Package 'freqpcr'

January 27, 2022

Type Package

Title Estimates Allele Frequency on qPCR DeltaDeltaCq from Bulk
Samples

Version 0.4.0

Maintainer Masaaki Sudo <masaaki@sudori.info>

Description Interval estimation of the population allele frequency from qPCR analy-
sis based on the restriction enzyme digestion (RED)-DeltaDeltaCq method (Osak-
abe et al. 2017, <doi:10.1016/j.pestbp.2017.04.003>), as well as general DeltaDeltaCq analy-
sis. Compatible with the Cq measurement of DNA extracted from multiple individu-
als at once, so called ``group-testing'', this model assumes that the quantity of DNA ex-
tracted from an individual organism follows a gamma distribution. Therefore, the point esti-
mate is robust regarding the uncertainty of the DNA yield.

URL https://github.com/sudoms/freqpcr

Depends R (>= 3.6), cubature

Imports methods

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, remotes

VignetteBuilder knitr

NeedsCompilation no

Author Masaaki Sudo [aut, cre] (<https://orcid.org/0000-0001-9834-9857>)

Repository CRAN

Date/Publication 2022-01-27 10:30:04 UTC

R topics documented:

.freqpcr_loglike   *Log-likelihood of obtaining Cq values under given parameter set.*

Description

The internal function is called from the optimizer, nlm(), running in freqpcr(). It defines the log-likelihood by obtaining the two ΔCq values (differences in the four Cq measurements) provided that the allele mixing ratio for each bulk sample is given together with other parameters. This function is vectorized over multiple bulk samples.

Usage

```
.freqpcr_loglike(
  X,
  N,
  DCW,
  DCD,
  zeroAmount,
  para.fixed = NULL,
  beta = TRUE,
  diploid = FALSE,
  dummyDCW = FALSE
)
```

Arguments

X	Numeric vector that stores the parameter values to be optimized via nlm(): P in logit scale and K, targetScale, sdMeasure, and EPCR in log scale.
N	Sample sizes as a numeric vector. N[i] signifies the number of individuals (both for haploidy and diploidy) contained in the *i*th bulk sample.
DCW, DCD	Numeric vectors having the same length as N. They store the measured values of the two ΔCq: DCW (= target0 -housek0) and DCD (= target1 -housek1). They can contain NA (simply ignored in the calculation).
zeroAmount	(In RED-$\Delta\Delta$Cq method) residue rate of restriction enzyme digestion, or (in general $\Delta\Delta$Cq analyses) small portion of the off-target allele on the target locus of the test sample, which will be amplified in the PCR. It needs to be always specified by the user as a number between 0 and 1, usually near 0.

para.fixed	Named numeric vector that stores the fixed parameters inherited from [freqpcr](), if specified. By default (NULL), all the parameters (P, K, targetScale, sdMeasure, and EPCR) are unknown. Unlike X, each element value is set in linear scale.
beta	Whether to use the beta distribution to approximate the sample allele ratio instead of specifying individual gamma distribution for each of the allelic DNA amounts? Default is TRUE, which accelerates the calculation.
diploid	Is the target organism diploidy? Default is FALSE, assuming haploidy. Current implementation of diploidy assumes i.i.d. between the amounts of R and S chromosomes owned by a heterozygote individual, which is unlikely in many animals but necessary for the calculation in a realistic time.
dummyDCW	Whether the ΔCq values of the control samples are dummy or not.

Value

Scalar of the log likelihood.

.freqpcr_loglike_cont   *Log-likelihood when sample allele ratio is continuous.*

Description

Called from [freqpcr]() instead of [.freqpcr_loglike]() when the model is 'continuous'. This function assumes that each sample does not consist of n individual organisms with certain genotypes, but the result of a direct DNA extraction from the sub-population having the allele ratio around p:(1-p). Each sample allele ratio is considered to follow Beta(apk,a(1-p)k), where a and k are the relative DNA content of the sample and the gamma shape parameter, respectively.

Usage

```
.freqpcr_loglike_cont(
  X,
  A,
  DCW,
  DCD,
  zeroAmount,
  para.fixed = NULL,
  beta = TRUE,
  dummyDCW = FALSE
)
```

Arguments

X	Numeric vector that stores the parameter values to be optimized via [nlm](): P in logit scale and K, targetScale, sdMeasure, and EPCR in log scale.
A	Relative DNA content between the samples. A continuous version of N in [.freqpcr_loglike](), as a numeric vector.

