

# Package: bayescount (via r-universe)

September 7, 2024

**Version** 1.0.0-9

**Date** 2017-03-20

**Title** Statistical Analyses and Power Calculations for Count Data and Faecal Egg Count Reduction Tests (FECRT)

**Depends** R (>= 3.3.0)

**Imports** parallel, stats, utils, runjags, Rcpp (>= 1.0.0)

**Suggests** rjags, runjags, knitr, shiny, shinythemes, dplyr, tibble, tidyr, ggplot2

**VignetteBuilder** knitr

**LinkingTo** Rcpp

**Description** Power calculations and hypothesis testing for the difference in mean of two negative binomial distributions. A set of functions to allow analysis of count data (such as faecal egg count data) using Bayesian MCMC methods. Returns information on the possible values for mean count, coefficient of variation and zero inflation (true prevalence) present in the data. A complete faecal egg count reduction test (FECRT) model is implemented, which returns inference on the true efficacy of the drug from the pre- and post-treatment data provided, using non-parametric bootstrapping as well as using Bayesian MCMC. Functions to perform power analyses for faecal egg counts (including FECRT) are also provided. A working installation of JAGS (<<http://mcmc-jags.sourceforge.net>>) is required for MCMC-based methods

**License** GPL-2

**URL** <http://bayescount.sourceforge.net>

**RoxygenNote** 6.1.1

**Repository** <https://mdenwood.r-universe.dev>

**RemoteUrl** <https://github.com/mdenwood/bayescount>

**RemoteRef** HEAD

**RemoteSha** a9906dcb6d8fcf1563df5591d1bf325c742e7935

## Contents

bayescount . . . . .	2
count.analysis . . . . .	3
count.model . . . . .	5
count.power . . . . .	7
count.precision . . . . .	9
count_analysis . . . . .	11
count_model . . . . .	11
count_power . . . . .	12
count_precision . . . . .	12
fecrt.power . . . . .	13
fecrt.precision . . . . .	15
likelihood . . . . .	18
lnormal.params . . . . .	20
lnormal_params . . . . .	21
maximise.likelihood . . . . .	22
normal.params . . . . .	23
reduction_analysis . . . . .	24
reduction_model . . . . .	27
reduction_power . . . . .	30
reduction_precision . . . . .	31
<b>Index</b>	<b>32</b>

---

bayescount	<i>Analysis and power calculations for faecal egg count (FEC) and faecal egg count reduction test (FECRT) data using computationally intensive statistical methods</i>
------------	--

---

## Description

The bayescount package analyses the over-dispersed count datasets (such as those commonly encountered in parasitology) using computationally intensive statistical techniques such as non-parametric bootstrapping and Bayesian MCMC. The FEC data analysis returns information on the mean count, coefficient of variation and zero inflation (true prevalence) present in the data. The faecal egg count reduction test (FECRT) analysis calculates efficacy as well as post-treatment changes in count variation, and allows the analysis of repeat observations from the same animals. Power and sample size calculations for FECRT and FEC studies (given any expected parameter values) are also provided. Requires Just Another Gibbs Sampler (JAGS) for MCMC analysis functions (see <http://mcmc-jags.sourceforge.net>)

### *FEC Analysis*

You can do that

### *FECRT analysis*

That too

### *Power*

And that

## References

M. J. Denwood, S. W. J. Reid, S. Love, M. K. Nielsen, L. Matthews, I. J. McKendrick, and G. T. Innocent. Comparison of three alternative methods for analysis of equine Faecal Egg Count Reduction Test data. *Prev. Vet. Med.* (2009), doi:10.1016/j.prevetmed.2009.11.009

Denwood, M. J. (2010). A quantitative approach to improving the analysis of faecal worm egg count data. University of Glasgow. Retrieved from [http://www.gla.ac.uk/media/media\\_149338\\_en.pdf](http://www.gla.ac.uk/media/media_149338_en.pdf)

## See Also

[fecrt](#), [bayescount](#), [bayescount.single](#), [fec.power](#), [fecrt.power](#), [runjags](#)

---

count.analysis

*Analyse Count data using MCMC*

---

## Description

Apply a Bayesian [zero-inflated] gamma / Weibull / lognormal / independant / simple Poisson model to count data to return possible values for mean count, coefficient of variation, and zero-inflation, as either summary statistics or mcmc objects. Convergence is assessed for each dataset by calculating the Gelman-Rubin statistic for each parameter, see [autorun.jags](#). Optionally, the log likelihood for the model fit is also calculated. The time taken to complete each analysis (not including calculation of the likelihood) is also recorded. The lower level functions in the runjags package are used for calling JAGS.

Note: The GUI interface for R in Windows may not continually refresh the output window, making it difficult to track the progress of the simulation (if silent.jags is FALSE). To avoid this, you can run the function from the terminal version of R (located in the Program Files/R/bin/ folder).

## Usage

```
count.analysis(data = stop("Data must be specified"), model="ZILP",
  alt.prior = FALSE, adjust.zi.mean = FALSE, raw.output = FALSE,
  likelihood=FALSE, ...)
```

## Arguments

**data** an existing R object containing the data. Data can either be specified as a numeric vector of single counts for each sample, or as a matrix of repeated counts (columns) for each sample (rows). Repeated counts are modelled as part of the same Poisson process. Data can also be specified as a (ragged) list of repeated counts, with each element of the list representing a seperate sample. Finally, data can be specified as a list containing the elements 'totals' representing the sum of the repeated McMasters counts, and 'repeats' representing the number of McMasters counts performed per sample. Missing data (or unused elements of non-ragged arrays) may be represented using NA, which will be removed from the data before analysis. The likelihood calculation is not available for ragged arrays and missing data, and will print a warning. No default.

model	the model to use. Choices are "GP" (gamma Poisson = negative binomial), "ZIGP" (zero-inflated gamma Poisson = zero-inflated negative binomial), "LP" (lognormal Poisson), "ZILP" (zero-inflated lognormal Poisson), "WP" (Weibull Poisson), "ZIWP" (zero-inflated Weibull Poisson), "SP" (simple Poisson), "ZISP" (zero-inflated simple Poisson) or "IP" (independant Poisson). Case insensitive. The simple Poisson model forces each count to have the same mean, whereas the independant Poisson process allows each count to have an unrelated mean (therefore a zero-inflated version is not possible). Default "ZILP".
alt.prior	should the model run the [ZI] [WPIGPILP] models using the standard or the alternative prior distribution for variance? (logical) Can also be a character value of a user-specified prior distribution. Default FALSE. Where information concerning overdispersion in the data is sparse, the choice of prior distribution will have an affect on the posterior distribution for ALL parameters. It is recommended to run a simulation using both types of prior when working with small datasets, to make sure results are consistent.
adjust.zi.mean	should the mean count parameter of the zero-inflated models be adjusted to reflect the mean of the whole population? (logical) If FALSE the mean count of the zero-inflated models reflects the mean of the gamma or Poisson distribution only, if TRUE the mean includes extra zeros. Used for comparing results between zero-inflated and non zero-inflated models. Default FALSE.
raw.output	the function can return either a summary of the results (as with bayescount), or an MCMC object representing the estimates at each iteration for both chains (including the likelihood estimates where appropriate). If TRUE, the latter is output. (logical) Default FALSE.
likelihood	should the (log) likelihood for the fit of the model to the dataset be calculated? (logical) The likelihood for the [ZI] WP, LP and GP models are calculated using a likelihood function integrated over all possible values for lambda, which can take some time. The likelihood is calculated using a thinned chain of 1000 values to reduce the time taken. Default FALSE.
...	additional arguments to be passed directly to <a href="#">autorun.jags</a> .

### Value

Either a vector similar to that obtained from [bayescount](#) containing an indication of the error/crash/convergence status, the number of sampled updates used, and a lower/upper 95% highest posterior density interval (see [HPDinterval](#)), and median estimate for each relevant parameter (optionally including the likelihood), or an MCMC object representing the estimates at each iteration for both chains (optionally including the likelihood).

