Package 'HMMASE'

February 4, 2014

Type Package

Title HMMASE R package

Version 1.0

Date 2014-02-04

Author Juan R. Steibel, Heng Wang, Ping-Shou Zhong

Maintainer Heng Wang <hengwang@msu.edu>

Description An R package that predicts cSNP genotypes from RNA sequence data in presence of allelic imbalance.

License GPL-2|GPL-3

Depends R.utils

R topics documented:

`HMMASE-package` *HMMASE R package*

Description

An R package that predicts cSNP genotypes from RNA sequence data in presence of allelic imbalance.

Details

Package:	HMMASE
Type:	Package
Version:	1.0
Date:	2014-02-04
License:	GPL-2 | GPL-3
Depends:	R.utils

Author(s)

Juan P. Steibel, Heng Wang, Ping-Shou Zhong

Maintainer: Heng Wang <hengwang@msu.edu>

References

J. Steibel, H. Wang, P.S. Zhong (2014). A Hidden Markov Approach for Ascertaining cSNP Genotypes from RNA Sequence Data in Presence of Allelic Imbalance by Exploiting Linkage Disequilibrium.

Examples

```
library(HMMASE)
library(R.utils)
data(total)
mbs<-membership(total,1000,2)
reps=1
epsilon0<-0.0001
alpha10=3
beta010=3
alpha20=3
beta020=3
maxiter=20
estimate.alpha=TRUE
common.alpha=TRUE
dist.dep=FALSE
ASE=TRUE

result<-select_run(gpr=5,members=mbs,tcon=total,epsilon0=epsilon0,
alpha10=alpha10, beta010=beta010, alpha20=alpha20, beta020=beta020,
estimate.alpha=estimate.alpha, common.alpha=common.alpha,
dist.dep=dist.dep, ASE=ASE)
```

HMM *function HMM()*

Description

The main function used to predict the genotype, ASE status and allelic specific ratio.

Usage

```
HMM(allecounts, dist, reps = 1, epsilon0 = 0.075, alpha10 = 2,
    beta010 = 2, alpha20 = 2, beta020 = 2, maxiter = 20, cnv = NULL,
    estimate.alpha = TRUE, common.alpha = FALSE, dist.dep = FALSE)
```

Arguments

`allecounts`	An array of #SNP * 2 * #animal. Element [s, 1, a] is the allele count of the first allele for s-th SNP of the a-th animal; and element [s, 2, a] is the allele count of the second allele for s-th SNP of a-th animal a.
`dist`	A vector of length #SNP-1 for the distances between two adjacent SNP markers.
`reps`	Number of replications. Set reps=1 for real data set.
`epsilon0`	Error rate, a value between 0 and 1.
`alpha10`	Starting values for the first parameter, alpha, in beta distribution for ASE-Low rate.
`beta010`	Starting values for the second parameter, beta, in beta distribution for ASE-Low ratio.
`alpha20`	Starting values for the first parameter, alpha, in beta distribution for ASE-High ratio.
`beta020`	Starting values for the second parameter, beta, in beta distribution for ASE-High ratio.
`maxiter`	Maximum number of iteration
`cnv`	Convergence criterion. HMM function stops with convergence either the number of estimated genotype state is different with less than T*n*cnv from the last iteration or the log-likelihood does not change. If cnv=NULL (by default), then the default value cnv=0.01.
`estimate.alpha`	TRUE or FALSE. alpha1 and alpha2 are estimated by HMM function if estimate.alpha==TRUE.
`common.alpha`	TRUE or FALSE. alpha1 is set to be equal to alpha2 if common.alpha==TRUE.
`dist.dep`	TRUE or FALSE. The transition matrix depends on distances between adjacent SNPs if dist.dep==TRUE.

Value

A list of 8.

`allecounts`	The original input data.
`GenoEsti`	3 dimentional array with genotype prediction result [SNP, Animal, replicate]. For real data, #replicate=1, so the dimension is [#SNPs, #Animals, 1]. 1 represents homogeneous genotype carrying two alleles form the first allele, 2 represents heterozygous genotype with no ASE, 3 represents heterozygous genotype with first allele ASE, 4 represents heterozygous genotype with second allele ASE, 5 represents homogeneous genotype carrying two alleles form the second allele.
`Parameters`	A vector of length 31 if dist.dep=TRUE. Estimation of paramter values after convergence. These parameters includes epsilon, alpha1, beta1, alpha2, beta2, rho, and a 5 by 5 transition matrix in a vector form. It is of length 30 if dist.dep=FALSE with the parameter rho deleted.