DCW, DCD	Numeric vectors. They store the measured values of the two ΔCq: DCW (= target0 -housek0) and DCD (= target1 -housek1).
zeroAmount	(In RED-$\Delta\Delta$Cq method) residue rate of restriction enzyme digestion, or (in general $\Delta\Delta$Cq analyses) small portion of the off-target allele on the target locus of the test sample, which will be amplified in the PCR. It needs to be always specified by the user as a number between 0 and 1, usually near 0.
para.fixed	Named numeric vector that stores the fixed parameters inherited from [freqpcr](), if specified. By default (NULL), all the parameters (P, K, targetScale, sdMeasure, and EPCR) are unknown. Unlike X, each element value is set in linear scale.
beta	Whether to use the beta distribution to approximate the sample allele ratio instead of specifying individual gamma distribution for each of the allelic DNA amounts? Default is TRUE, which accelerates the calculation.
dummyDCW	Whether the ΔCq values of the control samples are dummy or not.

Value

Scalar of the log likelihood.

CqFreq-class	*Output object of* [freqpcr]().

Description

Output object of [freqpcr]().

Slots

report A matrix of the simultaneous parameter estimation result. The rows represent the parameters P, K, targetScale (δ_T), sdMeasure (σ_c), and EPCR (η).

obj Returned value of the optimizer function [nlm]() as a list.

cal.time Calculation time of [nlm](), stored as a proc_time class object.

CqList-class	*S4 class storing the dummy Cq data for performance test.*

Description

A dummy Cq dataset suitable for a package test, typically obtained as the output of [make_dummy]().

Slots

N Sample sizes as a numeric vector. The ntrap and npertrap arguments of [make_dummy]() are inherited to the length of N and each N[i] element, the number of individuals (both for haploidy and diploidy) contained in the *i*th bulk sample, respectively.

m Segregation ratio. As for haploidy, m is a matrix with 2 rows and ntrap columns. m[1,i] and m[2,i] stores the number of R (mutant) or S (wild type) individuals while N[i] = sum(m[,i]) specifies the total in the bulk sample. It has 3 rows and ntrap columns as for diploidy. While m[1,i] stands for the number of RR hogozygote individuals, m[2,i] and m[3,i] stand for the numbers of RS heterozygotes and SS homozygotes, respectively.

xR,xS Numeric vector of the same length with N. xR[i] stores the amount of the template DNA for R allele contained in the *i*th bulk sample.

housek0,target0,housek1,target1 Numeric vectors of the same lengths with N. Store the generated Cq values.

DCW ΔCq values measured on the control samples (DNA extract without endonuclease digestion in the RED-ΔΔCq method, or pure R solution in a general ΔΔCq method), DCW, is defined as (target0 -housek0).

DCD ΔCq values measured on the test samples (samples after endonuclease digestion in the RED-ΔΔCq method, or samples with unknown allele mixing ratios in a general ΔΔCq method), DCD, is defined as (target1 -housek1).

deldel ΔΔCq value, defined as (DCD -DCW).

RFreqMeasure A classical index of the allele frequency calculated for each bulk sample, which is defined as (1.0+EPCR)^(-deldel). Note that the values of EPCR and other parameters, such as P or K, are not recorded in the object to avoid leakage of information.

ObsP As RFreqMeasure can exceed 1 by definition, ObsP is defined as min(RFreqMeasure,1).

rand.seed The seed of the random-number generator (RNG) which was fed to the current R session to generate dummy m, xR and xS data.

---

freqpcr   *Estimate population allele frequency from the set of Cq measurements.*

Description

The function estimates the population allele frequency using the dataset of Cq values measured over length(N) bulk samples, each of which has a sample size of N[i] as the number of individuals included. N[i] can be 1, meaning that every individual is analyzed separately. For the *i*th sample, the four Cq values were measured as housek0[i], target0[i], housek1[i], and target1[i]. The function can estimate up to five parameters simultaneously when the Cq sets are available for more than two (bulk) samples: P, K, targetScale, sdMeasure, and EPCR.

Since v0.3.2, user can also use an experimental 'continuous model' by specifying A instead of N. That is, each sample DNA is directly extracted from the environment and the sample allele ratio y follows y ~ Beta(apk,a(1-p)k) instead of y ~ Beta(mk,(n-m)k),m ~ Binomial(n,p), where p and k are the population allele frequency and the gamma shape parameter of the individual DNA yield, respectively. Each element of A, a is a scaling factor of relative DNA contents between the

samples. The continuous model is likely when each sample directly comes from the environment e.g., water sampling in an eDNA analysis or cell culture in a petri dish.

Since v0.4.0, [freqpcr](#)() also works without specifying housek0 and target0 i.e., it can estimate population allele frequency from ΔCq values instead of $\Delta\Delta$Cq. In this setting, the sizes of targetScale and sdMeasure should be fixed in addition to EPCR and zeroAmount.