### See Also

[count.model](#)

### Examples

```
# use a zero-inflated lognormal Poisson model to analyse some count
# data, and suppressing JAGS output:
```

```
## Not run:
bayescount.single(data=c(0,5,3,7,0,4,3,8,0), model="ZILP",
silent.jags=TRUE)

## End(Not run)
```

---

count.model

---

*Analyse Count Data Using Jags*


---

## Description

Apply a Bayesian (zero-inflated) (gamma / Weibull / lognormal / independant / simple) Poisson model to count data to return possible values for mean count, variance, shape parameter, scale parameter (overdispersion or 'k') and zero-inflation where appropriate to the model selected. This function generates the model specifications and starting values, and is used by the higher level functions bayescount and bayescount.single, but can also be called directly. Requires Just Another Gibbs Sampler (JAGS). \*THIS SOFTWARE IS INTENDED FOR EDUCATIONAL PURPOSES ONLY AND SHOULD NOT BE RELIED UPON FOR REAL WORLD APPLICATIONS\* The GUI interface for R in Windows may not continually refresh the output window, making it difficult to track the progress of the simulation (if silent.jags is FALSE). To avoid this, you can run the function from the terminal version of R (located in the Program Files/R/bin/ folder).

## Usage

```
count.model(data=stop("No data supplied"),
  model=stop("No model specified"), call.jags = TRUE,
  alt.prior=FALSE, monitor.lambda=FALSE,
  monitor.deviance=FALSE, ...)
```

## Arguments

data	an existing R vector containing the data (integer vector). No default.
model	model to use. Choices are "GP" (gamma Poisson = negative binomial), "ZIGP" (zero-inflated gamma Poisson = zero-inflated negative binomial), "LP" (lognormal Poisson), "ZILP" (zero-inflated lognormal Poisson), "WP" (Weibull Poisson), "ZIWP" (zero-inflated Weibull Poisson), "SP" (simple Poisson), "ZISP" (zero-inflated simple Poisson) or "IP" (independant Poisson). The simple Poisson model forces each count to have the same mean, whereas the independant Poisson process allows each count to have an unrelated mean (therefore a zero-inflated version is not possible). Default "ZILP".
call.jags	should the function run the model using run.jags? If not, the model specification, initial values and data in the correct format are returned so that the model can be run using run.jags with no modifications. If TRUE, the model is run and the results are returned.

<code>alt.prior</code>	should the model run the [ZI] [WPIGPILP] models using the standard or the alternative prior distribution for variance? (logical) Can also be a character value of a user-specified prior distribution. Default FALSE. Where information concerning overdispersion in the data is sparse, the choice of prior distribution will have an affect on the posterior distribution for ALL parameters. It is recommended to run a simulation using both types of prior when working with small datasets, to make sure results are consistent.
<code>monitor.lambda</code>	should the model or model specification monitor the mean of the Poisson process for each datapoint? This is required to calculate the likelihood for the Independent Poisson model only, but may be useful for other purposes. Default FALSE.
<code>monitor.deviance</code>	option to monitor the deviance of the model (using the built-in 'deviance' monitor). Default FALSE.
<code>...</code>	additional arguments to be passed directly to <code>autorun.jags</code> .

### Value

Either a list of character strings representing [[1]] the model specification, [[2]] the data string, [[3]] the initial value string(s) and [[4]] the monitors required (if `call.jags` is FALSE) or an MCMC object returned from `run.jags` (if `call.jags` is TRUE).

### See Also

[count.analysis](#)

### Examples

```
# Return the model specification and starting values for a
# lognormal Poisson, then run the model using run.jags:

## Not run:
data <- rpois(100, rlnorm(3, 0.2))

strings <- run.model(model="LP", data=data, call.jags=FALSE)
modelstring <- strings[[1]]
datastring <- strings[[2]]
initial.values <- strings[[3]]
monitors <- strings[[4]]

run.jags(model=modelstring, inits=initial.values, check.conv=TRUE,
data=datastring, monitor=monitors)

# This is equivalent to using:

run.model(model="LP", data=data, call.jags=TRUE, check.conv=TRUE)

## End(Not run)
```

---

count.power

*Count Data Power Analysis Calculations*


---

### Description

Finds the power for a faecal egg count study with the given combination of parameters. This represents the probability that the observed empirical mean FEC will lie between the lower.limit and upper.limit specified. The power is calculated using the negative binomial distribution when considering the true mean of a single individual, or using Monte Carlo integration for more than one animal. Confidence intervals for the true power are produced for the latter.

### Usage

```
count.power(meanepg=200, g.faeces=3, sensitivity=1/25,
  replicates=1, animals=10, coeffvarrep=0.4, coeffvarind=0.3,
  coeffvargroup=0.7, true.sample=FALSE, accuracy=0.1,
  lower.limit=meanepg*(1-accuracy), upper.limit=meanepg*(1+accuracy),
  maxiterations=1000000, precision=2, confidence = 0.99,
  feedback=FALSE, forcesim=FALSE)
```

### Arguments

meanepg	the mean egg count of the group (in EPG). If this is unknown then use a value likely to be encountered, or iterate over a distribution of values to obtain a distribution of values for power.
g.faeces	the number of grams of faeces used per sample (must be the same for all samples).
sensitivity	the minimum egg detection threshold or counting sensitivity of the technique used. If using the McMasters technique, this is the number of McMasters chambers counted divided by 50.
replicates	the number of different samples (individually analysed) taken from each animal. Must be the same for all animals. This would normally be 1, but increasing this provides an effective way of improving power at the cost of additional laboratory work.
animals	the number of animals in the group. Can be 1, in which case the ability to predict the true mean of the animal (if true.sample=TRUE) or the true mean of a group from which the animal is taken (if true.sample=FALSE) can be calculated.
coeffvarrep	coefficient of variation between sub-samples taken from the same faecal sample, assuming 1g faeces is used per sub-sample (the effective value is automatically adjusted according to g.faeces in the function). The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals.
coeffvarind	coefficient of variation between samples taken from different faecal piles from the same animal over a period of days, not including that due to coeffvarrep. The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals.

<code>coeffvargroup</code>	coefficient of variation between animals. This includes the variability between samples taken from different animals within a group, not including that due to within animal variability. The default value for this is inferred from a combination of several equine datasets and an intensive study of a small group of horses, but may not be accurate in other groups of animals.
<code>true.sample</code>	option to calculate the power for the population mean (if <code>true.sample=FALSE</code> ) or the true sample mean (if <code>true.sample=TRUE</code> ). The difference is that conceptually the true mean of a small group of animals may not reflect the mean of the population from which they are derived. If only one animal is considered, the true sample mean is the true mean of the individual whereas the population mean is the mean of a (theoretical) group of animals from which it is derived. The power will always be greater (or identical) for the true sample mean.
<code>accuracy</code>	specify the lower and upper limits as the <code>meanepg +/-</code> this fraction of the <code>meanepg</code> . Ignored if non-default values are specified for <code>lower.limit</code> or <code>upper.limit</code> .
<code>lower.limit</code>	lower limit for the tolerance. Specifying <code>lower.limit</code> and <code>upper.limit</code> separately allows non-symmetrical tolerance around the mean with which to calculate the power. The default is to use <code>meanepg*(1-accuracy)</code> .
<code>upper.limit</code>	upper limit for the tolerance. Specifying <code>lower.limit</code> and <code>upper.limit</code> separately allows non-symmetrical tolerance around the mean with which to calculate the power. The default is to use <code>meanepg*(1+accuracy)</code> .
<code>maxiterations</code>	the maximum number of iterations to use for the Monte Carlo integration, or the number to use if <code>precision=NA</code> . If <code>precision</code> is defined, then only the number of iterations required to estimate the power to the given precision are performed (up to a maximum of <code>maxiterations</code> ).
<code>precision</code>	the number of decimal places with which to calculate the power, unless <code>maxiterations</code> is reached first. A larger precision will give a more precise estimate of the true power but will take longer to calculate. Specifying this as <code>NA</code> performs the calculation on a fixed ( <code>=maxiterations</code> ) number of iterations.
<code>confidence</code>	the degree of confidence required with which to report confidence limits for the true power when using Monte Carlo integration to report the power.
<code>feedback</code>	option to display a progress indicator for calculation of the values used for Monte Carlo integration. Using <code>feedback</code> with some GUI versions of R may slow down the analysis considerably.
<code>forcesim</code>	option to force the function to use the Monte Carlo method of approximating the power when the default would be to use a negative binomial approximation (when <code>true.sample=FALSE</code> and <code>coeffvargroup</code> is very small, or when <code>true.sample=TRUE</code> and <code>animals=1</code> ). Results using the two methods should be very similar, however there may be differences due to the different optimisation mechanisms when calculating both lower and upper limits. In addition, Monte Carlo integration will take longer and gives a confidence interval for the true mean rather than the absolute value.

### Value

Returns a list containing the elements `'roundedci'` and `'ci'`, which specifies the median and confidence limits (as defined by `'confidence'`) for the true power both rounded by `'precision'` and unrounded. For analyses using the Monte Carlo integration method, `'within'` `'without'` and `'total'` are



also returned, and indicate the number of iterations for which the observed mean fell outside and inside the specified limits and the total number of iterations.

