

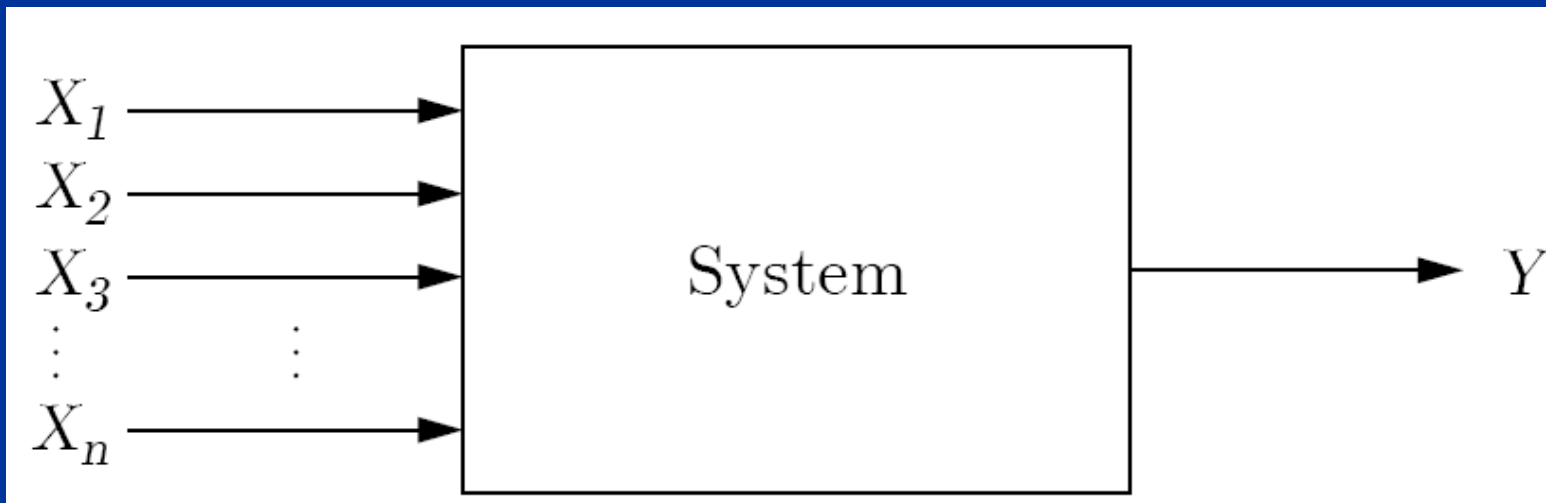
The calculation of real-time PCR ratios by means of Monte Carlo Simulation or high-order Taylor expansion

Andrej-Nikolai Spiess

Department of Andrology,
University Hospital Hamburg-Eppendorf

Do we need error propagation or **WHY THE FUSS...?**

- If we have some input variables X_1, X_2, \dots, X_i , each with some inherent error $\sigma_1, \sigma_2, \dots, \sigma_i$, then we inevitably must calculate (estimate) the error of $Y = f(X_1, X_2, \dots, X_i)$.
- We COULD ignore this, but we shouldn't, because it is good practice to describe the uncertainty of any acquired measurement.



Commonly used methods for estimating the uncertainty of qPCR ratios

- Using the first-order Taylor expansion on the errors estimated from threshold cycles and the efficiency obtained from a dilution calibration curve ([Karlen et al.](#), BMC Bioinformatics 2007, 8:131)
- An approach similar to 1) by using Gaussian error propagation as implemented in the qBase quantification framework ([Hellemans et al.](#), Genome Biology 2007, 8:R19)
- Using a permutation regime by reallocating the threshold cycles between sample and control groups as implemented in the different REST software versions ([Pfaffl et al.](#), NAR 2002, 30:E36)
- An extended Gaussian error propagation approach that also takes the covariance structure between the estimated variables into account ([Nordgard et al.](#), Anal Biochem 2006, 356:182-193.)

The four different approaches for estimating uncertainties

Gaussian error propagation

assumes normality

parametric

computationally (fairly) simple

restricted to small errors (1%)

Permutation approach

distribution independent

stochastic

computationally (fairly) demanding

more tolerant to error range

Monte Carlo simulation

distribution independent

stochastic

computationally very demanding

very tolerant to error range

High-order Taylor expansion

distribution independent

parametric

arithmetically very demanding

more tolerant to error range

The four approaches exemplified... with qPCR ratios

#1: Gaussian error propagation

eff.c	eff.s	ref.C	ref.S	goi.C	goi.S
2.01	1.97	26.74	26.77	27.57	24.54
		26.85	26.47	27.61	24.95
		26.83	27.03	27.82	24.57
		26.68	26.92	27.12	24.63
		27.39	26.97	27.76	24.66
		27.03	26.97	27.74	24.89
		26.78	26.07	26.91	24.71
		27.32	26.3	27.49	24.9
			26.14		24.26
			26.81		24.44

mean	26.95	26.65	27.50	24.66
s.d.	0.27	0.37	0.32	0.22
	1.1%	1.3%	1.2%	0.9%

$$\frac{\text{eff.c}_{(\text{goi.C-goi.S})}}{\text{eff.s}_{(\text{ref.C-ref.S})}}$$

$$\sigma_Y = \sqrt{\sum_i \left(\frac{\partial f}{\partial X_i} \right)^2 \sigma_{X_i}^2}$$

5.93 ± 2.44

41.2%

The four approaches exemplified... with qPCR ratios

#2: Permutation approach (REST)

eff.c	eff.s	ref.C	ref.S	goi.C	goi.S	
2.01	1.97	26.74	26.77	27.57	24.54	8.62
		26.85	26.47	27.61	24.95	2.93
		26.83	27.03	27.82	24.57	5.18
		26.68	26.92	27.12	24.63	9.48
		27.39	26.97	27.76	24.66	7.02
		27.03	26.97	27.74	24.89	4.67
		26.78	26.07	26.91	24.71	4.79
		27.32	26.3	27.49	24.9	5.18
			26.14		24.26	5.18
			26.81		24.44	4.84

$$\frac{\text{eff.c}^{(\text{goi.C-goi.S})}}{\text{eff.s}^{(\text{ref.C-ref.S})}}$$



Ratio = 5.927

Confidence 95% = [2.983; 9.996]

Beware: Only 80 possible combinations, n > 80 repetitive!

