1], "minm")
```

---

plot.bg *Plot* bg *objects*

Description

Draw quick diagnostic plots of amplification reaction.

Usage

```
## S4 method for signature 'bg'
plot(x, what = 1:3, add = FALSE, indicators = TRUE,
    legend = TRUE, stan.labs = TRUE,
    plot.colors = c("black", "red", "blue"), ...)
```

Arguments

x	is a bg object.
what	is a vector specifying what should be plotted. 1 means raw data, 2 means first derivative and 3 means second derivative. Any combination of mentioned values is valid, for example: c(1,2), c(1, 2, 3), c(2, 3). See Details and Examples.
add	is a "logical" argument. If TRUE, plot is added to existing plot. Moreover, enforces indicators = FALSE.
indicators	is a "logical" argument. If FALSE, background start, stop and plateau transition indication lines aren't plotted.
legend	is a "logical" argument. If TRUE, legend is added to the plot.
stan.labs	is a "logical" argument. If TRUE, standard axis labels ("Cycle" and "Fluorescence") are added.
plot.colors	is a vectors of colors used in plot. Must have length 3.
...	Arguments to be passed further to the plot function, such as graphical parameters.

Details

plot.bg is simplified, ready-to-use version of plot.der, which still can be used whenever more flexible function is needed.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

bg, plot.der

Examples

```
#step-by-step plotting bg object

res <- AmpSim(cyc = 1:40, Cq = 25)
background <- bg.max(res)

plot(background)

#above is equivalent of below
plot(rounder(inder(res)), xlab = "Cycles",
     ylab = "Fluorescence",
     pch = 20, legend = FALSE)

abline(v = slot(background, "bg.start"))
  text(slot(background, "bg.start"), 0.2, "Background start", pos = 4)
abline(v = slot(background, "bg.stop"), col = "blue")
  text(slot(background, "bg.stop"), 0.25, "Background stop", pos = 4,
```

```
       col = "blue")
abline(v = slot(background, "amp.stop"), col = "green")
  text(slot(background, "amp.stop"), 0.3, "Plateau transition", pos = 4,
       col = "green")
legend(4, 1, c("Raw data", "First derivative", "Second derivative"),
       pch = rep(20, 3), col = c(1, 2, 4))
```

---

plot.der *Plot* der *objects*

Description

Plot interpolate derivatives.

Usage

```
## S4 method for signature 'der'
plot(x, what = 1:3, add = FALSE, legend = TRUE,
        plot.colors = c("black", "red", "blue"), ...)
```

Arguments

x	is a der object.
what	is a vector specifying what should be plotted. 1 means raw data, 2 means first derivative and 3 means second derivative. Any combination of mentioned values is valid, for example: c(1,2), c(1, 2, 3), c(2, 3). See Details and Examples.
add	is a "logical" argument. If TRUE, plot is added to existing plot.
legend	is a "logical" argument. If TRUE, legend is added to the plot.
plot.colors	is a vectors of colors used in plot. Must have length 3.
...	Arguments to be passed further to the plot function, such as graphical parameters.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

der

Examples

```
res <- inder(AmpSim(cyc = 1:40, Cq = 25))
plot(res)
#round the result of inder
plot(rounder(res))
```

plot.eff *Plot* eff *objects*

Description

Draw quick diagnostic plots of the amplification efficiency.

Usage

```
## S4 method for signature 'eff'
plot(x, xlab = "log10(Concentration)",
                      ylab = "Cq", main = "Efficiency Plot",
                      trend = TRUE, res.fit = TRUE, CI = FALSE,
                      level = 0.95, type = "p", pch = 19,
                      er.length = 0.05, col = "black")
```

Arguments

x	is a eff object.
xlab	a title for the x axis.
ylab	a title for the y axis.
main	an overall title for the plot.
trend	Setting the option trend = TRUE shows the linear regression line in the plot.
res.fit	Setting the option res.fit = TRUE shows the results (goodness of fit, amplification efficiency, correlation) of the linear regression line in the plot.
CI	Setting the option CI = TRUE shows the confidence interval lines in the plot.
level	Tolerance/confidence level.
type	is a graphical parameter setting the plot use lines, points or both (see plot).
pch	is a graphical parameter used to define the symbol used in the plot.
er.length	length is a graphical parameter used to define the length of the error bar used in the plot.
col	col is a graphical parameter used to define the color of the points on the plot.

Details

The plot being a result of this function is built from three subplots.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

See Also

eff

---

plot.refMFI *Plot* refMFI *objects*

Description

Draw quick diagnostic plots of amplification reaction.

Usage