### See Also

[fec.power](#), [fecrt.analysis](#)

---

count.precision

*Count Data Predicted Precision Calculations*

---

### Description

Finds the appropriate tolerance with which to consider the observed mean for a faecal egg count study with the given combination of power and other parameters. The precision is calculated using the negative binomial distribution when considering the true mean of a single individual, or using Monte Carlo integration for more than one animal. Confidence intervals for the true power are produced for the latter. Tolerance can be defined as either a lower limit only if upper.limit is defined, as an upper limit only if lower.limit is defined, or as both (equidistant) limits if neither are defined.

### Usage

```
count.precision(meanepg=200, g.faeces=3, sensitivity=1/25,
  replicates=1, animals=10, coeffvarrep=0.4, coeffvarind=0.3,
  coeffvargroup=0.7, true.sample=FALSE, lower.limit=NA,
  upper.limit=NA, iterations=100000, power = 0.95,
  confidence = 0.99, feedback=FALSE, forcesim=FALSE)
```

### Arguments

meanepg	the mean egg count of the group (in EPG). If this is unknown then use a value likely to be encountered, or iterate over a distribution of values to obtain a distribution of values for tolerance.
g.faeces	the number of grams of faeces used per sample (must be the same for all samples).
sensitivity	the minimum egg detection threshold or counting sensitivity of the technique used. If using the McMasters technique, this is the number of McMasters chambers counted divided by 50.
replicates	the number of different samples (individually analysed) taken from each animal. Must be the same for all animals. This would normally be 1, but increasing this provides an effective way of improving power at the cost of additional laboratory work.
animals	the number of animals in the group. Can be 1, in which case the ability to predict the true mean of the animal (if true.sample=TRUE) or the true mean of a group from which the animal is taken (if true.sample=FALSE) can be calculated.

<code>coeffvarrep</code>	coefficient of variation between sub-samples taken from the same faecal sample, assuming 1g faeces is used per sub-sample (the effective value is automatically adjusted according to <code>g.faeces</code> in the function). The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals.
<code>coeffvarind</code>	coefficient of variation between samples taken from different faecal piles from the same animal over a period of days, not including that due to <code>coeffvarrep</code> . The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals.
<code>coeffvargroup</code>	coefficient of variation between animals. This includes the variability between samples taken from different animals within a group, not including that due to within animal variability. The default value for this is inferred from a combination of several equine datasets and an intensive study of a small group of horses, but may not be accurate in other groups of animals.
<code>true.sample</code>	option to calculate the power for the population mean (if <code>true.sample=FALSE</code> ) or the true sample mean (if <code>true.sample=TRUE</code> ). The difference is that conceptually the true mean of a small group of animals may not reflect the mean of the population from which they are derived. If only one animal is considered, the true sample mean is the true mean of the individual whereas the population mean is the mean of a (theoretical) group of animals from which it is derived. The power will always be greater (or identical) for the true sample mean.
<code>lower.limit</code>	(optional) lower limit for the tolerance. If this is supplied, the required upper limit to maintain the desired power with the specified lower limit will be calculated. If neither lower limit or upper limit are specified, symmetrical limits about the mean are found that satisfy the power condition.
<code>upper.limit</code>	(optional) upper limit for the tolerance. If this is supplied, the required lower limit to maintain the desired power with the specified upper limit will be calculated. If neither lower limit or upper limit are specified, symmetrical limits about the mean are found that satisfy the power condition.
<code>iterations</code>	the number of iterations to use for the Monte Carlo integration. More iterations will take longer but provide a smaller confidence interval for the true power.
<code>power</code>	the desired power with which to interpret the given tolerances. Default is 0.95, so that the observed mean FEC will lie within the calculated lower and upper limits 95% of the time.
<code>confidence</code>	the degree of confidence required with which to report confidence limits for the true power when using Monte Carlo integration to report the power. Only used for calculating the true power after determining the tolerance limits.
<code>feedback</code>	option to display a progress indicator for calculation of the values used for Monte Carlo integration. Using feedback with some GUI versions of R may slow down the analysis considerably.
<code>forcesim</code>	option to force the function to use the Monte Carlo method of approximating the power when the default would be to use a negative binomial approximation (when <code>true.sample=FALSE</code> and <code>coeffvargroup</code> is very small, or when <code>true.sample=TRUE</code> and <code>animals=1</code> ). Results using the two methods should be very similar, however there may be differences due to the different optimisation mechanisms when calculating both lower and upper limits. In addition, Monte

Carlo integration will take longer and gives a confidence interval for the true mean rather than the absolute value.

### Value

Returns a list containing the elements 'limits', which is the calculated lower and upper tolerance level which provides the required power, and 'power' which specifies the median estimate and confidence intervals for the true power when using Monte Carlo integration, or the absolute value (replicated 3 times for consistency) if using the negative binomial approximation method. The true power returned may not exactly match the required power input due to the integer nature of FEC data.

### See Also

[count.power](#), [count.analysis](#)

---

count_analysis	<i>Count data analysis</i>
----------------	----------------------------

---

### Description

Analyses count data for difference to either known or fixed mean using any of (1) Christian approach, (2) my approach, (3) bootstrap, (4) MCMC (default 1-3). (4) calls count\_model and then runs the model using runjags, and is the only one to permit ZI and LP. Requires Just Another Gibbs Sampler (JAGS) (see <http://mcmc-jags.sourceforge.net>)

### Usage

```
count_analysis(data = stop("Data must be specified"), model = "ZILP",
  alt.prior = FALSE, adjust.zi.mean = FALSE, raw.output = FALSE,
  likelihood = FALSE, ...)
```

### See Also

[count\\_model](#), [reduction\\_analysis](#), [bayescount](#), [autoextend.jags](#), [runjags](#)

---

count_model	<i>Count data model</i>
-------------	-------------------------

---

### Description

Creates a count model for the specified data and returns it as an un-run runjags object

**Usage**

```
count_model(data = stop("No data supplied"),
  model = stop("No model specified"), call.jags = FALSE,
  alt.prior = FALSE, monitor.lambda = FALSE, monitor.deviance = FALSE,
  ...)
```

**See Also**

[count\\_analysis](#), [reduction\\_model](#), [bayescount](#), [autoextend.jags](#), [runjags](#)

count\_power

*Count data power calculations*

**Description**

Power calculation based on BNB code and christian approach yet to be written. Do either test vs known single mean, or test for difference in mean to observed dataset (k can vary between datasets or be fixed).

**Usage**

```
count_power(type = "fec", cutoff = if ("fec") 400 else 95,
  alternative = "two.sided", conf.level = 0.05, maxiterations = 1e+06,
  mc.decimals = 2, mc.conf = 0.99, feedback = FALSE, ...)
```

**See Also**

[count\\_analysis](#), [reduction\\_power](#), [bayescount](#)

count\_precision

*Count data precision calculations*

**Description**

Precision analysis by Monte Carlo simulation and examination of distribution of mean

**Usage**

```
count_precision(meanepg = 200, g.faeces = 3, sensitivity = 1/25,
  replicates = 1, animals = 10, coeffvarmcm = 0.1, coeffvarrep = 0.4,
  coeffvarind = 0.3, coeffvargroup = 0.7, true.sample = FALSE,
  type = "confidence", precision = if (type == "confidence") 0.1 else NA,
  lower.interval = meanepg * (1 - precision), upper.interval = meanepg * (1
  + precision), confidence = 0.95, iterations = if (type == "confidence")
  10^6 else 10^5, mc.decimals = 2, mc.conf = 0.99, feedback = FALSE,
  forcesim = FALSE)
```

**See Also**

[count\\_analysis](#), [count\\_power](#), [bayescount](#)

---

 fecrt.power

*FECRT Power Analysis Calculations*


---

**Description**

Finds the power for a faecal egg count reduction test with the given combination of parameters. This represents the probability that the observed empirical mean reduction will lie between the lower.limit and upper.limit specified. The power is calculated using Monte Carlo integration and confidence intervals for the true power are produced. This function can be used to determine the probability that the observed empirical mean reduction will lie within a the range (for example) 90% to 100% if the true reduction is 80% (the false negative rate if considering anthelmintic resistance), or the probability that the observed empirical mean reduction will lie within a the range (for example) 0% to 90% if the true reduction is 95% (the false positive rate if considering anthelmintic resistance)