The four approaches exemplified... with qPCR ratios

#3: Monte Carlo simulation

eff.c	eff.s	ref.C	ref.S	goi.C	goi.S
2.01	1.97	26.74	26.77	27.57	24.54
		26.85	26.47	27.61	24.95
		26.83	27.03	27.82	24.57
		26.68	26.92	27.12	24.63
		27.39	26.97	27.76	24.66
		27.03	26.97	27.74	24.89
		26.78	26.07	26.91	24.71
		27.32	26.3	27.49	24.9
			26.14		24.26
			26.81		24.44

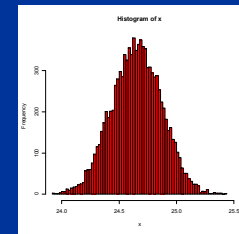
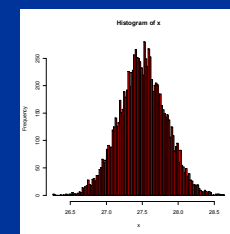
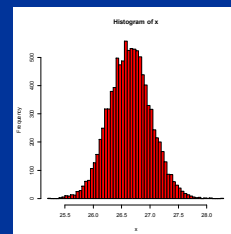
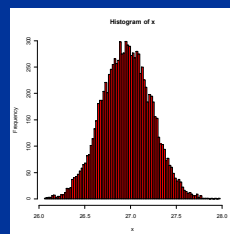
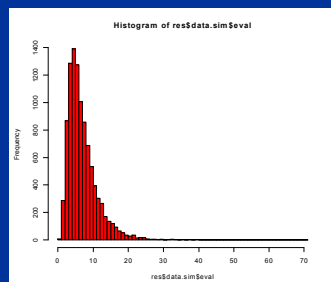
$$\frac{\text{eff.c}^{(\text{goi.C}-\text{goi.S})}}{\text{eff.s}^{(\text{ref.C}-\text{ref.S})}}$$

mean
s.d.

26.95	26.65	27.50	24.66
0.27	0.37	0.32	0.22

6.91 ± 4.22 ?

5.89 ± 3.17 ?



The four approaches exemplified... with qPCR ratios

#4: High-order Taylor expansion

eff.c	eff.s	ref.C	ref.S	goi.C	goi.S
2.01	1.97	26.74	26.77	27.57	24.54
		26.85	26.47	27.61	24.95
		26.83	27.03	27.82	24.57
		26.68	26.92	27.12	24.63
		27.39	26.97	27.76	24.66
		27.03	26.97	27.74	24.89
		26.78	26.07	26.91	24.71
		27.32	26.3	27.49	24.9
			26.14		24.26
			26.81		24.44

mean	26.95	26.65	27.50	24.66
s.d.	0.27	0.37	0.32	0.22

$$\frac{\text{eff.c}^{(\text{goi.C-goi.S})}}{\text{eff.s}^{(\text{ref.C-ref.S})}}$$



$$\sigma_Y^2 = \sum_i^k \frac{1}{k!} (F_{x_i} C_x^k F_{x_i}^T)$$



$$5.93 \pm 2.47$$

$$(5.93 \pm 2.41)$$

Monte Carlo simulation:

Many Random samples that characterize my variable

How does it work?

If I have some variable $X \pm \sigma$, I generate 1000000 datapoints with mean μ and standard deviation σ . I do this for every variable in my formula $f(X_1, X_2, \dots X_i)$. Then I do my calculation with each of my generated datapoints, i.e. $X_1[1\dots 1000000] * X_2[1\dots 1000000]$.

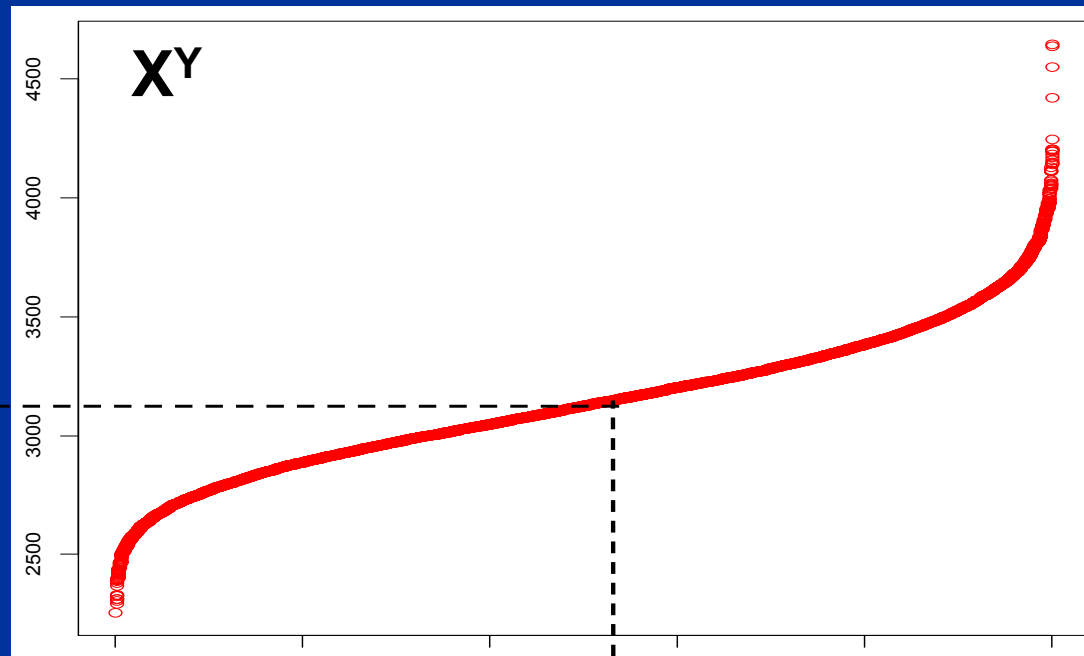
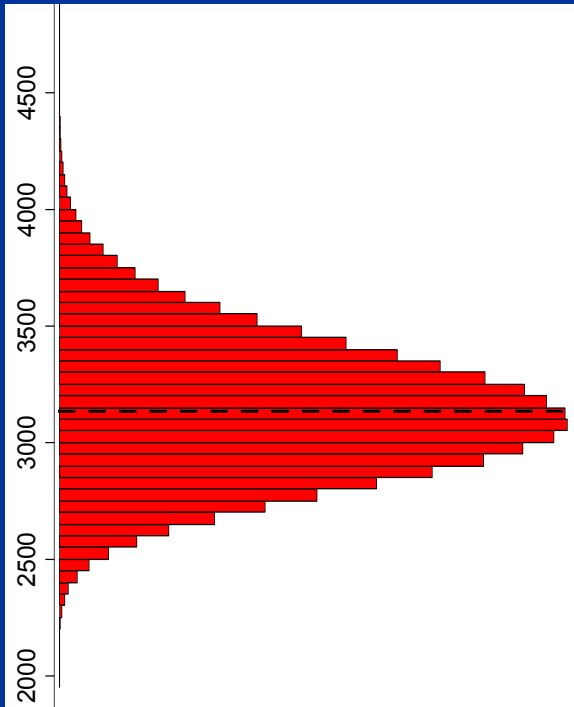
This gives a dataset of 1000000 results. From this dataset I calculate $Y \pm \sigma$.