```
## S4 method for signature 'refMFI'
plot(x, CV = FALSE, type = "p", pch = 19, length = 0.05,
                        col = "black")
```

Arguments

x	is a refMFI object. refMFI means referenced Mean Fluorescence Intensity (Roediger et al. 2013)
CV	If CV is true the coefficient of variation (RSD, CV) is plotted. If set to FALSE the deviation as Standard Deviation or Median Absolute Deviation is plotted.
type	is a graphical parameter setting the plot use lines, points or both (see plot).
pch	is a graphical parameter used to define the symbol used in the plot.
length	length is a graphical parameter used to define the length of the error bar used in the plot.
col	col is a graphical parameter used to define the length of the error bar used in the plot.

Details

The plot being a result of this function is built from three subplots.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

See Also

refMFI

Examples

```
#step-by-step plotting refMFI object
```

`plotCurves` *Plot Curves in an Orthogonal Matrix*

Description

Plots many curves on one plot in separate cells allowing quick assessment.

Usage

```
plotCurves(x, y, cyc = 1, fluo = 2:ncol(x), one.plot = FALSE,
           nrow = ceiling(sqrt(ncol(y))), CPP = FALSE, ...)
```

Arguments

`x`	is the column of a data frame for the cycle or data.frame/matrix with whole data.
`y`	are multiple columns of fluorescence values from a `data.frame` (e.g., [, c(1:n)]).
`cyc`	is the index of column containing the cycle data. Used only if `y` is `NULL`.
`fluo`	are the columns containing the fluorescence data. Used only if `y` is `NULL`.
`one.plot`	`logical`, curves may be plotted on single chart (`TRUE`) or in matrix-like multi-plot (`FALSE`).
`nrow`	number of rows in plot. Applies only if `one.plot` is `FALSE`.
`CPP`	logical, if `TRUE` CPP analysis is added to a plot. Applies only if `one.plot` is `FALSE`.
`...`	additional arguments to `plot` function.

Details

`plotCurves` is a function for the quick assessment of amplification curve raw data in an orthogonal matrix.

Value

None.

Author(s)

Stefan Roediger, Michal Burdukiewicz, Konstantin A. Blagodatskikh

Examples

```
# First example
plotCurves(VIMCFX96_60[, 1], VIMCFX96_60[, 2L:16], type = "l")

# Second example
y <- VIMCFX96_60[, 2L:16]
# Introduce some missing values.
y[c(1,4,5,6,23,34), c(2,4,9,15)] <- NA
plotCurves(VIMCFX96_60[, 1], y, nrow = 4, type = "l")

# Third example
# Same as second example but the CPP option is set to TRUE.
# Noise and missing values will be removed.

y <- VIMCFX96_60[, 2L:16]
# Introduce some missing values.
y[c(1,4,5,6,23,34), c(2,4,9,15)] <- NA
plotCurves(VIMCFX96_60[, 1], y, nrow = 4, CPP = TRUE, type = "l")

# Fourth example
plotCurves(VIMCFX96_60, y = NULL, one.plot = TRUE, type = "l")
plotCurves(VIMCFX96_60, y = NULL, one.plot = FALSE, type = "l")
```

---

refMFI *Class* "refMFI"

Description

An S4 class containing the output MFIaggr function. refMFI means referenced Mean Fluorescence Intensity (Roediger et al. 2013)

Slots

`.Data`: "matrix" containing the "Cycle", "Location" (mean, median), "Deviation" (standard deviation, median absolute deviation), "Coefficient of Variance" (CV, RSD) sequential in the columns.

`density`: "density" containing results of the density analysis.

`qqnorm.data`: "data.frame" containing data required for Quantile-Quantile plots.

`stats`: "numeric" vector containing general statistics.

Methods

qqplot `signature(y = "refMFI")`: plots a normal QQ plot. See qqnorm

qqline `signature(y = "refMFI")`: adds a line to QQ plot. See qqline

plot `signature(x = "refMFI")`: plots the object. See plot.refMFI

show `signature(object = "refMFI")`: prints only `.Data` slot of the object.

summary `signature(object = "refMFI")`: prints general statistics and allows easy access to `stats` slot. See `summary.refMFI`

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

See Also

`MFIaggr`, `plot.refMFI`, `summary.refMFI`

Examples

