

Package ‘BMDE’

October 14, 2010

Type Package

Title a Bayesian hierarchical model for the inference on the differential gene expression based on RNA-Seq data

Version 1.0

Date 2010-10-08

Author Juhee Lee, Yuan Ji, Shoudan Liang, Guoshuai Cai, Peter Muller

Maintainer Juhee Lee <jlee14@mdanderson.org>

Description Implement a Bayesian hierarchical model to develop statistical inference for differential gene expression on the basis of RNA-Seq data. The model is proposed by Lee, Ji, Liang, Cai, and Muller (2010).

License GPL(>=2)

LazyLoad yes

R topics documented:

BMDE	1
Index	6

BMDE	<i>Bayesian hierarchical model for differential gene expression on the basis of RNA-Seq data</i>
------	--

Description

This function generates posterior samples for a Bayesian hierarchical model for the inference on the differential gene expression based on RNA-Seq data.

Usage

```
BMDE(Gene, n.gene, n.loci, MCMC, cur.sample, hyper.para)
```

Usage

```
BM_DE(Gene, n.gene, n.loci, MCMC, cur.sample,
      hyper.para)
```

Arguments

Gene	a numeric list of n and N where $N=n+m$, where $N > 0$ at any genomic position. e.g., <code>Gene\$n[[1]]</code> a sequence of read counts along genomic locations under condition 0 for gene 1, and <code>Gene\$N[[1]]</code> a sequence of sum of read counts along genomic locations under the two conditions for gene 1.
n.gene	the total number of genes
n.loci	a numeric sequence of the number of genomic positions
MCMC	a list of two arguments, <code>n.MCMC.iter</code> =the total number of MCMC samples including burn-in samples, <code>burn.in</code> =the number of samples as burn-in
cur.sample	a numeric list of initial values of parameters, <code>xi</code> , <code>xi.bar</code> , <code>w.i</code> , <code>pi.w</code> , <code>pi.lambda</code> , <code>lambda</code> , <code>s2.xi</code> , <code>delta1</code> , <code>delta2</code>
hyper.para	a numeric list of hyper-parameter values, (a.w, b.w) for <code>pi.w</code> , (k.omega, theta.omega) for <code>1/s2.xi</code> , (k1, theta1) for <code>delta1</code> , (k2, theta2) for <code>delta2</code> , (a.lambda, b.lambda, c.lambda) for <code>pi.lambda</code> , eta

Details

This generic function fits a Bayesian hierarchical model for inference on gene differential expression based on RNA Seq data (Lee, Ji, Liang, Cai, and Muller, 2010). For the detailed description, see Lee, Ji, Liang, Cai, and Muller (2010).

Value

The function returns MCMC samples:

w	MCMC samples of <code>w_ij</code> as a matrix of $(\text{sum}(n.\text{loci}) * (n.\text{MCMC.iter} - \text{burn.in}))$
pi.w	MCMC samples of <code>pi_w</code> as a matrix of $(n.\text{gene} * (n.\text{MCMC.iter} - \text{burn.in}))$
xi	MCMC samples of <code>xi</code> as a matrix of $(n.\text{gene} * (n.\text{MCMC.iter} - \text{burn.in}))$
lambda	MCMC samples of <code>lambda</code> as a matrix of $(n.\text{gene} * (n.\text{MCMC.iter} - \text{burn.in}))$
delta1	MCMC samples of <code>delta1</code> as a sequence of length $(n.\text{MCMC.iter} - \text{burn.in})$
delta2	MCMC samples of <code>delta2</code> as a sequence of length $(n.\text{MCMC.iter} - \text{burn.in})$
s2.xi	MCMC samples of <code>s2.xi</code> as a sequence of length $(n.\text{MCMC.iter} - \text{burn.in})$
xi.bar	MCMC samples of <code>xi.bar</code> as a sequence of length $(n.\text{MCMC.iter} - \text{burn.in})$
pi.lambda	MCMC samples of <code>s2.xi</code> as a matrix of $(3 * (n.\text{MCMC.iter} - \text{burn.in}))$

Author(s)

Juhee Lee, Yuan Ji, Shoudan Liang, Guoshuai Cai, Peter Muller Maintainer: Juhee Lee <jlee14@mdanderson.org>

References

Lee, Ji, Liang, Cai, and Muller (2010), On Differential Gene Expression Using RNA-Seq Data.

Examples

```

#Illustrate a simulation study to demonstrate Bayes.DiffGene.RAN.Seq
library(BMDE)
set.seed(43268)

#Illustrate the usage of Bayes.DiffGene.RAN.Seq with a simulation
N.gene <- 100 #number of genes in the simulation
N.loci <- rep(300, N.gene) #number of positions for each gene

#SET THE PARAMETER VALUES TO SIMULATE THE DATE
true.parameter <- NULL
true.parameter$delta1 <- true.parameter$delta2 <- 1
true.parameter$pi.lambda <- c(0.95, 0.025, 0.025)
true.parameter$pi.w <- c(0.95, 0.05)
true.parameter$mean.eta <- 3
true.parameter$s2.eta <- 0.25^2
true.parameter$s2.xi <- 0.1
true.parameter$xi.bar <- 0

#SIMULATE THE RNA-SEQ DATA
Gene <- NULL
Gene$N <- NULL; Gene$n <- NULL
for(i in 1:N.gene)
{
  lambda <- sample(c(0, -1, 1), 1, replace=FALSE, true.parameter$pi.lambda)
  if(lambda==-1) delta <- true.parameter$delta1
  if(lambda==1) delta <- true.parameter$delta2
  if(lambda==0) delta <- 0

  xi <- rnorm(1, mean=(true.parameter$xi.bar+lambda*delta), sd=sqrt(true.parameter$s2.xi))
  eta <- rnorm(1, mean=true.parameter$mean.eta, sd=sqrt(true.parameter$s2.eta))

  alpha <- exp(eta)*exp(xi)/(1+exp(xi))
  beta <- exp(eta)/(1+exp(xi))

  w <- sample(c(1, 0), N.loci[i], replace=TRUE, true.parameter$pi.w)

  p <- rbeta(N.loci[i], alpha, beta)
  p[w==0] <- rbeta(sum(w==0), 1/2, 1/2)

  Gene$N[[i]] <- round(exp(rnorm(N.loci[i], 5, sd=1)))
  Gene$n[[i]] <- rbinom(N.loci[i], Gene$N[[i]], p)
}

#FINDING THE INITIAL VALUES AND THE HYPER-PARAMETERS.
hyper.parameter <- NULL
initial.sample <- NULL
hyper.parameter$eta <- rep(NA, N.gene)
initial.sample$xi <- rep(NA, N.gene)
r.i <- rep(NA, N.gene)
for(i in 1:N.gene)
{
  N <- Gene$N[[i]]
  n <- Gene$n[[i]]