Usage

```
freqpcr(
  N,
  A,
  housek0,
  target0,
  housek1,
  target1,
  P = NULL,
  K = NULL,
  targetScale = NULL,
  sdMeasure = NULL,
  EPCR = 0.99,
  XInit0 = c(P = NULL, K = NULL, targetScale = NULL, sdMeasure = NULL, EPCR = NULL),
  zeroAmount = NULL,
  beta = TRUE,
  diploid = FALSE,
  pvalue = 0.05,
  gradtol = 1e-04,
  steptol = 1e-09,
  iterlim = 100,
  maxtime = 600,
  print.level = 1,
  ...
)
```

Arguments

N	Sample sizes as a numeric vector. N[i] signifies the number of individuals (both for haploidy and diploidy) contained in the *i*th bulk sample. N must not contain a missing value (NA). If N is not applicable (= even not 1), feed A instead of N and then the estimation process runs with the 'continuous model'.
A	Use instead of N in the continuous model. This is a scale factor to control the relative DNA content between samples. A[i] can take any positive value, but must not be NA. Considering the case you have arranged each sample by e.g. water filrtation or extraction from a culture in a petri dish, it is convenient to define the unit size of A[i] == 1.0 to be same as the vessel volume (e.g. 2.0 for two petri dishs, 0.5 for half bottle of water, etc.). When neither N nor A is specified by the user, the function stops. If both N and A are specified, only N is evaluated.
housek0	A numeric vector. In RED-$\Delta\Delta$Cq method, housek0 is the Cq values of the test sample without the restriction enzyme digestion, which is amplified with

the primer set for a housekeeping gene. In general $\Delta\Delta$Cq analyses, housek0 is defined for the control sample (typically, 100% mutant) solution, which is also amplified with the primer set for the housekeeping gene.

Since v0.4.0, you can run freqpcr() without specifying housek0 and target0 (a ΔCq method). As this setting halves the effective data points, it is recommended to fix other parameters, especially targetScale.

The four Cq arguments, housek0, target0, housek1, and target1, all must have the same data length. They also must be the same length as N or A. If the Cq dataset has missing values, they must be filled with NA so that the length of the data vectors will not differ.

target0	In RED-$\Delta\Delta$Cq method, target0[i] signifies the measured Cq value of the *i*th bulk sample without the digestion, for which both alleles, wild-type (S: susceptible) and mutant (R: resistant to a pesticide), on the target locus are amplified. In general $\Delta\Delta$Cq analyses, target0 is the Cq values of the pure-S control sample, which is amplified with a R-allele-specific primer set.
housek1	The Cq values of the test sample measured on the housekeeping gene after the restriction enzyme digestion (in RED-$\Delta\Delta$Cq method), or the test sample amplified on the housekeeping gene (in general $\Delta\Delta$Cq analyses).
target1	For each test sample with unknown allele-ratio, target1[i] is defined as the Cq value for the target locus amplified after the restriction enzyme digestion (in RED-$\Delta\Delta$Cq method), or the target locus amplified with the R-allele-specific primer set (in general $\Delta\Delta$Cq analyses).
P	Scalar. Population allele frequency from which the test samples are derived. Default is NULL and to be estimated. If the parameter is known, it is given as a numeric between 0 and 1.
K	Scalar. The gamma shape parameter of the individual DNA yield. Default is NULL and to be estimated. If known, given as a positive numeric.
targetScale	(δ_T) Scalar. The relative template DNA amount of the target locus to the housekeeping locus. If known, given as a positive numeric.
sdMeasure	(σ_c) Scalar. The measurement error (standard deviation) on each Cq value following Normal(0, σ_c^2). If known, given as a positive numeric.
EPCR	(η) Scalar. Amplification efficiency per PCR cycle. If known, given as a positive numeric. When EPCR = 1, template DNA doubles every cycle (EPCR + 1 = 2).
XInit0	Optionally the initial value for the parameter optimization can be specified, but it is strongly recommended to keep the argument as is. Unlike XInit in knownqpcr(), the argument is not directly passed to the optimizer; used only when each parameter is set unknown (the parameter is absent or specified as NULL).
zeroAmount	(In RED-$\Delta\Delta$Cq method) residue rate of restriction enzyme digestion, or (in general $\Delta\Delta$Cq analyses) small portion of the off-target allele on the target locus of the test sample, which will be amplified in the PCR. It needs to be always specified by the user as a number between 0 and 1, usually near 0.
beta	Whether to use the beta distribution to approximate the sample allele ratio instead of specifying individual gamma distribution for each of the allelic DNA amounts? Default is TRUE, which accelerates the calculation.

diploid	Is the target organism diploidy? Default is FALSE, assuming haploidy. Current implementation of diploidy assumes i.i.d. between the amounts of R and S chromosomes owned by a heterozygote individual, which is unlikely in many animals but necessary for the calculation in a realistic time.
pvalue	The two-sided confidence interval is calculated at the last iteration at given significance level. Default is 0.05, which returns the 95% Wald's CI (2.5 to 97.5 percentile) based on the Hessian matrix.
gradtol, steptol, iterlim	
	Control parameters passed to [nlm](). gradtol and steptol are the positive scalars giving the tolerance to terminate the algorithm and the minimum allowable relative step length. iterlim specifies the maximum number of iterations to be performed before the program is terminated (and evaluated at the last iteration). Usually 30 iterations are enough.
maxtime	A positive scalar to set the maximum calculation time in seconds to abort the optimizer (and return error). The total calculation time largely depends on N[i], the number of individuals contained in each bulk sample.
print.level	print.level=1 (the default) shows the initial values of the parameters and likelihood as well as the output in the last iteration. print.level=2 shows the parameter values and gradients in every step. print.level=0 does not output any intermediate state to R console, simply returning the result summary.
...	Additional arguments passed to the function.