Advantage:

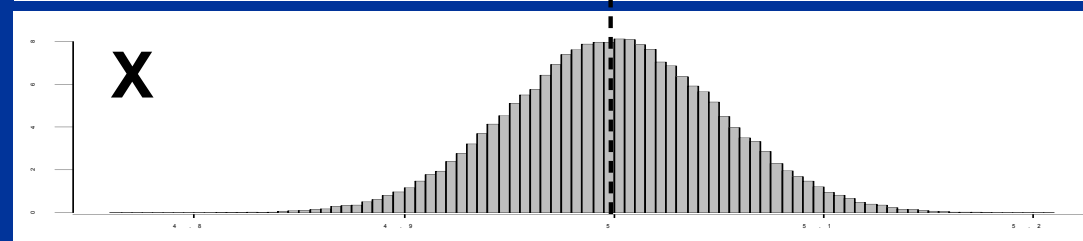
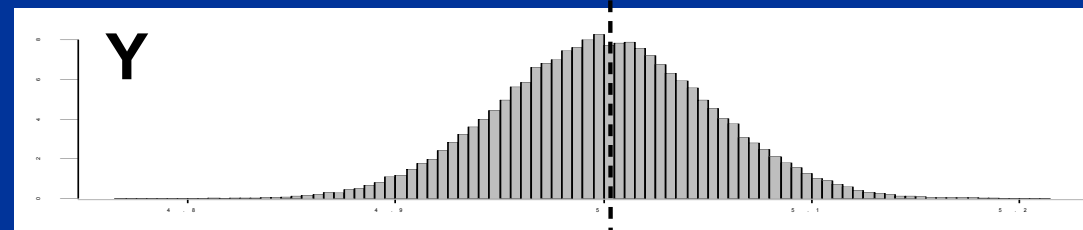
If I plot a histogram of all datapoints, I can see the distribution of $Y[1\dots 1000000]$. This can tell me if the mean and s.d. of Y is a realistic estimate.

Composite plot for XY

100000 random numbers with
mean = 5 and s.d. = 0.05

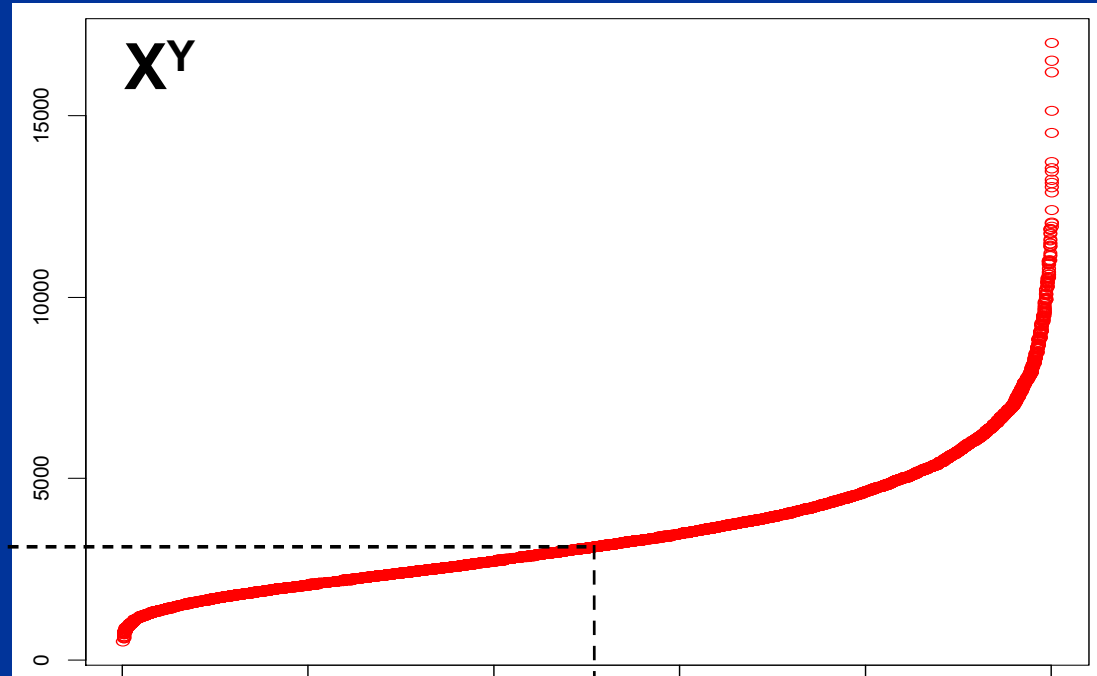
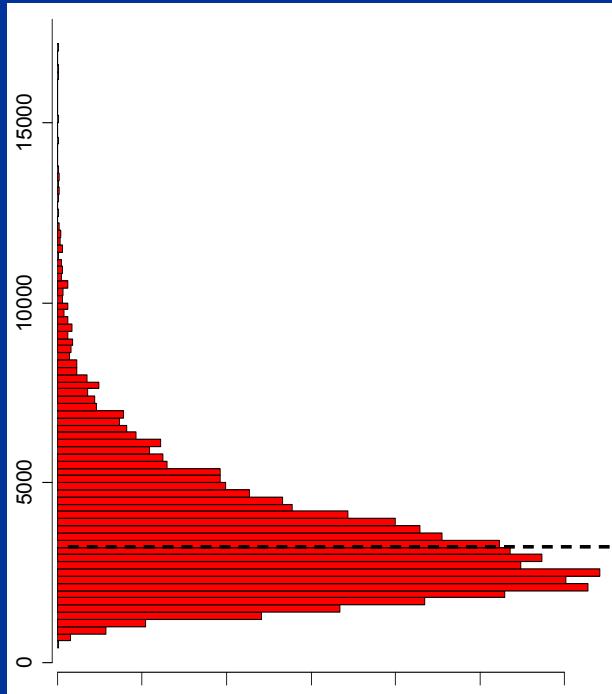