```
#PUT STH HERE
```

`rounder` *Round der objects*

Description

Function `inder` calculates numeric derivatives on smoothed data which results in data points not observable in reality. The `rounder` function averages such result to the real values of cycle number.

Usage

```
rounder(object, cyc = 1)
```

Arguments

`object`	a class `der` object.
`cyc`	a column containing cycle numbers after smoothing. In case of objects created by the `inder` function it is the first column.

Value

An object of the class `der` containing cycle number and averaged values of the fluorescence, first derivative and second derivative.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

inder, der

Examples

```
isPCR <- AmpSim(cyc = 1:40)
res <- inder(isPCR)
rd <- rounder(res)
plot(rd)
```

---

smoother *Wrapper for Several Smoothers of Amplification Data*

Description

Smoother is a wrapper for several smoothing functions including LOWESS, Moving Average, Friedman's SuperSmoother, Cubic Spline and Savitzky-Golay smoothing filter, Friedman's Super-Smoother, and Whittaker smoother for amplification curve data.

Usage

```
## S4 method for signature 'numeric,numeric'
smoother(x, y, trans = FALSE,
    bg.outliers = FALSE, method = "savgol",
    CPP = TRUE, paralell = NULL)
## S4 method for signature 'matrix,missing'
smoother(x, y, trans = FALSE,
    bg.outliers = FALSE, method = "savgol",
    CPP = TRUE, paralell = NULL)
## S4 method for signature 'data.frame,missing'
smoother(x, y, trans = FALSE,
    bg.outliers = FALSE, method = "savgol",
    CPP = TRUE, paralell = NULL)
```

Arguments

x	x is a vector containing values for smoothing or data frame/matrix with values for smoothing.
y	y values for smoothing. Used only if x is also a vector.
trans	perform a linear transformation based on the trend of the background range.
bg.outliers	logical parameter which indicates of outliers should be removed from background range.

method	a list where each element is character vector representing a smoothing method or a named list of additional arguments to a smoothing algorithm. See Examples section. The Savitzky-Golay smoothing filter is the default smoother. Use "lowess" for LOWESS smoother (locally-weighted polynomial regression, "mova" for moving average, "savgol" for Savitzky-Golay smoothing filter, "smooth" for cubic spline smooth, "spline" for standard cubic spline smooth, "supsmu" for Friedman's SuperSmoother, "whit1" for weighted Whittaker smoothing with a first order finite difference penalty, "whit2" for weighted Whittaker smoothing with a second order finite difference penalty or "all" for all implemented smoothing algorithms. Both upper and lower case names are accepted.
CPP	logical parameter which indicates if CPP (curve pre-procssor) should be used.
paralell	should contain a cluster object, created by package parallel or snow package. If NULL, no parallelization is used.

Details

Amplification curve data of experimental thermo-cyclers may deliver results which are hard to interpret due to noise and scatter. For data presentation it is often useful to smooth or filter the data prior to presentation. Smoothing and filtering are different approaches with a similar outcome to preprocess an input signal in order to make it available for an analysis step. Filtering uses methods of signal processing. They take a data input and apply a function to form an output. There are linear and non-linear filters. The most common example of a linear filter is the the moving average. A moving average filter replaces sequentially data points with the average of the neighbor data points. The average is calculated from a defined span ("window") of odd count (e.g., 3, 5). The average herein may also refer to the median, the geometric or exponential mean. Smoothing in contrast uses statistical approaches. Such approaches use for example local regression models (e.g., least squares estimate) or cubic splines. Splines apply non-parametric regression by local cubic polynomials between knot points. Other examples for smoothers include Savitzky-Golay smoothing filter, Friedman's SuperSmoother, and Whittaker smoother. Several methods were integrated in the chipPCR package.

The function smoother is a wrapper for smoother functions and filters commonly used to process amplification curve data. The smoother function was enhanced by functionality of the fixNA and CPP functions. The parameter "lowess" for LOWESS smoother (locally-weighted polynomial regression) can be tuned by the parameters f and iter (see lowess for details). The parameter "mova" for moving average can be tuned by the parameter movaww. movaww is the window size used for the moving average (see filter for details). The parameter "savgol" for Savitzky-Golay smoothing filter can be tuned by the parameter p (see sgolayfilt for details). The parameter "smooth" for cubic spline smooth can be tuned by the parameter df.fact. A df.fact value of 1 will leave the raw data almost unaffected while a value 0.5 will smooth the curve considerably. For further details refer to the smooth.spline function. The parameter "spline" for standard cubic spline smooth has currently no additional parameter. The parameter "supsmu" for Friedman's SuperSmoother can be tuned by the parameter span. For further details refer to the supsmu function. The parameter "lambda" is used in Weighted Whittaker smoothing.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

Examples