`Convergence`	Returns TRUE if convergence criterion is met. FALSE if convergence criterion is not met.
`niter`	Number of iterations take.
`ASEest`	The ASE ratio estimate for the ASE-High and ASE-Low SNPs.
`PostdenH`	The posterior densities of ASE ratio evaluated at a list ASE ratios from 0.5 to 1, for the ASE-High SNPs.
`PostdenL`	The posterior densities of ASE ratio evaluated at a list ASE ratios at from 0 to 0.5, for the ASE-Low SNPs.

Author(s)

J. Steibel, H. Wang, P.S. Zhong

References

J. Steibel, H. Wang, P.S. Zhong (2014). A Hidden Markov Approach for Ascertaining cSNP Genotypes from RNA Sequence Data in Presence of Allelic Imbalance by Exploiting Linkage Disequilibrium.

Examples

```
library("HMMASE")
library(R.utils)
data(total)
mbs<-membership(total,1000,2)
idid<-5


allecounts1<-total[mbs[[2]]==idid,]
allecounts<-array(0, c(nrow(allecounts1), 2, 24))
for (ani in 1:24)
{
  allecounts[,,ani]<-as.matrix(allecounts1[,c(3+ani*2, 4+ani*2)])
}

distance<-diff(allecounts1[,2])
reps<-1
epsilon0<-0.0001
alpha10=3
beta010=3
alpha20=3
beta020=3
maxiter=20
cnv=NULL
estimate.alpha=TRUE
common.alpha=TRUE
dist.dep=FALSE


result<-HMM(allecounts, dist=distance, reps=reps,
epsilon0=epsilon0, alpha10=alpha10, beta010=beta010, alpha20=alpha20,
beta020=beta020, common.alpha=common.alpha, dist.dep=dist.dep)
```

HMMNASE *function HMM()*

Description

The main function used to predict the genotype without considering allelic specific expression.

Usage

```
HMMNASE(allecounts, dist, reps = 1, epsilon0 = 0.075, maxiter = 20,
cnv = NULL, dist.dep = FALSE)
```

Arguments

allecounts	An array of #SNP * 2 * #animal. Element [s, 1, a] is the allele count of the first allele for s-th SNP of the a-th animal; and element [s, 2, a] is the allele count of the second allele for s-th SNP of a-th animal a.
dist	A vector of length #SNP-1 for the distances between two adjacent SNP markers.
reps	Number of replications. Set reps=1 for real data set.
epsilon0	Error rate, a value between 0 and 1.
maxiter	Maximum number of iteration
cnv	Convergence criterion. HMM function stops with convergence either the number of estimated genotype state is different with less than T*n*cnv from the last iteration or the log-likelihood does not change. If cnv=NULL (by default), then the default value cnv=0.01.
dist.dep	TRUE or FALSE. The transition matrix depends on distances between adjacent SNPs if dist.dep==TRUE.

Value

A list of 5.

allecounts	The original input data.
GenoEsti	3 dimentional array with genotype prediction result [SNP, Animal, replicate]. For real data, #replicate=1, so the dimension is [#SNPs, #Animals, 1]. 1 represents homogeneous genotype carrying two alleles form the first allele, 2 represents heterozygous genotype with no ASE, 3 represents heterozygous genotype with first allele ASE, 4 represents heterozygous genotype with second allele ASE, 5 represents homogeneous genotype carrying two alleles form the second allele.
Parameters	A vector of length 31 if dist.dep=TRUE. Estimation of paramter values after convergence. These parameters includes epsilon, alpha1, beta1, alpha2, beta2, rho, and a 5 by 5 transition matrix in a vector form. It is of length 30 if dist.dep=FALSE with the parameter rho deleted.
Convergence	Returns TRUE if convergence criterion is met. FALSE if convergence criterion is not met.
niter	Number of iterations take.

Author(s)

J. Steibel, H. Wang, P.S. Zhong

References

J. Steibel, H. Wang, P.S. Zhong (2014). A Hidden Markov Approach for Ascertaining cSNP Genotypes from RNA Sequence Data in Presence of Allelic Imbalance by Exploiting Linkage Disequilibrium.

Examples

```
library("HMMASE")
library(R.utils)
data(total)
mbs<-membership(total,1000,2)
idid<-5


allecounts1<-total[mbs[[2]]==idid,]
allecounts<-array(0, c(nrow(allecounts1), 2, 24))
for (ani in 1:24)
{
  allecounts[,,ani]<-as.matrix(allecounts1[,c(3+ani*2, 4+ani*2)])
}

distance<-diff(allecounts1[,2])
reps<-1
epsilon0<-0.0001
alpha10=3
beta010=3
alpha20=3
beta020=3
maxiter=20
cnv=NULL
estimate.alpha=TRUE
common.alpha=TRUE
dist.dep=FALSE


result<-HMMNASE(allecounts, dist=dist, reps=reps, epsilon0=epsilon0,
dist.dep=dist.dep)
```

`membership` *membership function.*

Description

membership() divides the total data set into several small pieces. Each piece contains at least one SNP with genotype from SNP chip data, which is considered as a "gold standard" in this our real data analysis. Only useful in the example data. For general use, see membershipc().