**Usage**

```
fecrt.power(meanepg=200, reduction = 80, g.faeces=3,
  sensitivity=1/25, replicates=1, animals=10, pre.coeffvarrep=0.4,
  pre.coeffvarind=0.3, pre.coeffvargroup=0.7, post.coeffvarrep=0.4,
  post.coeffvarind=0.3, post.coeffvargroup=0.7, true.sample=FALSE,
  lower.limit=0, upper.limit=90, maxiterations=1000000,
  precision=2, confidence = 0.99, feedback=FALSE)
```

**Arguments**

meanepg	the mean pre-treatment egg count of the group (in EPG). If this is unknown then use a value likely to be encountered, or iterate over a distribution of values to obtain a distribution of values for power.
reduction	the true mean egg count reduction (in %). If this is unknown then use a value likely to be encountered, or iterate over a distribution of values to obtain a distribution of values for power.
g.faeces	the number of grams of faeces used per sample (must be the same for all samples both pre and post treatment).
sensitivity	the minimum egg detection threshold or counting sensitivity of the technique used. If using the McMasters technique, this is the number of McMasters chambers counted divided by 50.
replicates	the number of different samples (individually analysed) taken from each animal. Must be the same for all animals, both pre and post treatment. This would normally be 1, but increasing this provides an effective way of improving power at the cost of additional laboratory work.

animals	the number of animals in the group. Can be 1, in which case the ability to predict the true mean reduction for the animal (if true.sample=TRUE) or the true mean reduction for a group from which the animal is taken (if true.sample=FALSE) can be calculated. The same number of animals MUST be sample pre and post treatment.
pre.coeffvarrep	coefficient of variation between sub-samples taken from the same faecal sample, assuming 1g faeces is used per sub-sample (the effective value is automatically adjusted according to g.faeces in the function). The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the pre-treatment distributions.
pre.coeffvarind	coefficient of variation between samples taken from different faecal piles from the same animal over a period of days, not including that due to coeffvarrep. The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the pre-treatment distributions.
pre.coeffvargroup	coefficient of variation between animals. This includes the variability between samples taken from different animals within a group, not including that due to within animal variability. The default value for this is inferred from a combination of several equine datasets and an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the pre-treatment distributions.
post.coeffvarrep	coefficient of variation between sub-samples taken from the same faecal sample, assuming 1g faeces is used per sub-sample (the effective value is automatically adjusted according to g.faeces in the function). The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the post-treatment distributions.
post.coeffvarind	coefficient of variation between samples taken from different faecal piles from the same animal over a period of days, not including that due to coeffvarrep. The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the post-treatment distributions.
post.coeffvargroup	coefficient of variation between animals. This includes the variability between samples taken from different animals within a group, not including that due to within animal variability. The default value for this is inferred from a combination of several equine datasets and an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the post-treatment distributions.
true.sample	option to calculate the power for the population mean reduction (if true.sample=FALSE) or the true sample mean reduction (if true.sample=TRUE). The difference is that conceptually the true mean of a small group of animals may not reflect the mean

	of the population from which they are derived. If only one animal is considered, the true sample mean is the true mean of the individual whereas the population mean is the mean of a (theoretical) group of animals from which it is derived. The power will always be greater (or identical) for the true sample mean.
lower.limit	lower limit for the tolerance (in %).
upper.limit	lower limit for the tolerance (in %).
maxiterations	the maximum number of iterations to use for the Monte Carlo integration, or the number to use if precision=NA. If precision is defined, then only the number of iterations required to estimate the power to the given precision are performed (up to a maximum of maxiterations).
precision	the number of decimal places with which to calculate the power, unless maxiterations is reached first. A larger precision will give a more precise estimate of the true power but will take longer to calculate. Specifying this as NA performs the calculation on a fixed (=maxiterations) number of iterations.
confidence	the degree of confidence required with which to report confidence limits for the true power when using Monte Carlo integration to report the power.
feedback	option to display a progress indicator for calculation of the values used for Monte Carlo integration. Using feedback with some GUI versions of R may slow down the analysis considerably.

### Value

Returns a list containing the elements 'roundedci' and 'ci', which specifies the median and confidence limits (as defined by 'confidence') for the true power both rounded by 'precision' and unrounded. The additional variables 'within' 'without' and 'total' are also returned, and indicate the number of iterations for which the observed mean reduction fell outside and inside the specified limits and the total number of iterations.

### See Also

[fecrt.precision](#), [fec.analysis](#)

---

fecrt.precision	<i>FECRT Predicted Precision Calculations</i>
-----------------	---

---

### Description

Finds the appropriate tolerance with which to consider the observed mean for a faecal egg count reduction test with the given combination of power and other parameters. The precision is calculated using Monte Carlo integration, and confidence intervals for the true power are produced. Tolerance can be defined as either a lower limit only if upper.limit is defined, as an upper limit only if lower.limit is defined, or as both (equidistant) limits if neither are defined. This function can be used for example to determine the upper limit for a true reduction of 80% by setting lower.limit=0, or to determine the lower limit for a true reduction of 99% by setting upper.limit=100.

**Usage**

```
fecrt.precision(meanepg=200, reduction = 95, g.faeces=3,
  sensitivity=1/25, replicates=1, animals=10, pre.coeffvarrep=0.4,
  pre.coeffvarind=0.3, pre.coeffvargroup=0.7, post.coeffvarrep=0.4,
  post.coeffvarind=0.3, post.coeffvargroup=0.7, true.sample=FALSE,
  lower.limit=NA, upper.limit=NA, iterations=100000, power = 0.95,
  confidence = 0.99, feedback=FALSE)
```

**Arguments**

meanepg	the mean pre-treatment egg count of the group (in EPG). If this is unknown then use a value likely to be encountered, or iterate over a distribution of values to obtain a distribution of values for tolerance.
reduction	the true mean egg count reduction (in %). If this is unknown then use a value likely to be encountered, or iterate over a distribution of values to obtain a distribution of values for tolerance.
g.faeces	the number of grams of faeces used per sample (must be the same for all samples both pre and post treatment).
sensitivity	the minimum egg detection threshold or counting sensitivity of the technique used. If using the McMasters technique, this is the number of McMasters chambers counted divided by 50.
replicates	the number of different samples (individually analysed) taken from each animal. Must be the same for all animals, both pre and post treatment. This would normally be 1, but increasing this provides an effective way of improving power at the cost of additional laboratory work.
animals	the number of animals in the group. Can be 1, in which case the ability to predict the true mean reduction for the animal (if true.sample=TRUE) or the true mean reduction for a group from which the animal is taken (if true.sample=FALSE) can be calculated. The same number of animals MUST be sample pre and post treatment.
pre.coeffvarrep	coefficient of variation between sub-samples taken from the same faecal sample, assuming 1g faeces is used per sub-sample (the effective value is automatically adjusted according to g.faeces in the function). The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the pre-treatment distributions.
pre.coeffvarind	coefficient of variation between samples taken from different faecal piles from the same animal over a period of days, not including that due to coeffvarrep. The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the pre-treatment distributions.
pre.coeffvargroup	coefficient of variation between animals. This includes the variability between samples taken from different animals within a group, not including that due to



within animal variability. The default value for this is inferred from a combination of several equine datasets and an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the pre-treatment distributions.

<code>post.coeffvarrep</code>	coefficient of variation between sub-samples taken from the same faecal sample, assuming 1g faeces is used per sub-sample (the effective value is automatically adjusted according to <code>g.faeces</code> in the function). The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the post-treatment distributions.
<code>post.coeffvarind</code>	coefficient of variation between samples taken from different faecal piles from the same animal over a period of days, not including that due to <code>coeffvarrep</code> . The default value for this is taken from an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the post-treatment distributions.
<code>post.coeffvargroup</code>	coefficient of variation between animals. This includes the variability between samples taken from different animals within a group, not including that due to within animal variability. The default value for this is inferred from a combination of several equine datasets and an intensive study of a small group of horses, but may not be accurate in other groups of animals. This value is applied only to the post-treatment distributions.
<code>true.sample</code>	option to calculate the power for the population mean (if <code>true.sample=FALSE</code> ) or the true sample mean (if <code>true.sample=TRUE</code> ). The difference is that conceptually the true mean of a small group of animals may not reflect the mean of the population from which they are derived. If only one animal is considered, the true sample mean is the true mean of the individual whereas the population mean is the mean of a (theoretical) group of animals from which it is derived. The power will always be greater (or identical) for the true sample mean.
<code>lower.limit</code>	(optional) lower limit for the tolerance (in %). If this is supplied, the required upper limit to maintain the desired power with the specified lower limit will be calculated. If neither lower limit or upper limit are specified, symmetrical limits about the true mean reduction are found that satisfy the power condition.
<code>upper.limit</code>	(optional) upper limit for the tolerance (in %). If this is supplied, the required lower limit to maintain the desired power with the specified upper limit will be calculated. If neither lower limit or upper limit are specified, symmetrical limits about the true mean reduction are found that satisfy the power condition.
<code>iterations</code>	the number of iterations to use for the Monte Carlo integration. More iterations will take longer but provide a smaller confidence interval for the true power.
<code>power</code>	the desired power with which to interpret the given tolerances. Default is 0.95, so that the observed mean FEC will lie within the calculated lower and upper limits 95% of the time.
<code>confidence</code>	the degree of confidence required with which to report confidence limits for the true power when using Monte Carlo integration to report the power. Only used for calculating the true power after determining the tolerance limits.

feedback option to display a progress indicator for calculation of the values used for Monte Carlo integration. Using feedback with some GUI versions of R may slow down the analysis considerably.

