Package ‘NormalGamma’

February 19, 2015

Type Package

Title Normal-gamma convolution model

Version 1.1

Date 2013-09-20

Author S. Plancade and Y. Rozenholc

Maintainer Sandra Plancade <Sandra.Plancade@jouy.inra.fr>

Description The functions proposed in this package compute the density of the sum of a Gaussian and a gamma random variables, estimate the parameters and correct the noise effect in a gamma-signal and Gaussian-noise model. This package has been used to implement the background correction method for Illumina microarray data presented in Plancade S., Rozenholc Y. and Lund E. ``Generalization of the normal-exponential model : exploration of a more accurate parameterization for the signal distribution on Illumina BeadArrays", BMC Bioinfo 2012, 13(329).

License GPL-2

Depends optimx , histogram

NeedsCompilation no

Repository CRAN

Date/Publication 2013-10-16 16:46:36

R topics documented:

NormalGamma-package *Background correction using the normal+gamma model*

Description

The NormalGamma package is devoted to the density convolution model Y = X + N where X and N are independent variables with respectively gamma and normal distributions. For this model called further normal-gamma convolution model, this package proposes the computation of the convoluted density, the parameter estimation by likelihood maximization and the background correction obtained by deconvolution.

This package has been used to implement the background correction method for Illumina microarray data presented in Plancade S., Rozenholc Y. and Lund E., BMC Bioinfo 2012.

Details

Package:	NormalGamma
Type:	Package
Version:	001.0
Date:	2011-12-20
License:	GPL-3

The NormalGamma package includes three functions and a data set:

dnormgam computes the convolution product of a normal and a gamma densities.

normgam.fit computes the Maximum Likelihood Estimator of the parameters.

normgam.signal performs the normal-gamma background correction.

Intensities provides data from one Illumina microarray.

Author(s)

Plancade S. and Rozenholc Y.

Maintainer: Plancade S. <Sandra.Plancade@jouy.inra.fr>

References

Plancade S., Rozenholc Y. and Lund E. "Generalization of the normal-exponential model : exploration of a more accurate parametrisation for the signal distribution on Illumina BeadArrays", BMC Bioinfo 2012, 13(329).

Examples

```
# Load Illumina microarray data previously transformed into two sets of intensities
# corresponding to negative and regular probes.
```

```
## Not run:

data(RegNegIntensities_Example)

X = Intensities$Regular
N = Intensities$Negative

## Compute the maximum likelihood estimator:
parmle = normgam.fit(X, N)$par

par(mfrow=c(2,1)) # split windows for post comparison

## plot the histogram and the estimated density:
F = dnormgam(parmle, plot=FALSE)


H = histogram(X, type='irregular', verbose=FALSE, plot=FALSE)
plot(H, xlim=c(0,500))
lines(F$xout, F$dout, col='red')

# Perform the normal-gamma background correction:
Shat = normgam.signal(X,parmle)
H = histogram(Shat, type='irregular', verbose=FALSE, plot=FALSE)
plot(H, xlim=c(0,30))


## End(Not run)
```

`dnormgam` *Normal-gamma density*

Description

Computes the convolution product of a normal and a gamma densities.

Usage

```
dnormgam(par, x = NULL, N0 = 65536, plot = TRUE,
                    log = FALSE, tail.cor = TRUE, cor = 1e-15,
                    mu = par[1], sigma = par[2], k = par[3], theta = par[4])
```

Arguments

`par` vector of parameters; (`par[1],par[2]`) are the mean and standard deviation of the normal distribution and (`par[3],par[4]`) are the shape and scale parameters of the gamma distribution.

x vector of values where the density is computed; if x == NULL, the density is computed on a sequence of values from 0 to par[1]+5*par[2]+q where q is the quantile of probability 0.99999 of the gamma distribution.

N0 number of equally spaced values in the Fast Fourier Transform (see details).

plot logical; if TRUE plot of the density.

log logical; if TRUE density d is given as log(d).

tail.cor logical; if TRUE a linear approximation of right tail adjusts numerical instability.

cor limit of right tail correction; if tail.cor == TRUE, linear approximation is applied to values with density estimate smaller than cor.

mu, sigma alternative definition of mean and standard deviation of the normal distribution.

k, theta alternative definition of shape and scale parameters of the gamma distribution.

Details

The convoluted density is computed using the fft function (Fast Fourier Transform). See details in Plancade S., Rozenholc Y. and Lund E., BMC Bioinfo 2012.

Only one definition of the parameters is required, either par or (mu, sigma, k, theta). If both are specified and do not match, an error message is returned.

Value

xout vector of values where normal-gamma density is computed; equal to x when x is not NULL.

dout vector of values of normal-gamma density.

Author(s)

Plancade S. and Rozenholc Y.

References

Plancade S., Rozenholc Y. and Lund E. "Generalization of the normal-exponential model : exploration of a more accurate parametrisation for the signal distribution on Illumina BeadArrays", BMC Bioinfo 2012, 13(329).

See Also

normgam.fit computes the Maximum Likelihood Estimator and normgam.signal implements the background correction using the normal-gamma model.