Value

Object of the S4 class [CqFreq]. The slot report is a matrix and each row contains the estimated parameter value with 100*(1-pvalue)% confidence interval. The following parameters are returned:

1. P, the population allele frequency from which the test samples are derived.
2. K, the gamma shape parameter of the individual DNA yield.
3. targetScale (δ_T), the relative template DNA amount of the target to the houskeeping loci.
4. EPCR (η), the amplification efficiency per PCR cycle.
5. sdMeasure or "Cq measurement error" (σ_c).

Choise of the parameters to be estimated

Estimation is conducted only for parameters where the values are not specified or specified explicitly as NULL. If one feeds a value e.g. K=1 or sdMeasure=0.24, it is then treated as fixed parameter. Notwithstanding, EPCR is estimated only when EPCR = NULL is specified explicitly.

You must verify the size of EPCR and zeroAmount beforehand because they are not estimable from the experiments with unknown allele ratios. Although targetScale and sdMeasure are estimable within [freqpcr](), it is better to feed the known values, especially when you have only a few bulk samples (length(N) <= 3). Fixing targetScale and sdMeasure is also strongly recommended when housek0 and target0 are absent (ΔCq method). The functions [knownqpcr]() or [knownqpcr_unpaired](), depending on your data format, provide the procedure to estimate the sizes of the experimental parameters using the DNA solutions of known allele mixing ratios.

For the unknown parameters, XInit0 optionally specifies initial values for the optimization using

[nlm](#)() though the use of internal default is strongly recommended. The specification as a fixed parameter has higher priority than the specification in XInit0. Every user-specified parameter values, fixed or unknown, must be given in linear scale (e.g. between 0 and 1 for the allele frequency); they are transformed internally to log or logit.

See Also

Other estimation procedures: [knownqpcr_unpaired](#)(), [knownqpcr](#)(), [sim_dummy](#)()

Examples

```
# Dummy Cq dataset with six bulk samples, each of which comprises of eight haploids.
EPCR <- 0.95; zeroAmount <- 0.0016; # True values for the two must be known.
P <- 0.75
dmy_cq <- make_dummy(   rand.seed=1, P=P, K=2, ntrap=6, npertrap=8,
                        scaleDNA=1e-07, targetScale=1.5, baseChange=0.3,
                        EPCR=EPCR, zeroAmount=zeroAmount,
                        sdMeasure=0.3, diploid=FALSE   )
print(dmy_cq)

# Estimation with freqpcr, where P, K, targetScale, and baseChange are marked unknown.
result <- freqpcr( N=dmy_cq@N, housek0=dmy_cq@housek0, target0=dmy_cq@target0,
                   housek1=dmy_cq@housek1, target1=dmy_cq@target1,
                   EPCR=EPCR, zeroAmount=zeroAmount, beta=TRUE, print.level=0 )
print(result)

# Estimation with freqpcr, assuming diploidy.
result <- freqpcr( N=dmy_cq@N, housek0=dmy_cq@housek0, target0=dmy_cq@target0,
                   housek1=dmy_cq@housek1, target1=dmy_cq@target1,
                   EPCR=EPCR, zeroAmount=zeroAmount, beta=TRUE, diploid=TRUE )

# Estimation where you have knowledge on the size of K.
result <- freqpcr( N=dmy_cq@N, housek0=dmy_cq@housek0, target0=dmy_cq@target0,
                   housek1=dmy_cq@housek1, target1=dmy_cq@target1,
                   K=2,
                   EPCR=EPCR, zeroAmount=zeroAmount, beta=TRUE, print.level=2 )
# (>= v0.3.2)
# Provided the model is continuous (specify A instead of N).
result <- freqpcr( A=dmy_cq@N, housek0=dmy_cq@housek0, target0=dmy_cq@target0,
                   housek1=dmy_cq@housek1, target1=dmy_cq@target1,
                   K=2, EPCR=EPCR, zeroAmount=zeroAmount, beta=TRUE, print.level=1 )
# (>= v0.4.0)
# If the dataset lacks control samples (housek0 and target0 are absent).
# Fixing the sizes of targetScale and sdMeasure is strongly recommended.
result <- freqpcr( N=dmy_cq@N, housek1=dmy_cq@housek1, target1=dmy_cq@target1,
                   K=2, EPCR=EPCR, zeroAmount=zeroAmount,
                   targetScale=1.5, sdMeasure=0.3, beta=TRUE, print.level=1 )
```

Description

The function to estimate the auxiliary experimental parameters using DNA solutions, provided the dataset contains samples with multiple allele mixing ratios and the exact mixing ratio are known for each sample. This function is used when all replicates in the dataset comprise the complete observations on the 2×2 combinations of the qPCR conditions in a RED-$\Delta\Delta$Cq analysis: (loci for target or housekeeping genes) and (the target gene is undigested or digested with endonuclease). The quartet of the four Cq data, housek0, target0 (these two are undigested samples amplified with housekeeping and target genes, respectively), housek1, and target1 (digested samples) should be prepared as four numeric vectors having the same length.