mean = 3140.5 (**3125**)
s.d. = 295.5 (**9.4%**)
skewness = 0.327

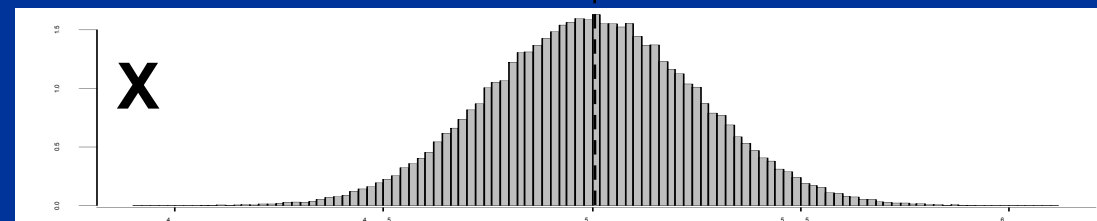
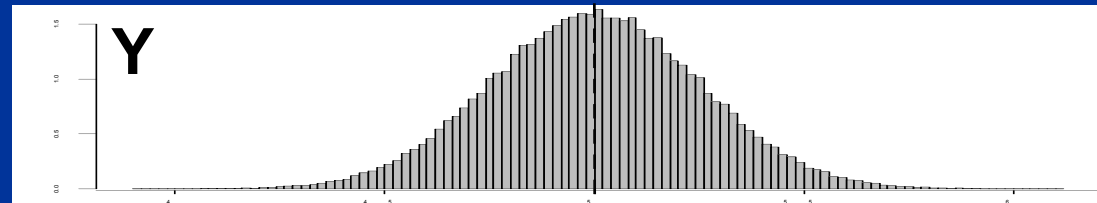


Composite plot for XY

100000 random numbers with
mean = 5 and s.d. = 0.25



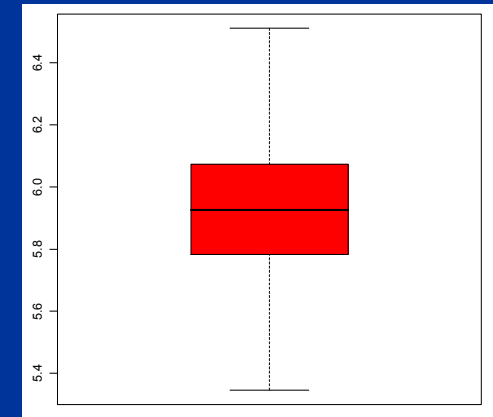
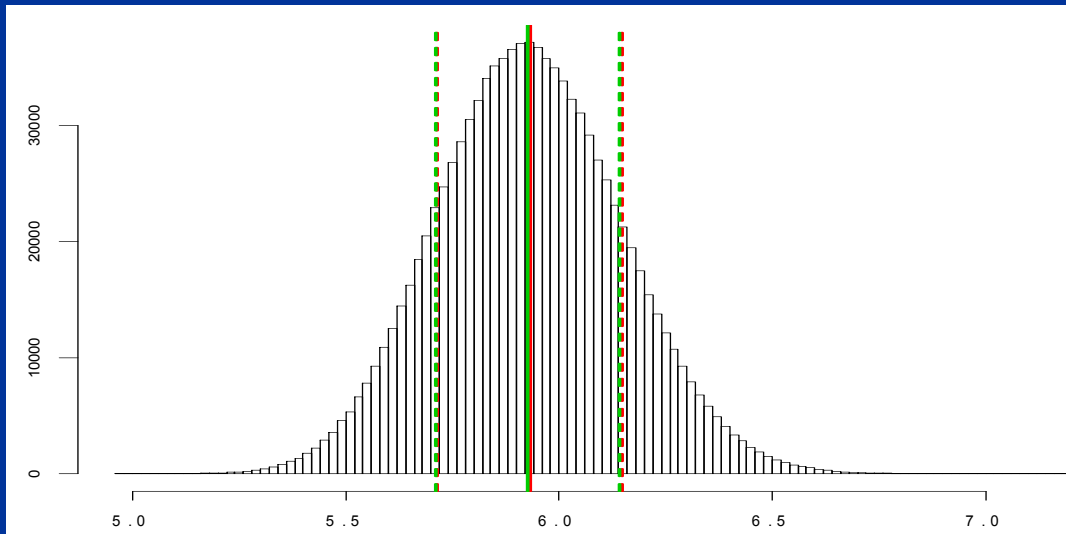
mean = 3487.5 (**3125**)
s.d. = 1744 (**50%**)
skewness = 1.54



