Package 'testassay'

October 14, 2022

Type Package

Title A Hypothesis Testing Framework for Validating an Assay for Precision

Version 0.1.1

Author Michael C Sachs and Michael P Fay

Maintainer Michael C Sachs <sachsmc@gmail.com>

Description A common way of validating a biological assay for is through a procedure, where m levels of an analyte are measured with n replicates at each level, and if all m estimates of the coefficient of variation (CV) are less than some prespecified level, then the assay is declared validated for precision within the range of the m analyte levels. Two limitations of this procedure are: there is no clear statistical statement of precision upon passing, and it is unclear how to modify the procedure for assays with constant standard deviation. We provide tools to convert such a procedure into a set of m hypothesis tests. This reframing motivates the m:n:q procedure, which upon completion delivers a 100q% upper confidence limit on the CV. Additionally, for a post-validation assay output of y, the method gives an ``effective standard deviation interval'' of log(y) plus or minus r, which is a 68% confidence interval on log(mu), where mu is the expected value of the assay output for that sample. Further, the m:n:q procedure can be straightforwardly applied to constant standard deviation assays. We illustrate these tools by applying them to a growth inhibition assay. This is an implementation of the methods described in Fay, Sachs, and Miura (2018) <doi:10.1002/sim.7528>.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-03 14:00:02 UTC

R topics documented:

`gia` *Growth Inhibition Assay*

Description

Data from a growth inhibition assay experiment. Samples were run repeatedly on different assays, for two different strains of parasites (3d7 and FVO). `elisa` is a measure of the amount of antibody and is measured once for each sample. `sample` is a unique name for each sample and is defined as `paste(gia$parasite,gia$elisa,sep=".")`. `gia` is the value of interest, and the `meanAAgia` is the sample level mean, which is the best estimate of the "true" gia level for that sample. `varAAgia` is the sample level variance.

Usage

```
gia
```

Format

A data frame with variables: `parasite`, `assay`, `elisa`, `gia`, `sample`, `meanAAgia`, and `varAAgia`

`lognormConstCVCI` *log-normal constant CV model*

Description

This function gets confidence intervals on mu=E(Y) assuming Y is lognormal and the coefficient of variation is known.

Usage

```
lognormConstCVCI(y, theta, conf.level = 0.6827)
```

Arguments

`y`	Observed value
`theta`	coefficient of variation (assumed known)
`conf.level`	Confidence level

Details

Let Y be lognormal, so that log(Y) is normal with mean xi and variance eta. Then E(Y) =mu = exp(xi+eta/2) and Var(Y)=sigma^2 = mu^2 (exp(eta)-1), so that the coefficient of variation is sigma/mu = sqrt(exp(eta)-1). We want to get log-centered confidence intervals on mu, meaning intervals such that log(y) +/- r(theta), where r(theta) is a constant function of theta.

Value

A list with the following components

- obs y
- lower lower confidence limit on mu=E(Y)
- upper upper confidence limit on mu=E(Y)

Examples

```
# defaults to 68.27 percent confidence level, same level as Normal plus or minus 1 std dev.
lognormConstCVCI(3.4,.6)
# compare to normal constant CV model result
normConstCVCI(3.4,.6)
```

`normConstCVCI` *Log-centered confidence intervals from a Normal constant coeffficient of variation model*

Description

Assume Y is normal with mean mu>0 and coefficient of variation theta, then Y/mu ~ N(1, theta^2). Get log-centered confidence intervals (when possible), meaning intervals such that log(y) +/- r(theta), where r(theta) is a constant function of theta.

Usage

```
normConstCVCI(y, theta, conf.level = 0.6827, eps = .Machine$double.eps^0.25)
```

Arguments

`y`	vector of observed values, should be positive
`theta`	coefficient of variation (assumed known)
`conf.level`	Confidence level
`eps`	a small number used in the algorithm (look at code before changing)

Value

A list with the following components

- obs y
- lower lower confidence limit on mu=E(Y)
- upper upper confidence limit on mu=E(Y)

Examples

```
# defaults to 68.27 percent confidence level, same level as Normal plus or minus 1 std dev.
normConstCVCI(3.4,.6)
# symmetric on log scale
log(normConstCVCI(3.4,.6))
```

`predict.assaytest` *Construct effective standard deviation intervals for observed assay values*

Description

Computes effective standard deviation intervals for observed assay results. These intervals have at least 68.27 percent coverage.