### Value

Returns a list containing the elements 'limits', which is the calculated lower and upper tolerance level which provides the required power, and 'power' which specifies the median estimate and confidence intervals for the true power. The true power returned may not exactly match the required power input due to the integer nature of FECRT data.

### See Also

[fecrt.power](#), [fecrt.analysis](#)

---

likelihood	<i>Calculate the (Log) Likelihood of Obtaining Data from a Distribution</i>
------------	---

---

### Description

Function to calculate the (log) likelihood of obtaining a set of data from one of the following distributions: Poisson (P), gamma (G), lognormal (L), Weibull (W), gamma Poisson (GP), lognormal Poisson (LP), Weibull Poisson (WP), all with or without zero-inflation (ZI). For mixture models, the likelihood is calculated for the data by integrating over each possible value of lambda for each data point, which may take some time for large datasets. \*THIS FUNCTION IS DEPRECATED AND WILL BE REMOVED FROM BAYESCOUNT VERSION 1\*

### Usage

```
likelihood(model, data, mean=NA, variance=NA,
           zi=NA, shape=NA, scale=NA,
           iterations=min(1000, (length(as.matrix(mean)[,1])*length(shape))),
           log=TRUE, silent=FALSE, raw.output=FALSE)
```

### Arguments

model	the distribution to fit to the data. Choices are: 'IP', 'P', 'ZIP', 'G', 'ZIG', 'L', 'ZIL', 'W', 'ZIW', 'GP', 'ZIGP', 'LP', 'ZILP', 'WP', 'ZIWP' (case insensitive). No default.
data	a vector of data to fit the distribution to. Count data for the count models, continuous data for the continuous distributions.
mean	a vector of mean values over which to calculate the likelihood. Must be a matrix with number of columns equal to the number of datapoints for the IP model only, since using the IP model each datapoint has an independant mean (this is effectively a shortcut for repeating the likelihood function for each datapoint). Mean is required for the IP, (ZI)P, (ZI)L, and (ZI)L models. Otherwise ignored (with a warning if silent=FALSE).

variance	a vector of values for variance over which to calculate the likelihood, with length equal to length of mean. Required for the (ZI)L, and (ZI)L models. Otherwise ignored (with a warning if silent=FALSE).
zi	a vector of values for zero-inflation, with length equal to either scale/shape or mean/variance as appropriate. Required for zero-inflated models only; if supplied for other models the appropriate zero-inflated model will be used instead (with a warning if silent=FALSE).
shape	a vector of values for the shape parameter. Must be of length equal to that of the scale parameter. Required for the (ZI)W, (ZI)WP, (ZI)G, and (ZI)GP models. Otherwise ignored (with a warning if silent=FALSE).
scale	a vector of values for the scale parameter. Must be of length equal to that of the shape parameter. Required for the (ZI)W, (ZI)WP, (ZI)G, and (ZI)GP models. Otherwise ignored (with a warning if silent=FALSE).
iterations	the total number of iterative samples for which to calculate the likelihood. If this is smaller than the number of iterations provided for mean/variance/shape/scale, then the chain will be thinned to the number of iterations required (this is useful for the mixture models where integrating the likelihood for each datapoint may take some time). Default is 1000 iterations, or the length of the parameters supplied if this is shorter than 1000.
log	choose to output the log likelihood (TRUE) or the likelihood (FALSE). Default TRUE.
silent	should warning messages and progress indicators be suppressed? Default FALSE.
raw.output	output a vector of length equal to the number of iterations representing the likelihood at each iteration if TRUE, or a summary of the results (median and 95 percent confidence interval) if FALSE. Default FALSE.

### Value

The (log) likelihood is returned either as a vector of length equal to the number of iterations representing the likelihood at each iteration if `raw.output==TRUE`, or a summary of the results (median, maximum and 95% highest posterior density interval (see `HPDinterval`) if `raw.output==FALSE`. It is possible for the functions that use integrate to produce incalculable probabilities for some iterations, in which case the likelihood for these iterations are returned as missing data (and therefore the summary statistics for the whole simulation are also returned as missing data). If only 1 iteration of values for mean/variance/shape/scale are provided, a single value for likelihood is output.

### See Also

[count.analysis](#)

### Examples

```
# calculate the likelihood of obtaining a set of count data from
# a zero-inflated Poisson distribution with set mean and zero-inflation
# values
## Not run:
data <- rpois(100, 10)
data[1:15] <- 0
```

```

likelihood('ZIP', data, mean=10, zi=15)

# now calculate the likelihood for the same data using an MCMC object
# to provide the values for mean and zero-inflation

values <- bayescount.single(data, model='ZISP', raw.output=TRUE)
means <- c(values[, 'mean'][[1]], values[, 'mean'][[2]])
zis <- (1-c(values[, 'prob'][[1]], values[, 'prob'][[2]]))*100
# The function outputs the prevalence of disease when raw.ouput is
# TRUE, so zero-inflation must be calculated from this

likes <- likelihood('ZIP', data, mean=means, zi=zis,
raw.output=TRUE)$likelihood
hist(likes, breaks='fd', col='red')

## End(Not run)

```

---

Inormal.params

*Calculate the Log-Normal Mean and Standard Deviation Using the Normal Mean and Standard Deviation*


---

## Description

Function to calculate the equivalent values for the mean and standard deviation of a log-normal distribution from the mean and standard deviation of the normal distribution. Outputs from this function can be used with the `dlnorm()` function, and with the lognormal distribution in JAGS.

## Usage

```
Inormal.params(mean, sd, coeff.variation)
```

## Arguments

mean	either a single value or vector of values for the mean of the normal distribution.
sd	either a single value or vector of values for the standard deviation of the normal distribution. Ignored if values are supplied for <code>coeff.variation</code> .
coeff.variation	either a single value or vector of values for the coefficient of dispersion.

## Value

A list with elements representing the mean of the lognormal distribution, the standard deviation of the lognormal distribution, and the coefficient of variation.

## See Also

[normal.params](#)

**Examples**

```
## Not run:
mean <- 10
sd <- 2
lmean <- lnormal.params(mean,sd)[[1]]
lsd <- lnormal.params(mean,sd)[[2]]

curve(dnorm(x, mean, sd), from=0, to=20)
dev.new()
curve(dlnorm(x, lmean, lsd), from=0, to=20)

## End(Not run)
```

---

lnormal_params	<i>Calculate the Log-Normal Mean and Standard Deviation Using the Normal Mean and Standard Deviation and Vice Versa</i>
----------------	---

---

**Description**

Function to calculate the equivalent values for the mean and standard deviation of a normal distribution from the mean and standard deviation of the log-normal distribution and vice versa. Outputs from this function can be used with the dnorm() or dlnorm functions, and with the equivalent distributions in JAGS.