```
# Results of different smoothers. A in-silico amplification was performed
# using the AmpSim function and different smoothers were applied. Optimally
# all smoothers should give the same result (which is not the case)).
# refMFI means referenced Mean Fluorescence Intensity
# (Roediger et al. 2013)
tmp <- AmpSim(cyc = 1:35, bl = 0)

plot(tmp, main = "In-silico real-time PCR\n Effect of Smoother",
     xlab = "Cycles", ylab ="refMFI", ylim = c(0,1), pch = 20,
     type = "b", lwd = 2)

legend(25, 0.8, c("Raw data", "savgol", "lowess", "mova 3", "mova 5",
  "smooth", "spline", "supsmu"), pch = 20, lwd = 2,
  col = c(1:8))

## Not run:
#on machines with multiple cores
library(parallel)
cl <- makeCluster(getOption("cl.cores", 2))
tmp.smooths <- smoother(tmp, method = list("savgol",
                                           "lowess",
                                           mova = list(movaww = 3),
                                           mova = list(movaww = 5),
                                           "smooth",
                                           "spline",
                                           "supsmu"),
                        parallel = TRUE)
stopCluster(cl)

## End(Not run)
#else
tmp.smooths <- smoother(tmp, method = list("savgol",
                                           "lowess",
                                           mova = list(movaww = 3),
                                           mova = list(movaww = 5),
                                           "smooth",
                                           "spline",
                                           "supsmu"))
for (i in 1:ncol(tmp.smooths))
  lines(tmp[, 1], tmp.smooths[, i], type = "b", pch = 20, lwd = 2, col = i
+ 1)


par(fig = c(0.15,0.6,0.45,0.99), new = TRUE)
plot(tmp, main = "", xlab = "Cycles", ylab ="refMFI",
     pch = 20, xlim = c(14,20), ylim = c(0,0.45))
```

```
for (i in 1:ncol(tmp.smooths))
  lines(tmp[, 1], tmp.smooths[, i], type = "b", pch = 20, lwd = 2, col = i
+ 1)


# Plot the difference of the smoothed / filtered data
# to the raw data against the cycles
# The largest error is in the transition phases between
# start and end of the detectable amplification process.
par(fig = c(0,1,0,0.65))
plot(NA, NA, type = "b", col = 2, pch = 20, xlim = c(1,35),
      ylim = c(-0.1,0.1), xlab = "Cycle",
      ylab = "delta refMFI (raw - smoothed)",
     main = "Smoothed / Filtered data")

legend(1.5, 0.1, ncol = 2, c("savgol", "lowess", "mova 3", "mova 5",
"smooth", "spline", "supsmu"), pch = 20, lwd = 2,
col = c(2:8))

for (i in 1:ncol(tmp.smooths))
  lines(tmp[, 1], tmp[, 2] - tmp.smooths[, i], type = "b", pch = 20, lwd =
2, col = i + 1)

par(fig = c(0,1,0.55,1), new = TRUE)

plot(tmp, type = "b", col = 1, pch = 20, xlab = "", ylab = "RFU",
      main = "Raw data")

#different ways of using smoother
#1. single method
single.smooth <- smoother(tmp, method = list("mova"))
#single smooth, additional argument specified
single.smooth.add <- smoother(tmp, method = list(mova = list(movaww = 3)))
#3. more than one smoothing method, no additional arguments specified
double.smooth <- smoother(tmp, method = list("savgol", "mova"))
#4. more than one smoothing method, additional arguments specified
double.smooth.add <- smoother(tmp, method = list("savgol", mova =
list(movaww = 3)))
#5. all smoothing methods, no additional arguments specified
all.smooth <- smoother(tmp, method = list("all"))
```

---

summary-bg *Summary* bg *objects*

Description

Summarize bg objects and easily access results of bg function.

Usage

