Package ‘NormalGamma’

February 15, 2013

Type  Package

Title  Normal-gamma convolution model

Version  0.1.0

Date  2011-12-20

Author  S. Plancade and Y. Rozenholc

Maintainer  Sandra Plancade <sandra.c.plancade@uit.no>

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

License  GPL-2

Depends  optimx, histogram

Repository  CRAN

Date/Publication  2011-12-30 11:22:36

NeedsCompilation  no

R topics documented:

NormalGamma-package ................................................. 2
dnormgam .......................................................... 3
Intensities ............................................................. 5
normgam.fit .......................................................... 6
normgam.signal ......................................................... 8

Index  11
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. (http://arxiv.org/abs/1112.4180).

Details

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.c.plancade@uit.no>

References


Examples

```R
# Load Illumina microarray data previously transformed into two sets of intensities corresponding
# to negative and regular probes.
## Not run:

# # 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(0,5))
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(0,3))

## 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.
dnormgam

x vector of values where the density is computed; if \( x = \) NULL, the density is computed on a sequence of values from 0 to \( \text{par}[1]+5*\text{par}[2]+q \) 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 \texttt{fft} function (Fast Fourier Transform). See details in Plancade S., Rozenholc Y. and Lund E. (http://arxiv.org/abs/1112.4180).

Only one definition of the parameters is required, either \( \text{par} \) or \( (\text{mu}, \text{sigma}, \text{k}, \text{theta}) \). 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


See Also

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

Examples

```r
## Example 1
par = c(-10, 5, 2, 20)
F = dnormgam(par)
```

```r
## 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(0,50000))
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. (http://arxiv.org/abs/1112.4180), Section 2). 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 25519 and a vector Negative of length 759.

References

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(0,50000))
normgam.fit

$\text{normgam.fit}$

**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**

```r
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 Planckade S., Rozenholc Y. and Lund E. (http://arxiv.org/abs/1112.4180)). 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
normgam.fit  

the initial parameters of the gamma distribution, computed following the method of the moments, are \( (\text{par0}[4] = \frac{\text{sd}(X)^2 - \text{par0}[2]}{\text{mean}(X) - \text{par0}[1]}), \text{par0}[3] = \frac{\text{mean}(X) - \text{par0}[1]}{\text{par0}[4]} \). Note that the RMA procedure for initial parameter computation when \( N = \text{NULL} \) stems from an heuristic adapted to microarray data. For parameters with different magnitude, user should specify initial parameters.

**Value**

- **par** vector of estimated parameters; \( \text{par}[1] \) and \( \text{par}[2] \) are the mean and standard deviation of the normal distribution and \( \text{par}[3] \) and \( \text{par}[4] \) are the shape and scale parameters of the gamma distribution.
- **lik** value of the normal-gamma log-likelihood corresponding to \( \text{par} \).
- **conv** integer code: 0 indicates successful convergence. This parameter has the value of the output parameter \( \text{conv} \) from the procedure \texttt{optimx} used for likelihood maximization (see \texttt{optimx} for details).

**Author(s)**

Plancade S. and Rozenholc Y.

**References**


**See Also**

\texttt{dnormgam} computes the density of the normal-gamma distribution and \texttt{normgam.signal} implements the background correction using the normal-gamma model. \texttt{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(0,5))
lines(F1$xout, F1$dout, col='red')

plot(H, xlim=c(0,5))
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(0,5))
lines(F1$xout, F1$dout, col='red')

plot(H, xlim=c(0,5))
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. (http://arxiv.org/abs/1112.4180).

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

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, 5))
I = seq(from=0, to=5, 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, 5), 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, 10))

## End(Not run)
Index

*Topic **background correction**
  normgam.signal, 8

*Topic **convolution**
  dnormgam, 3
  normgam.signal, 8

*Topic **datasets**
  Intensities, 5

*Topic **density**
  dnormgam, 3

*Topic **gamma**
  dnormgam, 3
  normgam.fit, 6
  normgam.signal, 8

*Topic **maximum likelihood**
  normgam.fit, 6

*Topic **normal**
  dnormgam, 3
  normgam.fit, 6
  normgam.signal, 8

*Topic **package**
  NormalGamma-package, 2

  dgamma, 9
  dnorm, 9
  dnormgam, 2, 3, 7, 9, 10

  Intensities, 2, 5, 7, 10

  NormalGamma (NormalGamma-package), 2
  NormalGamma-package, 2
  normgam.fit, 2, 4, 6, 10
  normgam.signal, 2, 4, 7, 8

  optimx, 6, 7