Package 'ebGenotyping'

April 13, 2016

Type Package

Title Genotyping and SNP Detection using Next Generation Sequencing Data

Version 2.0.1

Date 2016-04-07

Author Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

Maintainer Gongyi Huang<53hgy@163.com>

Description Genotyping the population using next generation sequencing data is essentially important for the rare variant detection. In order to distinguish the genomic structural variation from sequencing error, we propose a statistical model which involves the genotype effect through a latent variable to depict the distribution of non-reference allele frequency data among different samples and different genome loci, while decomposing the sequencing error into sample effect and positional effect. An ECM algorithm is implemented to estimate the model parameters, and then the genotypes and SNPs are inferred based on the empirical Bayes method.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-13 09:28:12

R topics documented:

ebGenotyping-package *Genotyping and SNP Detection using Next Generation Sequencing Data*

Description

Genotyping the population using next generation sequencing data is essentially important for the rare variant detection. In order to distinguish the genomic structural variation from sequencing error, we propose a statistical model which involves the genotype effect through a latent variable to depict the distribution of non-reference allele frequency data among different samples and different genome loci, while decomposing the sequencing error into sample effect and positional effect. An ECM algorithm is implemented to estimate the model parameters, and then the genotypes and SNPs are inferred based on the empirical Bayes method.

Details

Package:	ebGenotyping
Type:	Package
Version:	2.0.1
Date:	2016-04-07
License:	GPL-2

The most important function is ecm, which is used to establish the model described in 'An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data' to do genotyping and SNP detection using NGS data.

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

ecm *Genotyping and SNP Detection using Next Generation Sequencing Data*

Description

This function implements the method described in 'An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data'.

Usage

```
ecm(dat,cvg,eps=1e-6,max.steps=500,eps.bisec=1e-6,ini.m=-7,ini.d=-7)
```

Arguments

dat	a n*m matrix: the ith row, jth column of the matrix represents the non-reference counts of ith sample at jth position.
cvg	a n*m matrix: the ith row, jth column of the matrix represents the depth of ith sample at jth position.
eps	a single value: a threshold to control the convergence criterion. The default is 1e-06.
max.steps	a single value: the maximum steps to run iterative algorithm to estimate parameters. The default is 500.(Adjustment is needed according to the number of parameters to estimate and the initial value of them.)
eps.bisec	a single value: a threshold to control the convergence criterion of bisection criterion. The default is 1e-06.
ini.m	the initial value of each element of mu. We suggest users to use default -7.
ini.d	the initial value of each element of delta. We suggest users to use default -7.

Details

This function implements the method described in 'An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data'. According to the paper, users can do genotyping with the estimated genotypes("geno.est"), and do SNP detection with the posterior probabilities of RR("post.probs$zRR"), based on the non-reference counts("dat")and depth("cvg").

Value

par.est	a list including the estimate of position effect(mu), sample effect(delta), and the probability of RR and RV.
post.probs	3 matrix: the estimate of the posterior probabilities of 3 genotypes for n samples at m positions.
steps	the total steps to run iterative algorithm.
geno.est	a n*m matrix: the estimated genotypes(0 for RR, 1 for RV and 2 for VV) of n samples at m positions.

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

Examples

```
#---------generate simulation data-----------
#start:generate simulation data#
set.seed(2016)
m <- 100
m0 <- m*0.95
m1 <- m-m0
n <- 30
Q <- 0.8
z <- cbind(matrix(0,n,m0),matrix(rbinom(n*m1,1,Q),n,m1))
b <- which(z==1)
R <- 0.8 # proportion of homozygous SNP
w <- rbinom(length(which(z==1)),1,R)
# z are genotypes
z[b[which(w==0)]] <- 1
z[b[which(w==1)]] <- 2
mu <- rep(-3,m)# stands for no effect
delta <- rep(-3,n)# stands for no effect
er.p <- -abs(outer(delta,mu,"+"))
p <- rlogit(er.p)
p[which(z==1)] <- 1/2
p[which(z==2)] <- 1-p[which(z==2)]
cvg <- matrix(rbinom(m*n,50,0.5),n,m)
dat <- matrix(sapply(1:(m*n),function(i) rbinom(1,cvg[i],p[i])),n,m)
#end:generate simulation data-#
#-----genotyping and SNP detection----------------
res <- ecm(dat=dat,cvg=cvg)
mean(z!=res$geno.est)#genotyping error
#----------call SNP---------
#start:call snp#
# define a function to calculate power, typeI error and FDR.
cutsnp <- function(fdr,alpha,true){
wh <- (true!=0)
tp <- sum((wh)&(fdr<alpha));
tn <- sum((!wh)&(fdr>=alpha));
fp <- sum((!wh)&(fdr<alpha));
fn <- sum((wh)&(fdr>=alpha));
pw  <- tp/(tp+fn);
t1 <- fp/(fp+tn);
fdr <- fp/(fp+tp);
return(c(TP=tp,TN=tn,FP=fp,FN=fn,power=pw,typeI=t1,FDR=fdr))
}
cutsnp(fdr=res$post.probs$zRR,alpha=0.001,true=z)
cutsnp(fdr=res$post.probs$zRR,alpha=0.01,true=z)
cutsnp(fdr=res$post.probs$zRR,alpha=0.05,true=z)
#end:call snp#
```

---

estep *E step*

Description

This function calculates the E step of ECM algorithm for the model described in 'An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data'.