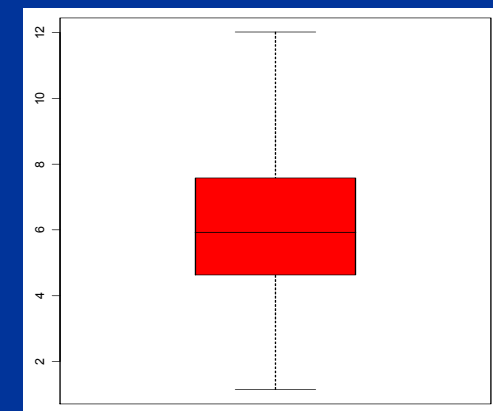
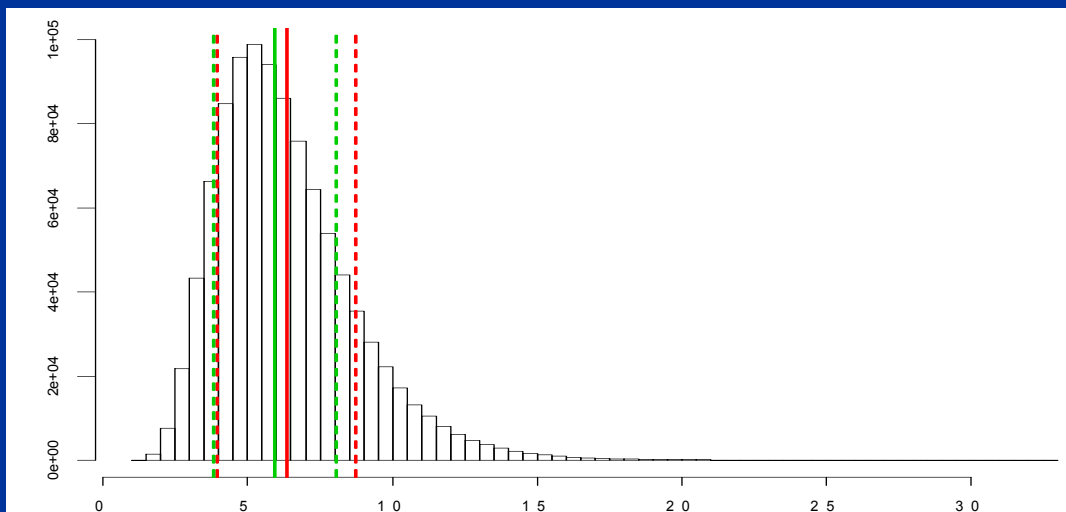
$$\frac{\text{eff.c}^{(\text{goi.C-goi.S})}}{\text{eff.s}^{(\text{ref.C-ref.S})}}$$

Monte Carlo simulation: So what does that mean for ratio calculation?

0.1 % error



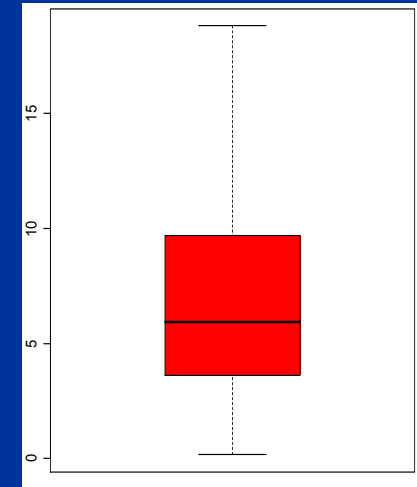
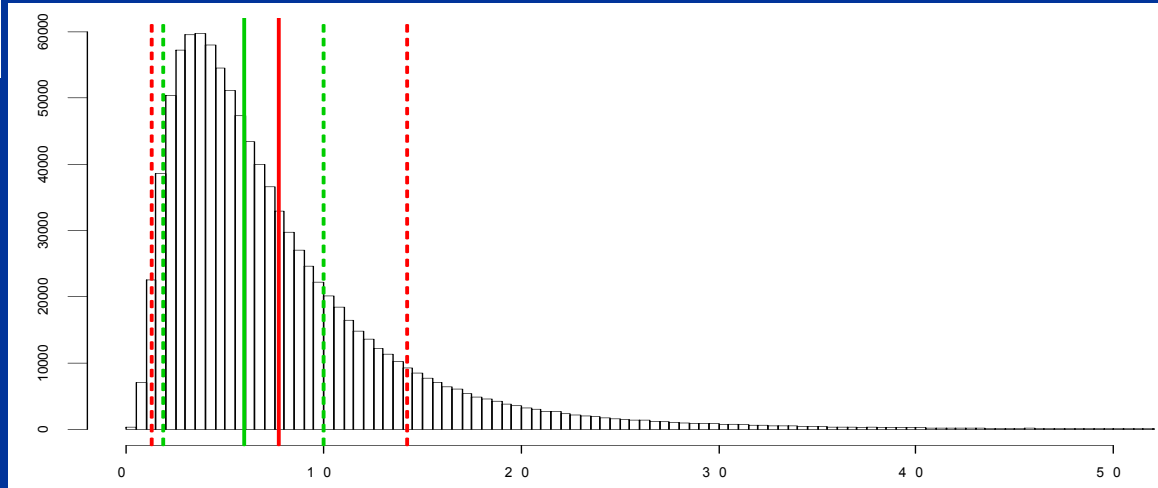
1 % error



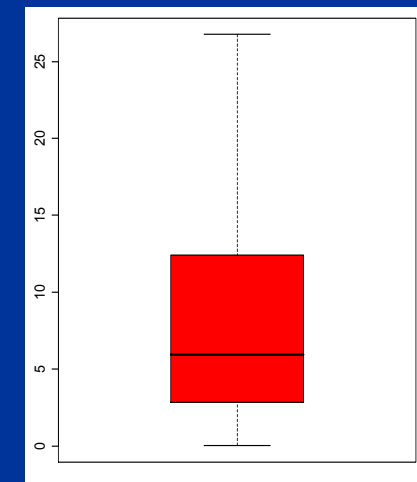
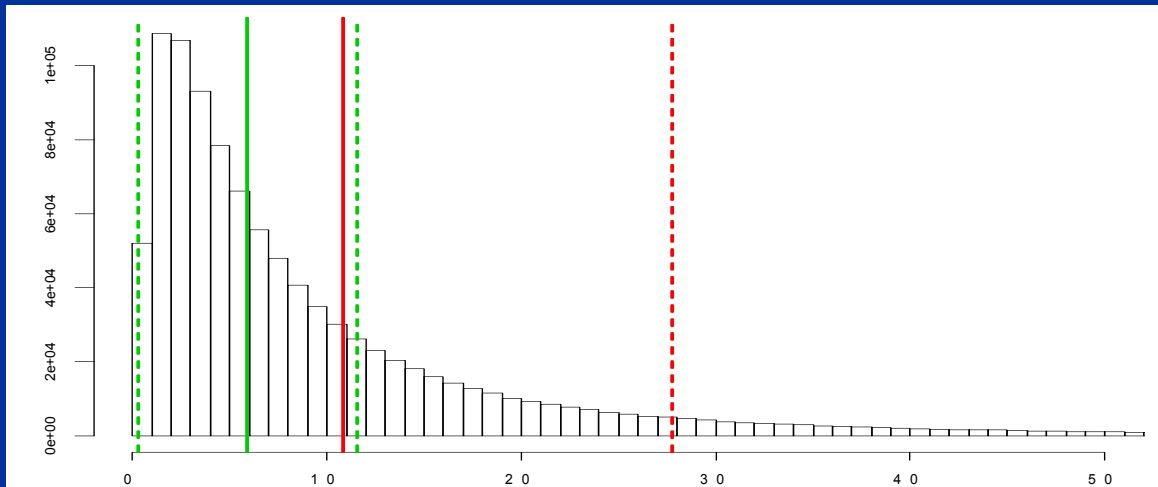
$$\frac{\text{eff.c}^{(\text{goi.C-goi.S})}}{\text{eff.s}^{(\text{ref.C-ref.S})}}$$