**Usage**

```
normal_params(log.mean, log.sd, coeff.variation = sqrt(exp(log.sd^2) - 1))
```

**Examples**

```
lmean <- 2.5
lsd <- 0.2
mean <- normal.params(lmean,lsd)[[1]]
sd <- normal.params(lmean,lsd)[[2]]

curve(dlnorm(x, lmean, lsd), from=0, to=25)
dev.new()
curve(dnorm(x, mean, sd), from=0, to=25)

mean <- 10
sd <- 2
lmean <- lnormal.params(mean,sd)[[1]]
lsd <- lnormal.params(mean,sd)[[2]]

curve(dnorm(x, mean, sd), from=0, to=20)
dev.new()
curve(dlnorm(x, lmean, lsd), from=0, to=20)
```

---

maximise.likelihood     *Calculate the Maximum Likelihood Parameters of a Continuous or Count Distribution*

---

### Description

Crude function to maximise the likelihood of one of the following distributions: Poisson (P), gamma (G), lognormal (L), Weibull (W), gamma Poisson (GP), lognormal Poisson (LP), Weibull Poisson (WP), all with or without zero-inflation (ZI). Uses the likelihood() function to calculate the likelihood at each iteration. For mixture models, the likelihood is calculated for the data by integrating over each possible value of lambda for each data point, which may take some time for large datasets. Starting values for each parameter are optional, but may improve the speed and reliability of the function if appropriate values are provided. If missing, starting values will be calculated from the data. This function is provided for interest only, and is vastly inferior as a data analysis tool to the Bayesian MCMC methods used by bayescount(). \*THIS FUNCTION IS DEPRECATED AND WILL BE REMOVED FROM BAYESCOUNT VERSION 1 - SEE ?count.analysis FOR AN ALTERNATIVE USING MCMC\*

### Usage

```
maximise.likelihood(data=stop("Data must be specified"),
  model=stop("Please specify a distribution"), mean=NA,
  variance=NA, zi=NA, shape=NA, scale=NA, silent=FALSE)
```

### Arguments

data	a vector of data to fit the distribution to. Count data for the count models, continuous data for the continuous distributions.
model	the distribution to fit to the data. Choices are: "P", "ZIP", "G", "ZIG", "L", "ZIL", "W", "ZIW", "GP", "ZIGP", "LP", "ZILP", "WP", "ZIWP" (case insensitive). No default.
mean	the starting value for mean. Optional.
variance	the starting value for variance. Optional.
zi	the starting value for zero-inflation. Optional.
shape	the starting value for the shape parameter. Optional.
scale	the starting value for the scale parameter. Optional.
silent	should warning messages and progress indicators be suppressed? Default FALSE.

### Value

The values for each parameter at the maximum likelihood are output. No standard error is given (Bayesian MCMC using bayescount() gives a much better analysis)

### See Also

[count.analysis](#)

**Examples**

```
# obtain values for mean and zero-inflation of a zero-inflated
# gamma Poisson model:

data <- rpois(100, rgamma(100, shape=1, scale=8))
data[1:15] <- 0

#maximise.likelihood(data, "ZIGP")
```

---

normal.params	<i>Calculate the Normal Mean and Standard Deviation Using the Log-Normal Mean and Standard Deviation</i>
---------------	--

---

**Description**

Function to calculate the equivalent values for the mean and standard deviation of a normal distribution from the mean and standard deviation of the log-normal distribution. Outputs from this function can be used with the `dnorm()` function, and with the normal distribution in JAGS.

**Usage**

```
normal.params(log.mean, log.sd, coeff.variation=sqrt(exp(log.sd^2)-1))
```

**Arguments**

log.mean	either a single value or vector of values for the mean of the lognormal distribution.
log.sd	either a single value or vector of values for the standard deviation of the lognormal distribution. Ignored if values are supplied for <code>coeff.variation</code> .
coeff.variation	either a single value or vector of values for the coefficient of dispersion.

**Value**

A list with elements representing the mean of the normal distribution, the standard deviation of the normal distribution, and the coefficient of variation.

**See Also**

[lnormal.params](#)

**Examples**

```
## Not run:
lmean <- 2.5
lsd <- 0.2
mean <- normal.params(lmean,lsd)[[1]]
sd <- normal.params(lmean,lsd)[[2]]

curve(dlnorm(x, lmean, lsd), from=0, to=25)
dev.new()
curve(dnorm(x, mean, sd), from=0, to=25)

## End(Not run)
```

---

reduction_analysis	<i>Analyse two sets of count data (such as FECRT) compared to a desired mean reduction</i>
--------------------	--

---

**Description**

Analyses two sets of count data for using any of (1) Christian approach, (2) my approach, (3) bootstrap, (4) MCMC (default 1-3). (4) calls `reduction_model` and then runs the model using `run-jags`, and is the only one to permit ZI and LP. Requires Just Another Gibbs Sampler (JAGS) (see <http://mcmc-jags.sourceforge.net>)

**Usage**

```
reduction_analysis(name = NA, pre.data = NA, post.data = NA,
  data = list(pre = pre.data, post = post.data), animal.names = FALSE,
  efficacy = 95, confidence = 95, restrict.efficacy = TRUE,
  control.animals = FALSE, paired.model = FALSE, fix.variation = FALSE,
  fix.efficacy = TRUE, zero.inflation = FALSE,
  individual.analysis = FALSE, divide.data = 1, record.chains = FALSE,
  write.file = FALSE, bootstrap.iter = 10000, plot.graph = TRUE,
  skip.mcmc = NULL, stat.method = c("MCMC", "WAAVP", "Bootstrap"),
  custom.model = NA, ...)
```

**Arguments**

name	a name for the analysis (character). Missing by default (function will require it to be input).
pre.data	the pre-treatment data. Either a numeric vector, amatrix with repeated McMasters counts from the same sample (each row) indifferent columns, or an array with different animals in dimension 1,repeat samples from the same animal in dimension 2, repeated McMasterscounts from the same sample in dimension 3 (can be of length 1 if only 1count recorded). If an array, then the use.paired must be TRUE. Noddefault. Ignored if a value is specified for data.



post.data	the post-treatment data, for acceptable formats see pre.data.
data	either a path to a comma delimited csv file, or an existing R object containing the data. Data can be specified in one of the following ways. If a matrix, the first column is pre-treatment data, the second column is the post-treatment data, and the third column (if supplied) indicates control (1) or treatment (0) animal. Alternatively, the first column is animal names if animal.names = TRUE, with columns 2 and 3 making up the pre- and post-treatment data and optionally column 4 the treatment of control status. If the data is specified as a list, then the first element is taken as the pre-treatment data, and the second element is taken as the post-treatment data. These elements of the list can be provided as for pre.data and post.data. If the data is in a comma delimited file, it must be in the format specified for a single matrix. Missing data, or unused elements of non-ragged arrays, may be represented using NA, which will be removed from the data before analysis. This argument is taken from the specified values of pre.data and post.data by default. If a value is specified for data then arguments specified for pre.data and post.data are ignored.
animal.names	either a character vector of names to be used for each animal specified in the data, TRUE or FALSE. If TRUE, then animal names are taken from the first column of the matrix supplied for data (a matrix must be supplied for data if animal.names is TRUE). If FALSE, then animal names are generated automatically. Default FALSE.
efficacy	the target % efficacy of the drug used. Used to calculate the probability of resistance with the MCMC and bootstrap methods. Default 95%.
confidence	the degree of confidence required with which to report the confidence limits for the true FEC reduction.
restrict.efficacy	option to restrict the estimate of efficacy from the MCMC method to values of greater than 0 (i.e. the mean faecal egg count cannot increase after treatment). If TRUE, then the prior for change in mean egg count is uniform between 0 and 1, and if FALSE then the prior for change in mean egg count is uniform between 0 and 10. If using control animals, then restrict.efficacy applies to the change in mean of the treated animals only, allowing the efficacy to be below 0. Default TRUE.
control.animals	indication of which animals are to be used as controls. Should be either a vector of TRUE/FALSE (or 1/0) corresponding to whether each animal is a control (TRUE) or treatment (FALSE), or simply FALSE (the default) in which case all animals are assumed to be treated. Ignored if data is specified as a matrix (or csv-file) with 3 columns, in which case the third column should reflect the treatment status. Default FALSE.
paired.model	the FECRT can be run using two compound gamma distributions to describe the variability within and between animals in place of the single gamma distribution combining the two sources of variability. When using two compound gamma distributions for pre-treatment data, the post-treatment data are paired to the pre-treatment data by animal. The advantage of using a single distribution is that the model is more identifiable, and therefore is likely to converge more quickly and return errors less frequently. The advantage of the paired model is that it

	allows repeat measurements to be incorporated in the model, and non-randomly missing data (ie. protocols that involve post-treatment sampling of only animals with a high pre-treatment count) to be modelled appropriately. The simple model cannot be used when using repeat samples within animal, and will provide inaccurate results if animals are targeted for post-treatment sampling based on their pre-treatment count. Default uses the simple model (FALSE).
<code>zero.inflation</code>	option to use a zero-inflated gamma Poisson in place of the gamma Poisson distribution. If TRUE, zero inflated distributions are used pre- and post-treatment (with the prevalence fixed between pre- and post-treatment distributions). Default FALSE.
<code>divide.data</code>	count division factor to allow egg count data in eggs per gram to be used raw (numeric). Default 1 (no transformation to data).
<code>record.chains</code>	option to allow the MCMC chains to be recorded for future use. If TRUE, the function returns the MCMC object as part of the return value (the MCMC object is not printed using the print method). If <code>write.file==TRUE</code> , the results are also saved using a filename containing the name of the analysis. Default FALSE.
<code>write.file</code>	option to write the results of the analysis to a text file using a filename containing the name of the analysis. The contents of the text file are identical to the return value of the function. Default FALSE.
<code>bootstrap.iters</code>	the number of bootstrap iterations to use for the bootstrap method. Default 10000.
<code>plot.graph</code>	an option to plot the posterior true egg count reduction from the MCMC method graphically. If <code>write.file==TRUE</code> then the graph is saved as a PDF, otherwise the graph is plotted on the active device. Default FALSE.
<code>skip.mcmc</code>	option to omit the MCMC analysis, and return bootstrap and WAAPV method analysis results alone. Default FALSE.
<code>...</code>	other options to be passed directly to <code>autorun.jags</code> .