Examples

```
## Example 1

par = c(-10, 5, 2, 20)
F = dnormgam(par)
```

```
## Example 2

n = 50000
par = c(60,5,0.15,400)
F = dnormgam(par, plot=FALSE)

X = rnorm(n, mean=par[1], sd=par[2]) + rgamma(n, shape=par[3], scale=par[4])
H = histogram(X, type='irregular', verbose=FALSE, plot=FALSE)
plot(H, xlim=c(0,500))
lines(F$xout, F$dout, col='red')
```

`Intensities` *Regular and Negative probes from one Illumina microarray*

Description

Provides probe intensities measured on Illumina Human HT-6 v4 Expression BeadChip, from one control in the NOWAC cohort (for more details, see Plancade S., Rozenholc Y. and Lund E. BMC Bioinfo 2012). Vectors `Regular` and `Negative` contain the intensities of regular and negative probes, respectively.

Usage

```
data(RegNegIntensities_Example)
```

Format

A list `Intensities` with two elements: a vector `Regular` of length 25 519 and a vector `Negative` of length 759.

References

Plancade S., Rozenholc Y. and Lund E. "Generalization of the normal-exponential model : exploration of a more accurate parametrisation for the signal distribution on Illumina BeadArrays", BMC Bioinfo 2012, 13(329).

Lund, E., Dumeaux, V., Braaten T., Hjartaker A., Engeset D., Skeie G., and Kumle M. (2008), Cohort profile: the norwegian women and cancer -nowac- kvinner og kreft, *Int. J. Epidemiol.*, **37**(1), 36–41.

Examples

```
data(RegNegIntensities_Example)
X = Intensities$Regular
N = Intensities$Negative

par(mfrow=c(2,1))
```

```
HX = histogram (X, type='irregular', verbose=FALSE, plot=FALSE)
plot(HX, xlim=c(0,500))
hist(N, 30)
```

normgam.fit *Normal-gamma Maximum Likelihood Estimator*

Description

Computes the Maximum Likelihood Estimator for the normal-gamma distribution, either from a normal-gamma distributed sample or from two samples respectively normal-gamma and normally distributed.

Usage

```
normgam.fit(X, N = NULL, par.init = NULL, lower = NULL, upper = NULL,
            control = NULL, verbose = FALSE)
```

Arguments

`X`	vector of normal-gamma distributed values.
`N`	vector of normal distributed values.
`par.init`	vector of initial values for parameters (optional). `par.init[1]` and `par.init[2]` are the mean and standard deviation of the normal distribution, and `par.init[3]` and `par.init[4]` are the shape and scale parameters of the gamma distribution. See details for default initial values.
`lower, upper`	Bounds on the variables for maximization (optional).
`control`	list of control parameters (see details).
`verbose`	logical; if `TRUE` initial values of the parameters are printed.

Details

Likelihood maximization is run by the `R` function `optimx`.

By default, maximization is run with the following control parameters: the maximum number of iterations is equal to `1000` and the vector of scaling values for the parameters is `(par0[1], par0[2], par0[3]*par0[4], sqrt(par0[3])*par0[4])/10` where `par0` is the vector of default initial parameters. In case of unsuccessful convergence, maximization is run with `optimx` default control parameters. A list of control parameters can also be chosen by the user (see `optimx`).

If `par.init == NULL`, the initial parameters are computed in two ways depending if `N` is `NULL` or not. If `N != NULL`, the initial parameters are computed following the method of the moments (see

Plancade S., Rozenholc Y. and Lund E., BMC Bioinfo 2012). If `N == NULL`, the initial parameters `(par0[1], par0[2])` of the normal distribution are computed following the RMA procedure of Xie Y., Wang X. and Story M. (2009) for the normal-exponential convolution model, and the initial parameters of the gamma distribution, computed following the method of the moments, are `(par0[4]=(sd(X)^2-par0[2])/(mean(X)-par0[1]), par0[3]=(mean(X)-par0[1]/par0[4])`. Note that the RMA procedure for initial parameter computation when `N == NULL` stems from an heuristic adapted to microarray data. For parameters with different magnitude, user should specify initial parameters.

Value

`par` vector of estimated parameters; `par[1]` and `par[2]` are the mean and standard deviation of the normal distribution and `par[3]` and `par[4]` are the shape and scale parameters of the gamma distribution

`lik` value of the normal-gamma log-likelihood corresponding to `par`.

`conv` integer code: `0` indicates successful convergence. This parameter has the value of the output parameter `conv` from the procedure `optimx` used for likelihood maximization (see `optimx` for details).

Author(s)

Plancade S. and Rozenholc Y.

References

Plancade S., Rozenholc Y. and Lund E. "Generalization of the normal-exponential model : exploration of a more accurate parametrisation for the signal distribution on Illumina BeadArrays", BMC Bioinfo 2012, 13(329).

Xie Y. Wang X. and Story M. (2009), Statistical methods of background correction for Illumina BeadArray data, *Bioinformatics*, **25**(6), 751–757.

See Also

`dnormgam` computes the density of the normal-gamma distribution and `normgam.signal` implements the background correction using the normal-gamma model. `Intensities` provides an example of Illumina microarray data.