Monte Carlo simulation: So what does that mean for ratio calculation?

2 % error



3 % error



**Gaussian error propagation
aka First-order Taylor Expansion
vs
High-order Taylor Expansion
What`s the benefit?**

It is possible that the First-order Taylor Expansion does not sufficiently estimate the propagated error (underestimation).

As we have seen, the assumption of normality holds only for VERY small errors that are in a range we seldomly encounter in qPCR analysis.

'propagate': an R function for general error propagation

General definitions:

$$\sigma_Y^2 = \sum_i \left(\frac{\partial f}{\partial X_i} \right)^2 \sigma_{X_i}^2$$

Gaussian Error/
First order Taylor

$$\sigma_Y^2 = \sum_i \left(\frac{\partial f}{\partial X_i} \right)^2 \sigma_{X_i}^2 + \sum_{i \neq j} \sum_{j \neq i} \frac{\partial f}{\partial X_i} \frac{\partial f}{\partial X_j} \sigma_{X_i X_j}$$

Gaussian Error/
First order Taylor
with Covariance



$$\begin{bmatrix} \frac{\partial f}{\partial X_1} & \dots & \frac{\partial f}{\partial X_i} \end{bmatrix} \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1 X_2} & \dots & \sigma_{X_1 X_i} \\ \sigma_{X_2 X_1} & \sigma_{X_2}^2 & \dots & \sigma_{X_2 X_i} \\ \dots & \dots & \dots & \dots \\ \sigma_{X_i X_1} & \sigma_{X_i X_2} & \dots & \sigma_{X_i}^2 \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial X_1} \\ \dots \\ \frac{\partial f}{\partial X_i} \end{bmatrix} \Rightarrow \sigma_Y^2 = F_x C_x F_x^T$$

k-order Taylor with Covariance

$$\begin{aligned} \sigma_Y^2 = & \frac{1}{1} \left(\sum_i \left(\frac{\partial}{\partial X_i} \right) \sigma_{X_i}^2 + \sum_{i \neq j} \sum_{j \neq i} \frac{\partial}{\partial X_i} \frac{\partial}{\partial X_j} \sigma_{X_i X_j} \right) f + \frac{1}{2} \left(\sum_i \left(\frac{\partial}{\partial X_i} \right) \sigma_{X_i}^2 + \sum_{i \neq j} \sum_{j \neq i} \frac{\partial}{\partial X_i} \frac{\partial}{\partial X_j} \sigma_{X_i X_j} \right)^2 f \\ & + \frac{1}{6} \left(\sum_i \left(\frac{\partial}{\partial X_i} \right) \sigma_{X_i}^2 + \sum_{i \neq j} \sum_{j \neq i} \frac{\partial}{\partial X_i} \frac{\partial}{\partial X_j} \sigma_{X_i X_j} \right)^3 f + \dots + \frac{1}{k!} \left(\sum_i \left(\frac{\partial}{\partial X_i} \right) \sigma_{X_i}^2 + \sum_{i \neq j} \sum_{j \neq i} \frac{\partial}{\partial X_i} \frac{\partial}{\partial X_j} \sigma_{X_i X_j} \right)^k f \end{aligned}$$



$$\sigma_Y^2 = \sum_i^k \frac{1}{k!} \left(F_{x_i} C_x^k F_{x_i}^T \right)$$

```
DD <- function(expr, name, order = 1) {  
  if(order < 1) stop("'order' must be >= 1")  
  if(order == 1) D(expr, name)  
  else DD(D(expr, name), name, order - 1)  
}
```

```
CX <- diag(SD^2)
```

Higher derivatives: maybe for your worst enemy...

$$\frac{\text{eff.c}^{(\text{goi.C}-\text{goi.S})}}{\text{eff.s}^{(\text{ref.C}-\text{ref.S})}}$$

first-order

[[1]]

[[1]][[1]]

$$\text{eff.c}^{(\text{goi.C} - \text{goi.S}) - 1} * (\text{goi.C} - \text{goi.S}) / (\text{eff.r}^{(\text{ref.C} - \text{ref.S})})$$

[[1]][[2]]

$$-((\text{eff.c}^{(\text{goi.C} - \text{goi.S})}) * (\text{eff.r}^{(\text{ref.C} - \text{ref.S}) - 1} * (\text{ref.C} - \text{ref.S})) / (\text{eff.r}^{(\text{ref.C} - \text{ref.S})})^2)$$

[[1]][[3]]

$$-((\text{eff.c}^{(\text{goi.C} - \text{goi.S})}) * (\text{eff.r}^{(\text{ref.C} - \text{ref.S})} * \log(\text{eff.r})) / (\text{eff.r}^{(\text{ref.C} - \text{ref.S})})^2)$$

[[1]][[4]]

$$(\text{eff.c}^{(\text{goi.C} - \text{goi.S})}) * (\text{eff.r}^{(\text{ref.C} - \text{ref.S})} * \log(\text{eff.r})) / (\text{eff.r}^{(\text{ref.C} - \text{ref.S})})^2$$