One more variable, trueY is needed to run the estimation process; it a numeric vector having the same length with the four Cq data. It holds the exact allele-mixing ratio for each quartet (also see the code example). Optionally, you can adjust the relative DNA concentration between the replicates with a parameter vector A.

Since version 0.3.2, the [knownqpcr](<link>)() function can also deal with general $\Delta\Delta$Cq analyses. In such cases, samples with any mixing ratios are generally marked as 'digested samples' i.e., either of housek1 or target1, depending on the loci to be amplified. The arguments of the corresponding undigested samples, housek0 and target0, must not be specified by the user. Then, the parameter baseChange (δ_B: the change rate of DNA contents before/after the endonuclease digestion) is not included in the estimation result.

Usage

```
knownqpcr(
  housek0,
  target0,
  housek1,
  target1,
  trueY,
  A = rep(1, length(trueY)),
 XInit = c(meanDNA = -10, targetScale = 0, baseChange = 0, sdMeasure = 1, zeroAmount =
    -5, EPCR = 0),
  method = "BFGS",
  pvalue = 0.05,
  trace = 0,
  report = 10,
  verbose = FALSE
)
```

Arguments

housek0, target0, housek1, target1

                    Measured Cq values. Numeric vectors having the same length as trueY. Values must not be duplicated (any single Cq measure must not be recycled); if

the dataset has missing Cq values, there are two ways. 1) Fill the missing values explicitly with NA and use [knownqpcr](), or 2) use another function [knownqpcr_unpaired](), which accepts a 'long' format dataset.

In RED-$\Delta\Delta$Cq method, housek0 and target0 corresponds to the intact test samples (not digested with endonuclease) amplified with the housekeeping- and target-loci, respectively. In general $\Delta\Delta$Cq analyses, housek0 and target0 are absent, and only test samples (housek1 and target1) are input by the user. At a first glance, the test samples seem to be unaffected by endonuclease when trueY[i] == 1, but they must also be input as housek1 or target1 because in fact their Cq values are affected by baseChange.

trueY	A numeric vector having the same length as the Cq data. trueY[i] signifies the exact allele frequency in the *i*th sample. The values must be between 0 and 1, and NA is not allowed. To improve the estimation accuracy, it is better to include the settings y == 0 (pure S solution) and y == 1 (pure R) in your dataset.
A	Optionally, you can specify relative DNA content between the samples, as a numeric vector having the same length as the Cq data. If present, A must not include missing values. It is the counterpart of the N argument in [freqpcr](), whereas an element of A is not restricted to integer. Because the concentration as a whole is also adjusted with the parameter meanDNA (see Value section), A is used exclusively to reflect the relative contents between the sample solutions. Otherwise, A should be left unspecified (the default is 1 for all replicates).
XInit	Optionally, the named vector specifies the initial parameter values to be optimized. Defined in the natural log scale; e.g. zeroAmount = -5 corresponds to the residue rate exp(-5) = 0.007. Keeping the default is highly recommended.
method	A string specifying the optimization algorithm used in [optim](). The default is BFGS, which is plausible in most situation.
pvalue	The two-sided confidence interval is calculated at the last iteration at given significance level. Default is 0.05, which returns the 95% Wald's CI (2.5 to 97.5 percentile) based on the Hessian matrix.
trace	Non-negative integer. If positive, [optim]() outputs trace information. The default is 0 (no information).
report	The frequency of reports if trace is positive. Defaults to every 10 iterations.
verbose	Send messages to stdout? Default is FALSE.

Value

A table containing the estimated values for the following parameters:

1. meanDNA is the template DNA concentration (of housekeeping gene, per unit volume of sample solution) compared to the threshold line of PCR.

2. targetScale (δ_T) is the relative template DNA amount of the target to the houskeeping loci.

3. baseChange (δ_B) is the change rate in the DNA amount after endonuclease digestion in RED-$\Delta\Delta$Cq method. In general $\Delta\Delta$Cq analyses (neither housek0 nor target0 is input), this parameter is not returned. In both cases, baseChange is not required to run [freqpcr]().

4. sdMeasure (σ_c) is the measurement error (standard deviation) at each Cq value.

5. zeroAmount (z) is the ratio of non-target allele amplified in qPCR (see the argument list of freqpcr()).