```
## S4 method for signature 'bg'
summary(object, print = TRUE)
```

Arguments

object	is a bg object.
print	is a logical value determining if pretty summary of the object should be printed.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

bg

Examples

```
res <- AmpSim(cyc = 1:40, Cq = 25)
background <- bg.max(res[, 1], res[, 2])
#just print summary
summary(background)

#assign summary to variable without printing
vals <- summary(background, print = FALSE)
print(vals)

#easily access different values
vals["FDM"]
```

---

summary-der *Summary* der *objects*

Description

Summarize der objects and easily access results of inder function.

Usage

```
## S4 method for signature 'der'
summary(object, digits = getOption("digits") - 3, print = TRUE)
```

Arguments

object	is a der object.
digits	is a numeric value determining the number of decimal places. Used only for printed values. See format.
print	is a logical value determining if pretty summary of the object should be printed.

Details

The approximate second derivative maximum (SDM) which is commonly used to quantify quantitative real-time PCR experiments. The SDM might also be useful for isothermal amplification processes. The SDM is calculated from a derived cubic spline. Similarly the first approximate approximate derivative maximum (FDM), second derivative minimum (SDm), and approximate second derivative center (SDC, geometric mean of SDM and SDm) are available. FDM, SDm and SDC values can be used to further characterize the amplification process.

Value

A named vector of four elements. The element named 2nd_der contains values of the second first derivative. The element named SDM contains approximate second derivative maximum, SDm contains approximate second derivative minimum, and SDC contains approximate second derivative center.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Ruijter JM, Pfaffl MW, Zhao S, et al. (2013) Evaluation of qPCR curve analysis methods for reliable biomarker discovery: bias, resolution, precision, and implications. *Methods* San Diego Calif 59:32–46.

See Also

der

Examples

```
res <- AmpSim(cyc = 1:40, Cq = 25)
test.der <- inder(res)
#just print summary
summary(test.der)
#print without rounding
summary(test.der, digits = 6)

#assign summary to variable without printing and
vals <- summary(test.der, print = FALSE)
print(vals)

#easily access different values
```

```
vals["bg.start"]
```

---

summary-refMFI *Summary* refMFI *objects*

Description

Summarize refMFI objects and access general statistics of amplification reaction. refMFI means referenced Mean Fluorescence Intensity (Roediger et al. 2013).

Usage

```
## S4 method for signature 'refMFI'
summary(object, digits = getOption("digits") - 3, print = TRUE)
```

Arguments

object	is a refMFI object.
digits	is a numeric value determining the number of decimal places. Used only for printed values. See format.
print	is a logical value determining if pretty summary of the object should be printed.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Roediger S, Boehm A, Schimke I. Surface Melting Curve Analysis with R. *The R Journal* 2013;5:37–53.

See Also

refMFI

Examples

```
res <- AmpSim(cyc = 1:50, Cq = 41)
summary(inder(res), print = TRUE)
```

th *Class* "th"

Description

An S4 class containing the output th.cyc function.

Slots

.Data: "matrix" is a matrix containing the threshold cycle and threshold fluorescence.

stats: "summary.lm" contains linear model used for Ct estimation.

input: "matrix" input data for linear model.

Methods

summary signature(object = "th"): prints summary of the object.

show signature(object = "th"): prints only .Data slot of the object.

Author(s)

Stefan Roediger, Michal Burdukiewicz

See Also

th.cyc

Examples

```
res <- th.cyc(VIMCFX96_69[, 1], VIMCFX96_69[, 3], r = 2300)
summary(res)
slot(res, "input")
```

th.cyc *Threshold Cycle*

Description

th.cyc is a function to calculate the number of cycles at which the fluorescence exceeds a defined threshold, called the threshold cycle (Ct). According to the MIQE guidelines the Ct is referred to as quantification cycle (Cq). The calculated Cq is a relative value, which depends on the template copy number, instrument, reagents, amplification efficiency and probe technology. Low Cqs correlate with high quantities template copy numbers. Real-time technologies enable the quantification of nucleic acids by calculation of specific curve parameters like the quantification point (Cq) and the amplification efficiency (AE) based on the kinetics of the amplification curve. The Cq represents the number of cycles (time for qIA) needed to reach a defined fluorescence signal level in the exponential phase of the amplification curve. The Cq can be determined from a fixed threshold value or by various analytical algorithm as described elsewhere (Bustin et al. 2009, Ruijter et al. 2013, Tellinghuisen et al. 2014).

Usage