Usage

```
estep(mu, delta, pm1, p0, dat, cvg)
```

Arguments

mu	a vetor of the same length as number of positions: the position effect.
delta	a vetor of the same length as number of samples: the sample effect.
pm1	a single value,which is larger than 0 and less than 1: the probability of RR.
p0	a single value,which is larger than 0 and less than 1: the probability of RV.
dat	a n*m matrix: the ith row, jth column of the matrix represents the non-reference counts of ith sample at jth position.
cvg	a n*m matrix: the ith row, jth column of the matrix represents the depth of ith sample at jth position.

Details

The value of mu and delta must satisfy that each element of outer(delta,mu,"+") must less than zero. This is the requirement of the model described in paper "Genotyping for Rare Variant Detection Using Next-generation Sequencing Data."

Value

zRR	a n*m matrix: the posterior probabilities of genotype RR for n samples at m positions
zRV	a n*m matrix: the posterior probabilities of genotype RV for n samples at m positions
zVV	a n*m matrix: the posterior probabilities of genotype VV for n samples at m positions

Note

The most important function in this package is "ecm". "estep" is a function called by "ecm" to realize one E step in the whole process of iteration in "ecm".

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

logit *Logit Transformation*

Description

This function is for calculating the logit transformation:ln(x/(1-x))

Usage

```
logit(x)
```

Arguments

x	A numeric vector, whose elements are all greater than 0 and less than 1.

Value

ln(x/(1-x))

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

Examples

```
logit(0.5)
```

mstep *CM steps*

Description

This function calculates the CM steps of ECM algorithm for the model described in 'An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data'.

Usage

```
mstep(mu0, delta0, zm1, z0, zp1, dat, cvg, eps = 1e-06)
```

Arguments

mu0	a vetor of the same length as number of positions: the initial value of position effect mu.
delta0	a vetor of the same length as number of samples: the initial value of sample effect delta.
zm1	the output of estep: the posterior probabilities of genotype RR for n samples at m positions
z0	the output of estep: the posterior probabilities of genotype RV for n samples at m positions
zp1	the output of estep: the posterior probabilities of genotype VV for n samples at m positions
dat	a n*m matrix: the ith row, jth column of the matrix represents the non-reference counts of ith sample at jth position.
cvg	a n*m matrix: the ith row, jth column of the matrix represents the depth of ith sample at jth position.
eps	a single value: a threshold to control the convergence criterion. The default is 1e-06.

Details

The value of mu and delta must satisfy that each element of outer(delta,mu,"+") must less than zero. This is the requirement of the model described in 'An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data'.

Value

mu	the optimal value of mu in current CM steps
delta	the optimal value of delta in current CM steps
pRR	the optimal value of the probability of RR in current CM steps
pRV	the optimal value of the probability of RV in current CM steps

Note

The most important function in this package is "ecm". "mstep" is a function called by "ecm" to realize one M step(several CM steps) in the whole process of iteration in "ecm".

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

my.bisec *Bisection method to find the root*

Description

This function is to apply bisection method to find the root of a function f.

Usage

```
my.bisec(f, int.l, int.u, eps = 1e-06)
```

Arguments

f	the function for which the root is sought.
int.l	a vector containing the lower bound of the interval to be searched for the root. The length of the vector is the same as that of the input of function f.
int.u	a vector containing the upper bound of the interval to be searched for the root. The length of the vector is the same as that of the input of function f.
eps	a single value: a threshold to control the convergence criterion. The default is 1e-06.

Details

Both int.l and int.u must be specified: the upper bound must be strictly larger than the lower bound.

The function f must be well defined without invalid output(NA, nan, Inf, etc).

The length of the input of function f, the output of function f, int.l and int.u must be the same.

Value

a vector containing the root of the function. If there is no root in the interval (int.l, int.u), lower bound of the interval will be returned.

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

Examples

```
f <- function(x){
a <- 1:10
return(x-a)
}
my.bisec(f=f, int.l=rep(-1,10), int.u=rep(11,10), eps = 1e-08)
```

rlogit *Inverse Logit Transformation*

Description

This function is for calculating the inverse logit transformation:exp(x)/(1+exp(x))

Usage

```
rlogit(x)
```

Arguments

x	A numeric vector

Details

In order to avoid overflow, we define the function like this: rlogit <- function(x) return(ifelse(x>100,1,exp(x)/(1+exp(x))))

Value

exp(x)/(1+exp(x))

Author(s)

Na You <youn@mail.sysu.edu.cn> and Gongyi Huang<53hgy@163.com>

References

Na You and Gongyi Huang.(2016) An Empirical Bayes Method for Genotyping and SNP detection Using Multi-sample Next-generation Sequencing Data.

Examples

```
rlogit(-3)
```

Index