6. EPCR (η) is the amplification efficiency per PCR cycle.

See Also

Other estimation procedures: freqpcr(), knownqpcr_unpaired(), sim_dummy()

Examples

```
# A dummy Cq dataset: four mixing ratios with four replicates.
# K:2, scaleDNA:1e-11, targetScale:1.5, baseChange:0.3, zeroAmount:1e-3,
# sdMeasure:0.3, and EPCR:0.95. Assuming a RED-DeltaDeltaCq analyses.
trueY <- c(rep(0.1, 4), rep(0.25, 4), rep(0.5, 4), rep(1, 4))
housek0 <- c( 19.39, 19.78, 19.28, 19.58,  18.95, 19.91, 19.66, 19.96,
              20.05, 19.86, 19.55, 19.61,  19.86, 19.27, 19.59, 20.21 )
target0 <- c( 19.16, 19.08, 19.28, 19.03,  19.17, 19.67, 18.68, 19.52,
              18.92, 18.79, 18.8, 19.28,   19.57, 19.21, 19.05, 19.15 )
housek1 <- c( 21.61, 21.78, 21.25, 21.07,  22.04, 21.45, 20.72, 21.6,
              21.51, 21.27, 21.08, 21.7,   21.44, 21.46, 21.5, 21.8 )
target1 <- c( 24.3, 24.22, 24.13, 24.13,   22.74, 23.14, 23.02, 23.14,
              21.65, 22.62, 22.28, 21.65,  20.83, 20.82, 20.76, 21.3 )
d.cmp <- data.frame(A=rep(1, 16), trueY, housek0, target0, housek1, target1)
print(d.cmp)

# In RED-DeltaDeltaCq analyses, four observations stem from one sample solution.
# Each argument must be specified with its name (name=source).
knownqpcr(  housek0=d.cmp$housek0, target0=d.cmp$target0,
            housek1=d.cmp$housek1, target1=d.cmp$target1,
            trueY=d.cmp$trueY, A=d.cmp$A, verbose=FALSE  )

# In general DeltaDeltaCq analyses, the experimental design will not include
# dedicated control samples. The function then runs without 'housek0' and 'target0'.
knownqpcr(  housek1=d.cmp$housek1, target1=d.cmp$target1,
            trueY=d.cmp$trueY, A=d.cmp$A, verbose=TRUE  )
```

---

knownqpcr_unpaired   *Estimate auxiliary parameters when the sample pairs are incomplete.*

Description

A variant of knownqpcr() that accepts the Cq values concatenated into a vector (the argument Cq) accompanied with the experimental conditions (the arguments Digest and Gene). Their exact allele mixing ratios are known as trueY.

Usage

```
knownqpcr_unpaired(
  Digest,
  Gene,
  trueY,
  Cq,
  A = rep(1, length(Cq)),
 XInit = c(meanDNA = -10, targetScale = 0, baseChange = 0, sdMeasure = 1, zeroAmount =
    -5, EPCR = 0),
  method = "BFGS",
  pvalue = 0.05,
  trace = 0,
  report = 10,
  verbose = FALSE
)
```

Arguments

Digest	Numeric vector having the same length as Gene, trueY, and Cq. NA is not allowed. In the RED-$\Delta\Delta$Cq method, it specify the sample is intact (= 0) or digested with endonuclease (= 1). In other $\Delta\Delta$Cq-based analyses, there will be no control sample and all observations must be marked Digest = 1.
Gene	Numeric vector that specify each Cq measure (element of Cq) was taken with housekeeping (= 0) or target (= 1) locus. NA is not allowed.
trueY	A numeric vector. trueY[i] signifies the exact frequency of the mutant allele in the *i*th sample. The values must be between 0 and 1, and NA is not allowed. To improve the estimation accuracy, it is better to include the settings y == 0 (pure S solution) and y == 1 (pure R) in your dataset.
Cq	Measured Cq values. This argument is a numeric vector and can contain NAs. The vector length must be the same as Digest, Gene, and trueY (i.e., missing values must be filled with NA).
A	Optionally, you can specify relative DNA content between the samples, as a numeric vector having the same length as Cq. If present, A must not include missing values. It is the counterpart of the N argument in [freqpcr](), whereas an element of A is not restricted to integer. Because the concentration as a whole is also adjusted with the parameter meanDNA (see Value section), A is used exclusively to reflect the relative contents between the sample solutions. Otherwise, A should be left unspecified (the default is 1 for all replicates).
XInit	Optionally, the named vector specifies the initial parameter values to be optimized. Defined in the natural log scale; e.g. zeroAmount = -5 corresponds to the residue rate exp(-5) = 0.007. Keeping the default is highly recommended.
method	A string specifying the optimization algorithm used in [optim](). The default is BFGS, which is plausible in most situation.
pvalue	The two-sided confidence interval is calculated at the last iteration at given significance level. Default is 0.05, which returns the 95% Wald's CI (2.5 to 97.5 percentile) based on the Hessian matrix.

trace	Non-negative integer. If positive, [optim]() outputs trace information. The default is 0 (no information).
report	The frequency of reports if trace is positive. Defaults to every 10 iterations.
verbose	Send messages to stdout? Default is FALSE.