```
th.cyc(x, y, r = 2500, auto = FALSE, linear = TRUE)
```

Arguments

x	is a vector containing the time or cycle values.
y	is a vector containing the fluorescence values.
r	a fluorescence value which defines the threshold.
auto	is logical parameter which indicates if an automatic estimation of the threshold should be used (Note: Experimental, not save to use).
linear	is logical parameter which indicates if a linear or quadratic regression should be used for the calculation. (Note: Experimental, not save to use).

Details

The Threshold Cycle (Ct) (Cq according to MIQE, see Bustin et al. 2009) is the cycle number at which the fluorescence exceeds significantly a point above the baseline and defined threshold in a particular samples. Thus the Ct is the cycle when sufficient numbers of amplicons have accumulated. The th.cyc calculates the intersection of the user defined Ct value (r) and a linear regression or quadratic polynomial in the range of the user defined Ct value. In contrast to other methods is does th.cyc have no requirement to fit a "complex" non linear model to the entire data set but rather focuses on the specific area. The polynomial is calculated from four neighbor values at the fluorescence threshold.

Author(s)

Stefan Roediger, Michal Burdukiewicz

References

Stephen A. Bustin, Vladimir Benes, Jeremy A. Garson, Jan Hellemans, Jim Huggett, Mikael Kubista, Reinhold Mueller, Tania Nolan, Michael W. Pfaffl, Gregory L. Shipley, Jo Vandesompele, and Carl T. Wittwer. (Apr 2009). "The MIQE Guidelines: Minimum Information for Publication of Quantitative Real-Time PCR Experiments". *Clin Chem.* 55 (4):611–22. doi:10.1373/clinchem.2008.112797. PMID 19246619

Ruijter, J.M., Pfaffl, M.W., Zhao, S., Spiess, A.N., Boggy, G., Blom, J., Rutledge, R.G., Sisti, D., Lievens, A., De Preter, K., Derveaux, S., Hellemans, J., Vandesompele, J.: Evaluation of qPCR curve analysis methods for reliable biomarker discovery: bias, resolution, precision, and implications. *Methods (San Diego, Calif.)* 59(1), 32–46 (2013). doi:10.1016/j.ymeth.2012.08.011. PMID: 22975077

Tellinghuisen, J., Spiess, A.-N.: Comparing real-time quantitative polymerase chain reaction analysis methods for precision, linearity, and accuracy of estimating amplification efficiency. *Analytical Biochemistry* 449, 76–82 (2014). doi:10.1016/j.ab.2013.12.020. PMID: 24365068

Examples

```
# First example
# Raw data from the VIMCFX96_69 data set.
# Cycles
x <- VIMCFX96_69[, 1]
# Fluoresce values
y <- VIMCFX96_69[, 2]

# Plot the raw data
plot(x, y, xlab = "Cycle", ylab = "Fluo")
# Calculate the the Ct value
res <- th.cyc(x, y, r = 2300)
lines(res@input, col = 2, lwd = 2)
# Threshold fluorescence value
abline(h = res@.Data[2], col = 3)
# Calculated Ct value
abline(v = res@.Data[1], col = 4)

# Second example
# Application of the th.cyc method to determine the Cq from a continous
# amplification reaction.
plot(NA, NA, xlim = c(0,80), ylim = c(0,1200), xlab = "Time [min]",
     ylab = "Voltage [micro V]", main = "ccPCR - Raw Data")