[[1]][[5]]

$$\text{eff.c}^{(\text{goi.C} - \text{goi.S})} * \log(\text{eff.c}) / (\text{eff.r}^{(\text{ref.C} - \text{ref.S})})$$

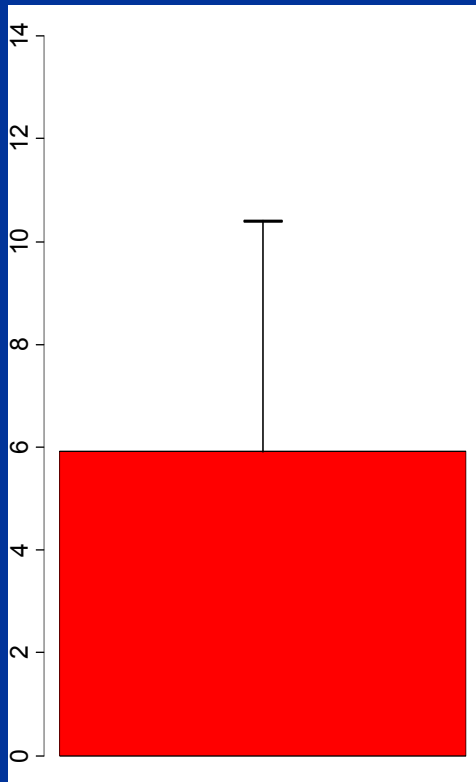
[[1]][[6]]

$$-(\text{eff.c}^{(\text{goi.C} - \text{goi.S})} * \log(\text{eff.c}) / (\text{eff.r}^{(\text{ref.C} - \text{ref.S})}))$$

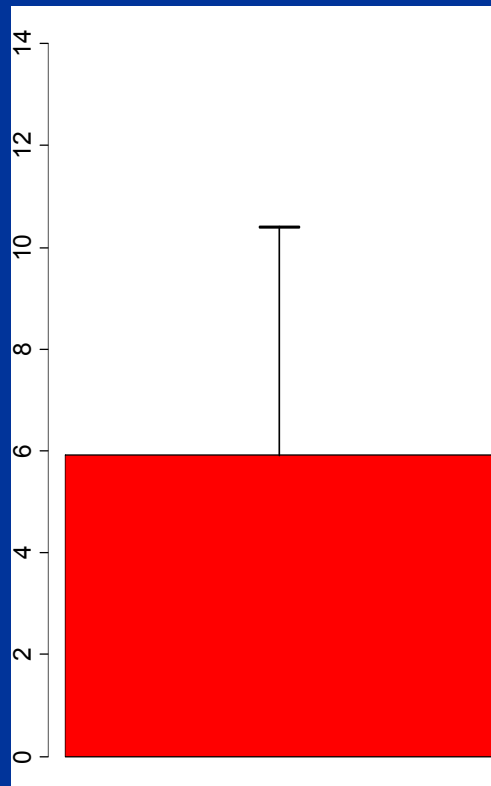
(only) third-order partial derivative of ref.S !

$$\begin{aligned} & (\text{eff.c}^{\text{goi.C} - \text{goi.S}}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * \log(\text{eff.r}) * \log(\text{eff.r})) / (\text{eff.r}^{\text{ref.C} - \text{ref.S}})^2 - \\ & (\text{eff.c}^{\text{goi.C} - \text{goi.S}}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * \log(\text{eff.r})) * (2 * \\ & (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}}))) / ((\text{eff.r}^{\text{ref.C} - \text{ref.S}})^2)^2 - \\ & (((\text{eff.c}^{\text{goi.C} - \text{goi.S}}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r})) * (2 * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \\ & \log(\text{eff.r}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r})) + \text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * \log(\text{eff.r}) * \\ & (\text{eff.r}^{\text{ref.C} - \text{ref.S}}))) + (\text{eff.c}^{\text{goi.C} - \text{goi.S}}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * \log(\text{eff.r})) \\ & * (2 * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}})))) / ((\text{eff.r}^{\text{ref.C} - \text{ref.S}})^2)^2 - \\ & (\text{eff.c}^{\text{goi.C} - \text{goi.S}}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r})) * (2 * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) \\ & * (\text{eff.r}^{\text{ref.C} - \text{ref.S}}))) * (2 * (2 * (\text{eff.r}^{\text{ref.C} - \text{ref.S}} * \log(\text{eff.r}) * (\text{eff.r}^{\text{ref.C} - \text{ref.S}})) * \\ & ((\text{eff.r}^{\text{ref.C} - \text{ref.S}})^2))) / (((\text{eff.r}^{\text{ref.C} - \text{ref.S}})^2)^2)^2 \end{aligned}$$

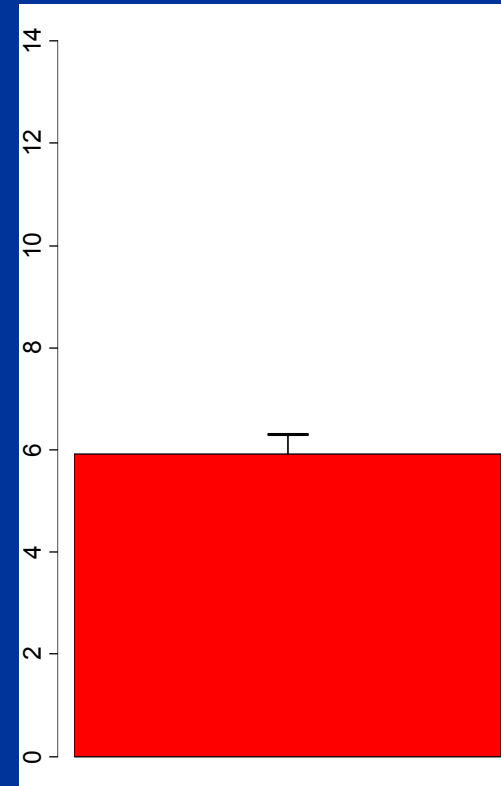
What's more important for error propagation: Reducing variance in efficiency or ct's?