### Details

Should probably discuss different methods here

### Value

Returns a list of the results obtained using each method, printed using a custom print method.

### References

M. J. Denwood, S. W. J. Reid, S. Love, M. K. Nielsen, L. Matthews, I. J. McKendrick, and G. T. Innocent. Comparison of three alternative methods for analysis of equine Faecal Egg Count Reduction Test data. *Prev. Vet. Med.* (2009), doi:10.1016/j.prevetmed.2009.11.009

### See Also

`reduction_model`, `count_analysis`, `bayescount`, `autoextend.jags`, `runjags` This function applies a Bayesian [zero-inflated] gamma Poisson (negative binomial) model to faecal egg count reduction test (FECRT) data to return possible values for the mean drug efficacy. Pre-treatment data

are assumed to arise from either a gamma-Poisson or zero-inflated gamma-Poisson distribution, with post-treatment data described by a separate gamma-Poisson or zero-inflated gamma-Poisson distribution. The change in mean between these distributions is therefore the mean egg count reduction. If `paired.model=TRUE`, a slightly different formulation is used whereby the observed count pre-treatment is assumed to follow a compound gamma-gamma-Poisson distribution with the variability within and between animals separated. The post treatment mean for each animal is derived from the pre-treatment animal mean and FEC reduction. This formulation allows data with non-random missing post-treatment counts to be analysed correctly, and also allows data with repeat counts from an individual to be analysed - providing a method of increasing the power of the method substantially. Results are also obtained using non-parametric bootstrapping and the method advocated in the 1992 World Association for the Advancement of Veterinary Parasitology (W.A.A.V.P.) methods for the detection of anthelmintic resistance in nematodes of veterinary importance (unless the data contains repeat values or missing values). Confidence intervals for the relevant statistics are printed to file if `write.file = TRUE`, and returned using a custom print method.

Lower level running of JAGS and assessing the simulation for convergence and required run length is done using `autorun.jags`. Note: The GUI interface for R in Windows may not continually refresh the output window, making it difficult to track the progress of the simulation (if `silent.jags` is `FALSE`). To avoid this, you can run the function from the terminal version of R (located in the Program Files/R/bin/ folder).

**\*THIS SOFTWARE IS INTENDED FOR EDUCATIONAL PURPOSES ONLY AND SHOULD NOT BE RELIED UPON FOR REAL WORLD APPLICATIONS\***

`fecrt.power` and `fecrt.power.limits` for power analysis methods for the FECRT, `bayescount`, `autorun.jags`

## Examples

```
#reduction_analysis()
```

---

reduction_model	<i>Generate an un-run MCMC model for FECRT data</i>
-----------------	---

---

## Description

Does not require Just Another Gibbs Sampler (JAGS) (see <http://mcmc-jags.sourceforge.net>) but not much use otherwise

## Usage

```
reduction_model(name = NA, pre.data = NA, post.data = NA,
  data = list(pre = pre.data, post = post.data), animal.names = FALSE,
  efficacy = 95, confidence = 95, restrict.efficacy = TRUE,
  control.animals = FALSE, paired.model = FALSE, fix.variation = FALSE,
  fix.efficacy = TRUE, zero.inflation = FALSE,
  individual.analysis = FALSE, divide.data = 1, record.chains = FALSE,
  write.file = FALSE, bootstrap.iters = 10000, plot.graph = TRUE,
  skip.mcmc = NULL, stat.method = c("MCMC", "WAAVP", "Bootstrap"),
  custom.model = NA, ...)
```

**Arguments**

<code>name</code>	a name for the analysis (character). Missing by default (function will require it to be input).
<code>pre.data</code>	the pre-treatment data. Either a numeric vector, a matrix with repeated McMaster counts from the same sample (each row) indifferent columns, or an array with different animals in dimension 1, repeat samples from the same animal in dimension 2, repeated McMaster counts from the same sample in dimension 3 (can be of length 1 if only 1 count recorded). If an array, then the <code>use.paired</code> must be TRUE. Noddefault. Ignored if a value is specified for <code>data</code> .
<code>post.data</code>	the post-treatment data, for acceptable formats see <code>pre.data</code> .
<code>data</code>	either a path to a comma delimited csv file, or an existing R object containing the data. Data can be specified in one of the following ways. If a matrix, the first column is pre-treatment data, the second column is the post-treatment data, and the third column (if supplied) indicates control (1) or treatment (0) animal. Alternatively, the first column is animal names <code>id</code> <code>animal.names = TRUE</code> , with columns 2 and 3 making up the pre- and post-treatment data and optionally column 4 the treatment of control status. If the data is specified as a list, then the first element is taken as the pre-treatment data, and the second element is taken as the post-treatment data. These elements of the list can be provided as for <code>pre.data</code> and <code>post.data</code> . If the data is in a comma delimited file, it must be in the format specified for a single matrix. Missing data, or unused elements of non-ragged arrays, may be represented using NA, which will be removed from the data before analysis. This argument is taken from the specified values of <code>pre.data</code> and <code>post.data</code> by default. If a value is specified for <code>data</code> then arguments specified for <code>pre.data</code> and <code>post.data</code> are ignored.
<code>animal.names</code>	either a character vector of names to be used for each animal specified in the data, TRUE or FALSE. If TRUE, then animal names are taken from the first column of the matrix supplied for <code>data</code> (a matrix must be supplied for <code>data</code> if <code>animal.names</code> is TRUE). If FALSE, then animal names are generated automatically. Default FALSE.
<code>efficacy</code>	the target % efficacy of the drug used. Used to calculate the probability of resistance with the MCMC and bootstrap methods. Default 95%.
<code>confidence</code>	the degree of confidence required with which to report the confidence limits for the true FEC reduction.
<code>restrict.efficacy</code>	option to restrict the estimate of efficacy from the MCMC method to values of greater than 0 (i.e. the mean faecal egg count cannot increase after treatment). If TRUE, then the prior for change in mean egg count is uniform between 0 and 1, and if FALSE then the prior for change in mean egg count is uniform between 0 and 10. If using control animals, then <code>restrict.efficacy</code> applies to the change in mean of the treated animals only, allowing the efficacy to be below 0. Default TRUE.
<code>control.animals</code>	indication of which animals are to be used as controls. Should be either a vector of TRUE/FALSE (or 1/0) corresponding to whether each animal is a control (TRUE) or treatment (FALSE), or simply FALSE (the default) in which case all

	animals are assumed to be treated. Ignored if data is specified as a matrix (or csv-file) with 3 columns, in which case the third column should reflect the treatment status. Default FALSE.
paired.model	the FECRT can be run using two compound gamma distributions to describe the variability within and between animals in place of the single gamma distribution combining the two sources of variability. When using two compound gamma distributions for pre-treatment data, the post-treatment data are paired to the pre-treatment data by animal. The advantage of using a single distribution is that the model is more identifiable, and therefore is likely to converge more quickly and return errors less frequently. The advantage of the paired model is that it allows repeat measurements to be incorporated in the model, and non-randomly missing data (ie. protocols that involve post-treatment sampling of only animals with a high pre-treatment count) to be modelled appropriately. The simple model cannot be used when using repeat samples within animal, and will provide inaccurate results if animals are targeted for post-treatment sampling based on their pre-treatment count. Default uses the simple model (FALSE).
zero.inflation	option to use a zero-inflated gamma Poisson in place of the gamma Poisson distribution. If TRUE, zero inflated distributions are used pre- and post-treatment (with the prevalence fixed between pre- and post-treatment distributions). Default FALSE.
divide.data	count division factor to allow egg count data in eggs per gram to be used raw (numeric). Default 1 (no transformation to data).
record.chains	option to allow the MCMC chains to be recorded for future use. If TRUE, the function returns the MCMC object as part of the return value (the MCMC object is not printed using the print method). If write.file==TRUE, the results are also saved using a filename containing the name of the analysis. Default FALSE.
write.file	option to write the results of the analysis to a text file using a filename containing the name of the analysis. The contents of the text file are identical to the return value of the function. Default FALSE.
bootstrap.iters	the number of bootstrap iterations to use for the bootstrap method. Default 10000.
plot.graph	an option to plot the posterior true egg count reduction from the MCMC method graphically. If write.file==TRUE then the graph is saved as a PDF, otherwise the graph is plotted on the active device. Default FALSE.
skip.mcmc	option to omit the MCMC analysis, and return bootstrap and WAAPV method analysis results alone. Default FALSE.
...	other options to be passed directly to <a href="#">autorun.jags</a> .