Usage

```
## S3 method for class 'assaytest'
predict(object, newdata, ...)
```

Arguments

`object`	An object of class "assaytest"
`newdata`	A vector of observed values. If missing, uses object$x.
`...`	additional arguments

Details

Takes the `Umax` element from the `assaytest` object and treats it as the known precision parameter. For the constant SD model, the effective standard deviation interval for observed data value y is (y-Umax, y+Umax). For the constant CV models the effective SD interval uses either `normConstCVCI` (for the "normal" model) or `lognormConstCVCI` (for the "lognormal" model).

Although `Umax` is an upper bound (not an estimate) of the precision parameter, simulations have shown that treating `Umax` as the true precision parameter gives effective SD intervals with coverage of at least 68.27 percent (see Fay, Sachs, and Miura, 2016).

Value

A data frame with the observed values, lower, and upper confidence limits

References

Fay, MP, Sachs, MC, and Miura, K (2016). A Hypothesis Testing Framework for Validating and Assay for Precision (unpublished manuscript).

`print.assaytest` *Print the results of an assaytest*

Description

Print the results of an assaytest

Usage

```
## S3 method for class 'assaytest'
print(x, ...)
```

Arguments

`x`	An object of class "assaytest", as created by the function testassay
`...`	Additional arguments, currently unused

`testassay` *Hypothesis testing procedure for assay validation for precision*

Description

Does an m:n:q procedure for assay validation for precision. Returns an object of class 'assaytest'. There is are `predict` and `print` methods for that class.

Usage

```
testassay(x, m, n, q = 0.9, model = "normal", constant = "SD", data = NULL)
```

Arguments

x	The vector of assay values
m	The vector of values indicating sample membership
n	The vector of values indicating replicate membership
q	The confidence level, typically 0.8 or 0.9
model	String specifying the distribution for the assay values. Valid values are "normal" or "lognormal"
constant	String specifying whether the standard deviation is assumed to be constant over the levels ("SD") or the coefficient of variation is assumed constant over the levels ("CV"). The values "sd", "var", or "variance" may be used for "SD", and "cv" may be used for "CV".
data	Data frame or environment in which to look for x

Details

The m:n:q procedure uses m different samples that have different levels of the true value with n replicates for each sample. The output is a 100q percent upper limit of the bound on the precision parameter when the true values within the range of values for the m samples all follow either a a constant coefficient of variation model or a constant standard deviation model (same as a constant variance model) (see `constant` argument).

For example, if the 4:4:90 percent procedure using a normal model with a constant variance model returns a bound on the standard deviation (the Umax element of the assaytest class) of 7.9 then under the assumptions we have 90 percent confidence that the true SD is less than 7.9.

The `predict` method gives effective standard deviation intervals (i.e., 68.27 pct CIs) for the expected response from subsequent observed values from the assay.

Value

An object of class "assaytest", which is a list of components including a data frame of the relevant statistics calculated on x. Print, summary, predict, and plot methods are available. The list has the following components

- sumtabTable summarizing the experiment, includes mean values, SD or CV estimates, and upper confidence limits on those.
- UmaxThe maximum of the upper limits on the SD or CV, used in the effective SD interval calculation
- nThe number of samples per level
- mThe number of levels
- qThe confidence level
- modelThe assumed model
- constantThe parameter assumed to be constant (either 'SD' or 'CV').
- alphaThe alpha level, calculated as (1 - q)^(1 / m)
- xThe data vector supplied by the user

References

Fay, MP, Sachs, MC, and Miura, K (2016). A Hypothesis Testing Framework for Validating and Assay for Precision (unpublished manuscript).

Examples

```
# reproduce Table 3 of Fay, Sachs and Miura
I<- gia$parasite=="3D7" & gia$meanAAgia<80
treD7.test<-testassay(x=gia, m=sample, n=assay, q=.9,
  data=subset(gia, parasite=="3D7" & meanAAgia<80))
treD7.test
# get estimated effective standard deviation intervals (68.27 percent CIs)
# for observed values 21.4 and 65.9
# using results from testassay
predict(treD7.test,c(21.4,65.9))
```

Index