# Threshold level "r" (50 micro Volts)
for (i in c(1,3,5,7)) {
  y.tmp <- capillaryPCR[, i + 1] - mean(capillaryPCR[1L:150, i + 1])
  Ct.tmp <- th.cyc(capillaryPCR[, i], y.tmp, r = 50, linear = FALSE)
  abline(v = Ct.tmp[1])
  text(Ct.tmp[1] * 1.1, 1200, paste(round(Ct.tmp[1], 1), "\nmin"))
  lines(capillaryPCR[, i], y.tmp, type = "b", pch = 20 - i)
  points(Ct.tmp@input, col = "red", pch = 19)
}
abline(h = 50)
legend(5,800, c("Run 1", "Run 2", "Run 3", "Control"), pch = c(19, 17, 15, 13),
      lwd = 1.5)
```

VIMCFX96_60 *Amplfication Reaction Using the Bio-Rad CFX96*

Description

Data set of an amplfication reaction using the Bio-Rad CFX96 thermo cycler. The samples of Vimentin were amplified in the CFX96 as replicates according to Roediger et al. (2013). The qunatitiation was performed during the annealing step (60 deg Celsius).

Usage

```
data(VIMCFX96_60)
```

Format

A data frame with 40 observations on the following 97 variables. The first column ("Cycle") contains the number of cylces and consecutive columns contain the replicates ("A1" to "H12").

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(VIMCFX96_60)
data(VIMCFX96_69)

T60 <- rowMeans(VIMCFX96_60[, 2:ncol(VIMCFX96_60)])
T69 <- rowMeans(VIMCFX96_69[, 2:ncol(VIMCFX96_69)])

plot(1:length(T60), T60, main = "Fluorescence at different
      temperatures\nQuantitation in CFX96 (Bio-Rad)", xlab = "Cycle",
      ylab = "Cycle dependent fluorescence", pch = 15, type = "b")
lines(1:length(T69), T69, pch = 19, type = "b", col = 2)
legend(1, 4500, c("Annealing (60 deg C)", "Elongation (69 deg C)"),
pch = c(15, 19), col = c(1,2))
```

VIMCFX96_69 *Amplfication Reaction Using the Bio-Rad CFX96*

Description

Data set of an amplfication reaction using the Bio-Rad CFX96 thermo cycler. The samples of Vimentin were amplified in the CFX96 as replicates according to Roediger et al. (2013). The qunatitiation was performed during the elongation step (69 deg Celsius).

Usage

```
data(VIMCFX96_69)
```

Format

A data frame with 40 observations on the following 97 variables. The first column ("Cycle") contains the number of cylces and consecutive columns contain the replicates ("A1" to "H12").

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(VIMCFX96_60)
data(VIMCFX96_69)

T60 <- rowMeans(VIMCFX96_60[, 2:ncol(VIMCFX96_60)])
T69 <- rowMeans(VIMCFX96_69[, 2:ncol(VIMCFX96_69)])

plot(1:length(T60), T60, main = "Fluorescence at different
      temperatures\nQuantitation in CFX96 (Bio-Rad)", xlab = "Cycle",
      ylab = "Cycle dependent fluorescence", pch = 15, type = "b")
lines(1:length(T69), T69, pch = 19, type = "b", col = 2)
legend(1, 4500, c("Annealing (60 deg C)", "Elongation (69 deg C)"),
pch = c(15, 19), col = c(1,2))
```

VIMCFX96_meltcurve *Melting Curve Measured with the Bio-Rad CFX96*

Description

Data set of a melting curve using the Bio-Rad CFX96 thermo cycler. The samples of Vimentin were measured in the CFX96 as replicates according to Roediger et al. (2013). The qunatitiation was performed during the gradient from 55 to 95 deg Celsius with a resolution of 0.5 deg Celsius per step.

Usage

```
data(VIMCFX96_meltcurve)
```

Format

A data frame with 81 observations on the following 97 variables. The first column ("Temperature") contains the temperature and consecutive columns contain the replicates ("A1" to "H12").

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(VIMCFX96_meltcurve)
tmp <- VIMCFX96_meltcurve

plot(NA, NA, xlim = c(55,95), ylim = c(2000, 7000), xlab = "Temperature
    (deg Celsius)",
    ylab = "RFU", main = "Melting curve in CFX96 (Bio-Rad)")
apply(tmp[, 2:ncol(tmp)], 2,
      function(x) lines(tmp[1:nrow(tmp),1],x))

Fmean <- rowMeans(tmp[, 2:ncol(tmp)])
lines(tmp[1:nrow(tmp),1], Fmean, col = "red", lwd = 3)

legend(55, 4000, c("Raw", "Mean"), pch = c(19,19), col = c(1,2))
```

VIMiQ5_595 *Amplfication Reaction Using the Bio-Rad iQ5*

Description

Data set of an amplfication reaction using the Bio-Rad iQ5 thermo cycler. The samples of Vimentin were amplified in the iQ5 as replicates according to Roediger et al. (2013). The qunatitiation was performed during the annealing step (59.5 deg Celsius).