Value

A table containing the estimated parameter values. The format is same as [knownqpcr]().

See Also

Other estimation procedures: [freqpcr](), [knownqpcr](), [sim_dummy]()

Examples

```
# A dummy Cq dataset: four mixing ratios with four replicates.
# K:2, scaleDNA:1e-11, targetScale:1.5, baseChange:0.3, zeroAmount:1e-3,
# sdMeasure:0.3, and EPCR:0.95. Assuming a RED-DeltaDeltaCq analyses.
trueY <- c(rep(0.1, 4), rep(0.25, 4), rep(0.5, 4), rep(1, 4))
housek0 <- c( 19.39, 19.78, 19.28, 19.58,  18.95, 19.91, 19.66, 19.96,
              20.05, 19.86, 19.55, 19.61,  19.86, 19.27, 19.59, 20.21 )
target0 <- c( 19.16, 19.08, 19.28, 19.03,  19.17, 19.67, 18.68, 19.52,
              18.92, 18.79, 18.8, 19.28,   19.57, 19.21, 19.05, 19.15 )
housek1 <- c( 21.61, 21.78, 21.25, 21.07,  22.04, 21.45, 20.72, 21.6,
              21.51, 21.27, 21.08, 21.7,   21.44, 21.46, 21.5, 21.8 )
target1 <- c( 24.3, 24.22, 24.13, 24.13,   22.74, 23.14, 23.02, 23.14,
              21.65, 22.62, 22.28, 21.65,  20.83, 20.82, 20.76, 21.3 )

# Incomplete observation dataset, prepared as the "long" format.
# If the undegested (Digest == 0) samples were only analyzed when trueY == 1.
d.long.all <- data.frame(
    trueY=rep(trueY, 4), Digest=c(rep(0, 16 + 16), rep(1, 16 + 16)),
    Gene=c(rep(0, 16), rep(1, 16), rep(0, 16), rep(1, 16)),
    A=rep(1, 16*4), Cq=c(housek0, target0, housek1, target1)  )
d.long <- d.long.all[d.long.all$Digest == 1 | d.long.all$trueY == 1, ]
print(d.long)

knownqpcr_unpaired( Digest=d.long$Digest, Gene=d.long$Gene,
                    trueY=d.long$trueY, Cq=d.long$Cq, A=d.long$A )

# In general DeltaDeltaCq analyses, the experimental design will not include
# dedicated control samples (Digest == 0).
d.long <- d.long.all[d.long.all$Digest == 1, ]
knownqpcr_unpaired( Digest=d.long$Digest, Gene=d.long$Gene,
                    trueY=d.long$trueY, Cq=d.long$Cq, A=d.long$A )
```

---

make_dummy   *Generate dummy DNA dataset ready for allele-frequency estimation.*

Description

The function generates a dummy dataset of typical RED-$\Delta\Delta$Cq analysis. You can directly feed the output of this function to the first argument of sim_dummy().

Usage

```
make_dummy(
  rand.seed,
  P,
  K,
  ntrap,
  npertrap,
  scaleDNA = (1/K) * 1e-06,
  targetScale,
  baseChange,
  EPCR,
  zeroAmount,
  sdMeasure,
  diploid = FALSE
)
```

Arguments

rand.seed	Seed for the R built-in random-number-generator.
P	A numeric between 0 and 1 giving the population allele frequency from which the test samples are generated.
K	A positive numeric of the gamma shape parameter of the individual DNA yield.
ntrap, npertrap	
	Scalar specifying the number of bulk samples (ntrap) and the numbers of individuals contained in each bulk sample (npertrap). Currently limited to the cases that all bulk samples have the same sample size: e.g. $(4 + 4 + 4)$ when ntrap = 3 and npertrap = 4 hold.
scaleDNA	Small positive scalar that specifies the scale parameter of the gamma distribution appriximating the DNA yield from (per-haploid) individual. The yield of 2*scaleDNA is expected from a diploid. The quantity is determined as the relative amount, in linear scale, to the termination threshold of the real-time PCR.
targetScale	(δ_T) The relative template DNA amount of the target gene to the houskeeping gene, given as a positive numeric.
baseChange	(δ_B) The change rate in the template DNA quantities after the restriction enzyme digestion (in the RED-$\Delta\Delta$Cq method), given as a positive numeric. This parameter is not used in freqpcr().