```

```

#INITIALIZE ETA AND XI
p.v <- var(n/N)
r.i[i] <- p.bar <- mean(n)/mean(N)

a <- p.bar*(p.bar*(1-p.bar)/p.v -1)
b <- (1-p.bar)*(p.bar*(1-p.bar)/p.v -1)

gam <- a+b
mu <- a/(a+b)

hyper.parameter$eta[i] <- log(gam)
initial.sample$xi[i] <- log(mu/(1-mu))
}

#HYPER-PARAMETERS
#PRIOR \PI_W_I
hyper.parameter$a.w <- 19
hyper.parameter$b.w <- 1

#PRIOR FOR S2.XI
#MEAN=VAR(LOGITS BETWEEN 2.5% AND 97.%)
#PRIOR VAR=AS BIG AS POSSIBLE (SO K==3, THEN ADJUST THETA TO MAKE THE MEAN THE ABOVE VALUE)
logit.upper <- quantile(initial.sample$xi, 0.975)
logit.lower <- quantile(initial.sample$xi, 0.025)
temp.logit <- initial.sample$xi[(initial.sample$xi < logit.upper)&(initial.sample$xi > logit.lower)]
hyper.parameter$k.omega <- 3
hyper.parameter$theta.omega <- var(temp.logit)*(hyper.parameter$k.omega-1)
#omega=1/s^2_xi THEN omega ~ GAMMA(k.omega, theta.omega) where theta.omega is the rate
#(not the scale, but theta is the scale for delta)
#E(s2.xi)=var(log(prob.i/(1-prob.i)))

#PRIOR FOR DELTA 1 (UNDER EXPRESSED)
temp.mean <- (quantile(initial.sample$xi, 0.5)-quantile(initial.sample$xi, 0.025)) #PRIOR MEAN
hyper.parameter$k1 <- 5
hyper.parameter$theta1[1] <- temp.mean/hyper.parameter$k1

#PRIOR FOR DELTA 2 (OVER EXPRESSED)
temp.mean <- (quantile(initial.sample$xi, 0.975)-quantile(initial.sample$xi, 0.5)) #PRIOR MEAN
hyper.parameter$k2 <- 5
hyper.parameter$theta2[1] <- temp.mean/hyper.parameter$k2

#PRIOR FOR PI_LAMBDA
hyper.parameter$a.lambda <- 38
hyper.parameter$b.lambda <- 1
hyper.parameter$c.lambda <- 1

#initialization
#PI_W SAMPLE FROM PRIOR
initial.sample$pi.w <- rbeta(N.gene, hyper.parameter$a.w, hyper.parameter$b.w)

#W_IJ
t.w <- NULL
for(i in 1:N.gene)
{
temp.w <- ifelse(runif(N.loci[i]) < initial.sample$pi.w[i], 1, 0)
t.w <- c(t.w, temp.w)
}

```

```

}
initial.sample$w <- t.w

#XI_BAR
initial.sample$xi.bar <- mean(initial.sample$xi)

#PI_LAMBDA
initial.sample$pi.lambda <- rep(NA, 3)
initial.sample$pi.lambda[1] <- rgamma(1, hyper.parameter$a.lambda)
initial.sample$pi.lambda[2] <- rgamma(1, hyper.parameter$b.lambda)
initial.sample$pi.lambda[3] <- rgamma(1, hyper.parameter$c.lambda)
initial.sample$pi.lambda <- initial.sample$pi.lambda/sum(initial.sample$pi.lambda)

#S^2_XI
initial.sample$s2.xi <- var(log(r.i/(1-r.i)))

#LAMBDA
initial.sample$lambda <- sample(c(0, -1, 1), N.gene, replace=TRUE, prob=initial.sample$pi)

#DELTA1 AND DELTA2
initial.sample$delta1 <- hyper.parameter$k1*hyper.parameter$theta1
initial.sample$delta2 <- hyper.parameter$k2*hyper.parameter$theta2

#MCMC parameters
MCMC.para <- NULL
MCMC.para$n.MCMC.iter <- 100
MCMC.para$burn.in <- 10

DiffGene.Results <- BMDE(Gene, N.gene, N.loci, MCMC.para, initial.sample, hyper.parameter)
#POSTERIOR PROB OF DIFF EXPRESSION
apply(abs(DiffGene.Results$lambda), 1, mean)
#POSTERIOR MEAN OF XI
apply(DiffGene.Results$xi, 1, mean)

```

Index

*Topic **Bayes, Differential Gene
Expression, RNA-Seq**

BMDE, [1](#)

BMDE, [1](#)