2% error on Eff
2% error on ct's



0.2% error on Eff
2% error on ct's



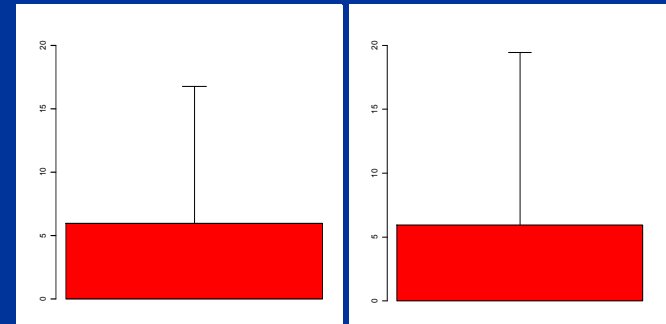
2% error on Eff
0.2% error on ct's

Finally...

How does it all compare?

eff.c	eff.s	ref.C	ref.S	goi.C	goi.S
2.01	1.97	26.74	26.77	27.57	24.54
		26.85	26.47	27.61	24.95
		26.83	27.03	27.82	24.57
		26.68	26.92	27.12	24.63
		27.39	26.97	27.76	24.66
		27.03	26.97	27.74	24.89
		26.78	26.07	26.91	24.71
		27.32	26.3	27.49	24.9
			26.14		24.26
			26.81		24.44

3% error

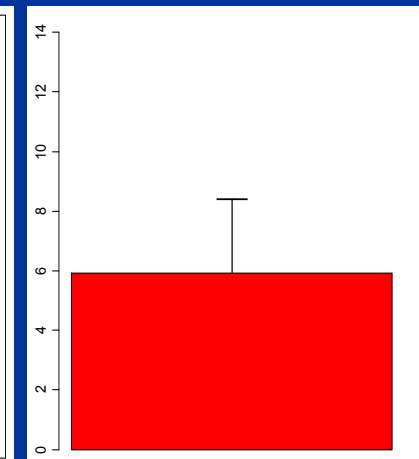
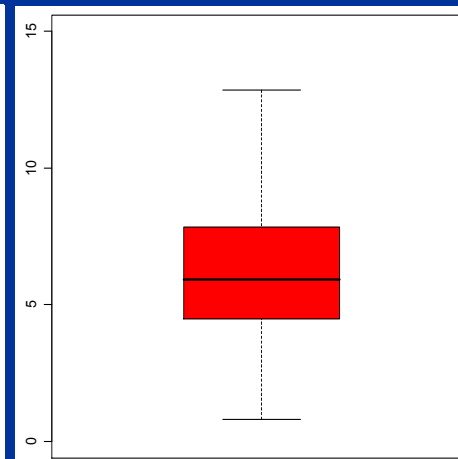
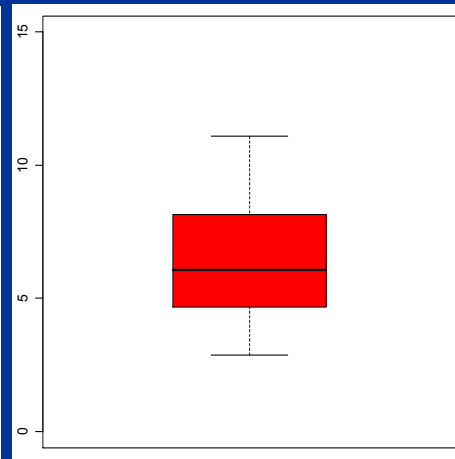
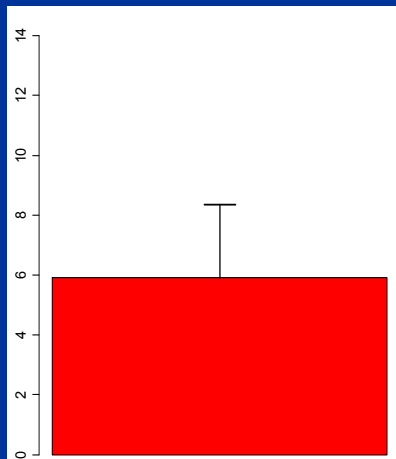


Gaussian

permutation
(REST)

Monte Carlo

fifth-order
Taylor



Please consider qpcR! (www.dr-spiess.de/qpcR.html)

Spiess et al., BMC Bioinformatics (2008)

Ritz & Spiess, Bioinformatics (2008)

R Help for package qpcR

qpcR(qpcR)

R Documentation

Modelling and analysis of real-time PCR data

Description

Model fitting, optimal model selection and calculation of various features that are essential in the analysis of quantitative real-time polymerase chain reaction (qPCR).

Details

Package: qpcR
Type: Package
LazyLoad: yes
LazyData: yes
Version: 1.1-7
License: GNU GPL (version 2 or any later version)
Depends: R (>= 2.0.0), drc
URL: <http://www.dr-spiess.de>; <http://www.r-project.org>; <http://www.bioassay.dk>
Packaged: Sep 22 2008, spiess
Built: R 2.7.1; ; 2008-09-22; windows

package: qpcR Modelling and analysis of real-time PCR data.

Functions:

- [confband](#) Confidence bands for a sigmoidal fit.
- [eff](#) The amplification efficiency curve of a fitted object.
- [efficiency](#) Calculation of qPCR efficiency and several other important qPCR parameters.
- [mchoice](#) Selection of the best model by nested F-tests, likelihood ratios or Akaike weights.
- [perplot](#) Plotting PCR data with fitted curves.
- [replot](#) Formatting of PCR data with or without replicates for analysis.
- [resVar](#) Residual variance of a fitted model.
- [Rsq](#) R-square value of a fitted model.
- [shwin](#) Calculation of PCR efficiency by the window-of-linearity method.
- [expfit](#) Calculation of PCR efficiency by fitting an exponential model.
- [perbatch](#) Batch calculation of qPCR efficiency and several other important qPCR parameters.
- [RMSE](#) Root-mean-squared-error of a fitted model.
- [calib](#) Calculation of qPCR efficiency by dilution curve analysis.

- * fit 4- and asymmetric 5-parameter sigmoidal models
- * calculate efficiency from sigmoidal, exponential and dilution curves
- * do model selection
- * batch analysis of many runs
- * average multiple reference curves
- * calculate ratios and errors thereof (MC, propagation)
- * many more...