EPCR	(η) Amplification efficiency per PCR cycle, given as a positive numeric. When EPCR = 1, template DNA doubles every cycle (EPCR + 1 = 2).
zeroAmount	A numeric between 0 and 1, usually near 0, giving the residue rate of restriction enzyme digestion in RED-$\Delta\Delta$Cq method.
sdMeasure	(σ_c) Scalar. The measurement error (standard deviation) on each Cq value following Normal(0, σ_c^2). If known, given as a positive numeric.
diploid	Is the target organism diploidy? Default is FALSE, assuming haploidy. Current implementation of diploidy assumes i.i.d. between the amounts of R and S chromosomes owned by a heterozygote individual, which is unlikely in many animals but necessary for the calculation in a realistic time.

Value

Object of the S4 class [CqList](#), storing the dummy experiment data of Cq-based qPCR analysis. Note that a [CqList](#) object in no way contains original information on P, K, targetScale, sdMeasure, and EPCR.

Examples

```
P <- 0.25
# Just a test: segregation ratios for six bulk samples, 1000 individuals for each.
rmultinom(n=6, size=1000, prob=c(P, 1-P)) # haploidy
rmultinom(6, size=1000, prob=c(P^2, 2*P*(1-P), (1-P)^2)) # diploidy

# Dummy Cq dataset with six bulk samples, each of which comprises eight haploids.
dmy_cq <- make_dummy( rand.seed=1, P=0.15, K=2, ntrap=6, npertrap=8,
                      scaleDNA=1e-07, targetScale=1.5, baseChange=0.3, EPCR=0.95,
                      zeroAmount=1e-3, sdMeasure=0.3, diploid=FALSE )
print(dmy_cq)
```

---

sim_dummy   *Simulate freqpcr estimation based on user-generated dummy data.*

Description

Wrapper of [freqpcr](#)() suitable for the performance test using a randomly-generated data object.

Usage

```
sim_dummy(
  CqList,
  EPCR,
  zeroAmount,
  P = NULL,
  K = NULL,
  targetScale = NULL,
  sdMeasure = NULL,
```

```
    beta,
    diploid,
    maxtime,
    print.level,
    aux = NULL,
    verbose = TRUE,
    ...
)
```

Arguments

CqList
: Object belonging to the [CqList](#) class, typically the output from [make_dummy](#)(). Having the slots N, housek0, target0, housek1, and target1, all of which are numeric vectors of the same length.

EPCR
: (η) Amplification efficiency per PCR cycle, given as a positive numeric. When EPCR = 1, template DNA doubles every cycle (EPCR + 1 = 2).

zeroAmount
: A numeric between 0 and 1, usually near 0, giving the residue rate of restriction enzyme digestion in RED-$\Delta\Delta$Cq method.

P, K, targetScale, sdMeasure
: If NULL (default), the parameter is considered unknown and estimated via [freqpcr](#)(). If a value is specified, it is passed to [freqpcr](#)() as a fixed parameter. On the contrary, EPCR and zeroAmount are always treated as fixed parameters, for which values must be supplied.

beta, diploid, maxtime, print.level
: Configuration parameters which are passed directly to [freqpcr](#)().

aux
: Additional information to be displayed on the console. The default is NULL. If some value is input by the user, it is echoed to stdout together with the contents of the argument CqList. This option is convenient when you want to record the original dummy dataset and the corresponding result sequentially e.g. using capture.output().

verbose
: Prints more information e.g. system time. Default is TRUE.

...
: Additional arguments passed to [freqpcr](#)().

Value

Object of the S4 class [CqFreq](#), which is same as [freqpcr](#)().

See Also

Other estimation procedures: [freqpcr](#)(), [knownqpcr_unpaired](#)(), [knownqpcr](#)()

Examples

```
# Prepare the parameter values.
K <- 2 # You already know the size of K in this case.
EPCR <- 0.97 # The sizes of EPCR and zeroAmount must always be supplied.
zeroAmount <- 1.6e-03
is.diploid <- FALSE
```

```
# First, make a dummy Cq dataset with six bulk DNA samples,
# each of which comprises of eight haploid individuals.
dmy_cq <- make_dummy( rand.seed=1, P=0.75, K=K, ntrap=6, npertrap=8, scaleDNA=1e-07,
                      targetScale=1.5, baseChange=0.3, EPCR=EPCR,
                      zeroAmount=zeroAmount, sdMeasure=0.3, diploid=is.diploid )

# Estimate the population allele frequency on the dummy dataset, presupposing K = 2.
sim_dummy(  CqList=dmy_cq, EPCR=EPCR, zeroAmount=zeroAmount,
            K=K,
            beta=TRUE, diploid=is.diploid, maxtime=60, print.level=2, aux="test" )

# If the maximum calculation time was too short to converge, nlm() returns error.
# Then sim_dummy() returns a matrix filled with zeros.
sim_dummy(  CqList=dmy_cq, EPCR=EPCR, zeroAmount=zeroAmount,
            beta=TRUE, diploid=is.diploid, maxtime=0.01, print.level=2 )
```

Index