Examples

```
# Example 1: simulated data

## Not run:

n = 1000
par = c(60,5,0.15,400)
X = rnorm(n, mean=par[1], sd=par[2]) + rgamma(n, shape=par[3], scale=par[4])
N = rnorm(100, mean=par[1], sd=par[2])

par1 = normgam.fit(X, N)$par
```

```
par2 = normgam.fit(X)$par


F1 = dnormgam(par1, plot=FALSE)
F2 = dnormgam(par2, plot=FALSE)

par(mfrow=c(2,1))

H = histogram(X, type='irregular', verbose=FALSE, plot=FALSE)

plot(H, xlim=c(0,500))
lines(F1$xout, F1$dout,col='red')

plot(H, xlim=c(0,500))
lines(F2$xout, F2$dout,col='blue')

## End(Not run)

# Example 2: Illumina data

## Not run:

data(RegNegIntensities_Example)

X = Intensities$Regular
N = Intensities$Negative

par1 = normgam.fit(X, N)$par
par2 = normgam.fit(X)$par

F1 = dnormgam(par1, plot=FALSE)
F2 = dnormgam(par2, plot=FALSE)

par(mfrow=c(2,1))

H = histogram(X, type='irregular', verbose=FALSE, plot=FALSE)

plot(H, xlim=c(0,500))
lines(F1$xout, F1$dout, col='red')

plot(H, xlim=c(0,500))
lines(F2$xout, F2$dout, col='blue')

## End(Not run)
```

normgam.signal *Normal-gamma background correction*

Description

Performs background correction using the normal-gamma model.

Usage

```
normgam.signal(x, par, tail.cor = TRUE, cor = 1e-15, gshift = FALSE,
               mu = par[1], sigma = par[2], k = par[3], theta = par[4])
```

Arguments

`x` vector of observed intensities.

`par` vector of parameters; `par[1]` and `par[2]` are the mean and standard deviation of the normal distribution and `par[3]` and `par[4]` are the shape and scale parameters of the gamma distribution.

`tail.cor` logical (see details).

`cor` limit of the right tail correction (see details).

`gshift` logical; if `TRUE` and `par[3]` is smaller than 1, an ad-hoc translation and a thresholding to 0 are applied to background-corrected values so that the mode of corrected value distribution is 0.

`mu, sigma` alternative definition of mean and standard deviation of the normal distribution.

`k, theta` alternative definition of shape and scale parameters of the gamma distribution.

Details

`normgam.signal` performs background correction in an additive background noise+signal model with a normal background noise and a gamma-distributed signal. The corrected value from an observed intensity `x` is the expectation of the signal given the signal and noise distributions. For a set of parameters `(mu, sigma, k, theta)`, it is given by the ratio of the convolution product of `dgamma(x, shape=k+1, scale=theta)` and `dnorm(x, mean=mu, sd=sigma)` and the convolution product of `dgamma(x, shape=k, scale=theta)` and `dnorm(x, mean=mu, sd=sigma)`. For more details see Plancade S., Rozenholc Y. and Lund E., BMC Bioinfo 2012.

If `tail.cor = TRUE`, a linear approximation of right tail is applied to values with density estimate smaller than `cor` in the computation of normal-gamma convoluted densities (see `dnormgam`).

Only one definition of the parameters is required, either `par` or `(mu, sigma, k, theta)`. If both are specified and do not match, an error message is returned.

Value

Vector of background noise-corrected intensities.

Author(s)

Plancade S. and Rozenholc Y.

References

Plancade S., Rozenholc Y. and Lund E. "Generalization of the normal-exponential model : exploration of a more accurate parametrisation for the signal distribution on Illumina BeadArrays", BMC Bioinfo 2012, 13(329).

See Also

`dnormgam` computes the density of the normal-gamma distribution and `normgam.fit` computes the Maximum Likelihood Estimator. `Intensities` provides an example of Illumina microarray data.

Examples

```
#Example 1: simulated data

n = 50000
par = c(60,5,0.15,400)
S = rgamma(n, shape=par[3], scale=par[4])
B = rnorm(n, mean=par[1], sd=par[2])
X = S + B

par(mfrow=c(2,1))

Shat1 = normgam.signal(X, par)
H1 = histogram(Shat1, type='irregular', verbose=FALSE, plot=FALSE)
plot(H1, xlim=c(0,50))
I = seq(from=0, to=50, length=1000)
lines(I, dgamma(I, shape=0.15, scale=400), col='red')

Shat2 = normgam.signal(X, par, gshift = TRUE)
H2 = hist(Shat2, 10000, plot=FALSE)
plot(H2, xlim=c(0,50), freq=FALSE)
lines(I, dgamma(I, shape=0.15, scale=400), col='red')


#Example 2: Illumina data

## Not run:

data(RegNegIntensities_Example)

X = Intensities$Regular
N = Intensities$Negative

# parameter estimation
parmle = normgam.fit(X, N)$par

Shat = normgam.signal(X,parmle)
H = histogram(Shat, type='irregular', verbose=FALSE, plot=FALSE)
plot(H, xlim=c(0,30))

## End(Not run)
```

Index