Usage

```
data(VIMiQ5_595)
```

Format

A data frame with 40 observations on the following 97 variables. The first column ("Cycles") contains the number of cylces and consecutive columns contain the replicates ("A01" to "H12").

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
T595 <- rowMeans(VIMiQ5_595[, 2:ncol(VIMiQ5_595)])
T685 <- rowMeans(VIMiQ5_685[, 2:ncol(VIMiQ5_685)])

plot(1:length(T595), T595, main = "Fluorescence at different
    temperatures\nQuantitation in iQ5 (Bio-Rad)", xlab = "Cycle",
    ylab = "Cycle dependent fluorescence", pch = 15, type = "b")
lines(1:length(T685), T685, pch = 19, type = "b", col = 2)
legend(1, 10000, c("Annealing (59.5 deg C)", "Elongation (68.5 deg C)"),
pch = c(15, 19), col = c(1,2))
```

VIMiQ5_685 *Amplfication Reaction Using the Bio-Rad iQ5*

Description

Data set of an amplfication reaction using the Bio-Rad iQ5 thermo cycler. The samples of Vimentin were amplified in the iQ5 as replicates according to Roediger et al. (2013). The qunatitiation was performed during the elongation step (68.5 deg Celsius).

Usage

```
data(VIMiQ5_685)
```

Format

A data frame with 40 observations on the following 97 variables. The first column ("Cycles") contains the number of cylces and consecutive columns contain the replicates ("A01" to "H12").

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
T595 <- rowMeans(VIMiQ5_595[, 2:ncol(VIMiQ5_595)])
T685 <- rowMeans(VIMiQ5_685[, 2:ncol(VIMiQ5_685)])

plot(1:length(T595), T595, main = "Fluorescence at different
      temperatures\nQuantitation in iQ5 (Bio-Rad)", xlab = "Cycle",
      ylab = "Cycle dependent fluorescence", pch = 15, type = "b")
lines(1:length(T685), T685, pch = 19, type = "b", col = 2)
legend(1, 10000, c("Annealing (59.5 deg C)", "Elongation (68.5 deg C)"),
      pch = c(15, 19), col = c(1,2))
```

VIMiQ5_melt *Melting Curve Measured with the Bio-Rad iQ5*

Description

Data set of a melting curve using the Bio-Rad iQ5 thermo cycler. The samples of Vimentin were measured in the CFX96 as replicates according to Roediger et al. (2013). The qunatitiation was performed during the gradient from 55 to 95 deg Celsius with a resolution of 0.5 deg Celsius per step.

Usage

```
data(VIMiQ5_melt)
```

Format

A data frame with 81 observations on the following 97 variables. The first column ("T") contains the temperature and consecutive columns contain the replicates ("A01" to "H12").

Source

Stefan Roediger, Clauia Deutschman (BTU Cottbus - Senftenberg)

References

A Highly Versatile Microscope Imaging Technology Platform for the Multiplex Real-Time Detection of Biomolecules and Autoimmune Antibodies. S. Roediger, P. Schierack, A. Boehm, J. Nitschke, I. Berger, U. Froemmel, C. Schmidt, M. Ruhland, I. Schimke, D. Roggenbuck, W. Lehmann and C. Schroeder. *Advances in Biochemical Bioengineering/Biotechnology*. 133:33–74, 2013. http://www.ncbi.nlm.nih.gov/pubmed/22437246

Examples

```
data(VIMiQ5_melt)
tmp <- VIMiQ5_melt

plot(NA, NA, xlim = c(55,95), ylim = c(0, 40000),
    xlab = "Temperature (deg Celsius)",ylab = "RFU",
    main = "Melting curve in iQ5 (Bio-Rad)")
apply(tmp[, 2:ncol(tmp)], 2,
      function(x) lines(tmp[1:nrow(tmp),1],x))

Fmean <- rowMeans(tmp[, 2:ncol(tmp)])
lines(tmp[1:nrow(tmp),1], Fmean, col = "red", lwd = 3)

legend(55, 4000, c("Raw", "Mean"), pch = c(19,19), col = c(1,2))
```

Index