### Details

Should probably discuss different methods here

### Value

Returns a list of the results obtained using each method, printed using a custom print method.

## References

M. J. Denwood, S. W. J. Reid, S. Love, M. K. Nielsen, L. Matthews, I. J. McKendrick, and G. T. Innocent. Comparison of three alternative methods for analysis of equine Faecal Egg Count Reduction Test data. *Prev. Vet. Med.* (2009), doi:10.1016/j.prevetmed.2009.11.009

## See Also

[reduction\\_analysis](#), [count\\_model](#), [bayescount](#), [autoextend.jags](#), [runjags](#) This function applies a Bayesian [zero-inflated] gamma Poisson (negative binomial) model to faecal egg count reduction test (FECRT) data to return possible values for the mean drug efficacy. Pre-treatment data are assumed to arise from either a gamma-Poisson or zero-inflated gamma-Poisson distribution, with post-treatment data described by a separate gamma-Poisson or zero-inflated gamma-Poisson distribution. The change in mean between these distributions is therefore the mean egg count reduction. If `paired.model=TRUE`, a slightly different formulation is used whereby the observed count pre-treatment is assumed to follow a compound gamma-gamma-Poisson distribution with the variability within and between animals separated. The post treatment mean for each animal is derived from the pre-treatment animal mean and FEC reduction. This formulation allows data with non-random missing post-treatment counts to be analysed correctly, and also allows data with repeat counts from an individual to be analysed - providing a method of increasing the power of the method substantially. Results are also obtained using non-parametric bootstrapping and the method advocated in the 1992 World Association for the Advancement of Veterinary Parasitology (W.A.A.V.P.) methods for the detection of anthelmintic resistance in nematodes of veterinary importance (unless the data contains repeat values or missing values). Confidence intervals for the relevant statistics are printed to file if `write.file = TRUE`, and returned using a custom print method.

Lower level running of JAGS and assessing the simulation for convergence and required run length is done using [autorun.jags](#). Note: The GUI interface for R in Windows may not continually refresh the output window, making it difficult to track the progress of the simulation (if `silent.jags` is `FALSE`). To avoid this, you can run the function from the terminal version of R (located in the Program Files/R/bin/ folder).

\*THIS SOFTWARE IS INTENDED FOR EDUCATIONAL PURPOSES ONLY AND SHOULD NOT BE RELIED UPON FOR REAL WORLD APPLICATIONS\*

## Examples

```
#reduction_model()
```

---

reduction_power	<i>Count reduction (eg FECRT) data power calculations</i>
-----------------	---

---

## Description

Power calculation based on BNB code and Christian approximation.

## Usage

```
reduction_pval(pre_data, post_data, H0_1 = 0.9, H0_2 = 0.95, edt_pre = 1,
  edt_post = 1, prob_priors = c(1, 1), k_change = 1, true_k = NA,
  delta_method = TRUE, beta_iters = 10^6)
```

**See Also**

[count\\_analysis](#), [reduction\\_power](#), [bayescount](#)

---

reduction\_precision     *Count data reduction precision calculations*

---

**Description**

Precision analysis by Monte Carlo simulation and examination of distribution of mean

**Usage**

```
reduction_precision(meanepg = 200, reduction = 95, g.faeces = 3,  
  sensitivity = 1/25, replicates = 1, animals = 10,  
  pre.coeffvarrep = 0.4, pre.coeffvarind = 0.3, pre.coeffvargroup = 0.7,  
  post.coeffvarrep = 0.4, post.coeffvarind = 0.3,  
  post.coeffvargroup = 0.7, true.sample = FALSE, lower.interval = NA,  
  upper.interval = NA, iterations = 1e+05, mc.decimals = 0.95,  
  mc.conf = 0.99, feedback = FALSE, mc.output = FALSE)
```

**See Also**

[reduction\\_analysis](#), [reduction\\_power](#), [bayescount](#)

# Index

## \* **methods**

lnormal.params, 20  
lnormal\_params, 21  
normal.params, 23

## \* **models**

count.analysis, 3  
count.model, 5  
count.power, 7  
count.precision, 9  
fecrt.power, 13  
fecrt.precision, 15  
likelihood, 18  
maximise.likelihood, 22  
reduction\_analysis, 24  
reduction\_model, 27

autoextend.jags, 11, 12, 26, 30  
autorun.jags, 3, 4, 6, 26, 27, 29, 30

bayescount, 2, 3, 4, 11–13, 26, 27, 30, 31  
bayescount-package (bayescount), 2  
bayescount.single, 3  
bayescount.single (count.analysis), 3  
bayescount.single (count\_analysis), 11  
bayescountpackage (bayescount), 2

count.analysis, 3, 6, 11, 19, 22  
count.analysis (count\_analysis), 11  
count.model, 4, 5  
count.model (count\_model), 11  
count.power, 7, 11  
count.power (count\_power), 12  
count.precision, 9  
count.precision (count\_precision), 12  
count\_analysis, 11, 12, 13, 26, 31  
count\_model, 11, 11, 30  
count\_power, 12, 13  
count\_precision, 12

FEC.analysis (count.analysis), 3

FEC.analysis (count\_analysis), 11  
fec.analysis, 15  
fec.analysis (count.analysis), 3  
fec.analysis (count\_analysis), 11  
FEC.model (count.model), 5  
FEC.model (count\_model), 11  
fec.model (count.model), 5  
fec.model (count\_model), 11  
FEC.power (count.power), 7  
FEC.power (count\_power), 12  
fec.power, 3, 9  
fec.power (count.power), 7  
fec.power (count\_power), 12  
FEC.power.limits (count.precision), 9  
FEC.power.limits (count\_precision), 12  
fec.power.limits (count.precision), 9  
fec.power.limits (count\_precision), 12  
FEC.precision (count.precision), 9  
FEC.precision (count\_precision), 12  
fec.precision (count.precision), 9  
fec.precision (count\_precision), 12  
FECRT (reduction\_analysis), 24  
fecrt, 3  
fecrt, (reduction\_analysis), 24  
fecrt.analysis, 9, 18  
fecrt.analysis (reduction\_analysis), 24  
FECRT.analysis, (reduction\_analysis), 24  
FECRT.model (reduction\_model), 27  
fecrt.model (reduction\_model), 27  
FECRT.power (fecrt.power), 13  
FECRT.power (reduction\_power), 30  
fecrt.power, 3, 13, 18, 27  
fecrt.power (reduction\_power), 30  
FECRT.power.limits (fecrt.precision), 15  
FECRT.power.limits  
    (reduction\_precision), 31  
fecrt.power.limits, 27  
fecrt.power.limits (fecrt.precision), 15  
fecrt.power.limits



- (reduction\_precision), 31
- FECRT.precision (fecrt.precision), 15
- FECRT.precision (reduction\_precision), 31
- fecrt.precision, 15, 15
- fecrt.precision (reduction\_precision), 31
  
- HPDinterval, 4, 19
  
- likelihood, 18
- lnormal.params, 20, 23
- lnormal.params (lnormal\_params), 21
- lnormal\_params, 21
  
- maximise.likelihood, 22
  
- normal.params, 20, 23
- normal.params (lnormal\_params), 21
- normal\_params (lnormal\_params), 21
  
- reduction.analysis
  - (reduction\_analysis), 24
- reduction.model (reduction\_model), 27
- reduction.power (reduction\_power), 30
- reduction.precision
  - (reduction\_precision), 31
- reduction\_analysis, 11, 24, 30, 31
- reduction\_model, 12, 26, 27
- reduction\_power, 12, 30, 31
- reduction\_precision, 31
- reduction\_pval (reduction\_power), 30
- run.model (count.model), 5
- run.model (count\_model), 11
- runjags, 3, 11, 12, 26, 30