Usage

```
membership(alc, maxdist, minsnp)
```

Examples

```
## The function is currently defined as
function (alc, maxdist, minsnp)
{
    memb <- rep(0, nrow(alc))
    mdi <- maxdist
    ns <- c(NULL, NULL)
    psts <- alc[alc[, 53], 2]
    j <- 0
    for (i in psts) {
        j <- j + 1
        dst <- alc[, 2] - i
        repeat {
            selected <- abs(dst) < mdi
            if (sum(selected) >= minsnp) {
                break
            }
            else {
                mdi <- mdi + 1000
            }
        }
        ns <- rbind(ns, c(sum(selected), mdi))
        mdi <- maxdist
        memb[selected] <- j
    }
    return(list(ns, memb))
  }
```

`membershipc` *membershipc function.*

Description

membershipc() divides the total data set into several small pieces. Each piece contains at least one SNP with genotype from SNP chip data, which is considered as a "gold standard" in this our real data analysis. For general use.

Usage

```
membershipc(alc, maxdist, minsnp)
```

Arguments

`alc`	Input data set with the following format: a column with chromosome, position, reference allele, alternative allele and then 2 columns per sample: count for reference and alternative.
`maxdist`	maximum distance to consider
`minsnp`	minimum number of SNP in segment

Value

A list of 2.

comp1	a two column matrix with the first column recording the number of SNPs of each segment, and the second column recording the length of that segment.
comp2	a vector indicates the segmentational membership of each SNP.

Examples

```
## The function is currently defined as
function (alc, maxdist, minsnp)
{
    memb <- rep(0, nrow(alc))
    mdi <- maxdist
    ns <- c(NULL, NULL)
    psts <- alc[, 2]
    j <- 1
    i <- 1
    repeat {
        dst <- psts - psts[j]
        repeat {
            selected <- (abs(dst) < mdi) & (dst >= 0)
            toselect <- (dst >= mdi)
            if (sum(selected) >= minsnp) {
                break
            }
            else {
                mdi <- mdi + 1000
            }
        }
        if (sum(toselect) < minsnp) {
            selected <- (toselect | selected)
            toselect <- toselect & F
            mdi <- max(psts[selected]) - min(psts[selected])
        }
        ns <- rbind(ns, c(sum(selected), mdi))
        mdi <- maxdist
        memb[selected] <- i
        i <- i + 1
        j <- j + sum(selected)
        if (sum(toselect) == 0) {
            break
        }
    }
    return(list(ns, memb))
  }
```

---

select_run *select_run function*

Description

A function to run a particular SNP segment.

Usage

```
select_run(gpr, members, tcon, epsilon0=0.0001, alpha10 = 3,
           beta010 = 3, alpha20 = 3, beta020 = 3, estimate.alpha = TRUE,
           common.alpha = TRUE, dist.dep = FALSE, ASE=TRUE)
```

Arguments

gpr	Group index to run
members	Membership information obtained from membershipc() or membership()
tcon	Original data set in data(total)
epsilon0	Error rate, a value between 0 and 1.
alpha10	Starting values for the first parameter, alpha, in beta distribution for ASE-Low ratio.
beta010	Starting values for the second parameter, beta, in beta distribution for ASE-Low ratio.
alpha20	Starting values for the first parameter, alpha, in beta distribution for ASE-High ratio.
beta020	Starting values for the second parameter, beta, in beta distribution for ASE-High ratio.
estimate.alpha	TRUE or FALSE for whether or not estimating the alpha parameter in the beta distributions. By default estimate.alpha = TRUE.
common.alpha	TRUE or FALSE. alpha1 is set to be the same as alpha2 if common.alpha=TRUE.
dist.dep	TRUE or FALSE. If dist.dep = TRUE, the transition probability is modeled as a function of distances among adjacent SNPs. By default, dist.dep = FALSE.
ASE	TRUE or FALSE. If ASE = TRUE, the algorithm takes allelic specific expression into consideration when genotyping. In this case, there are 5 possible genotype outcomes: two homozygous genotypes and three heterozygous genotypes (without ASE, with ASE with one reads more than the other, with ASE with one reads less than the other). If ASE = FALSE, the algorithm does not take allelic specific expression into consideration. In this case, there are 3 possible genotype outcomes: two homozygous genotypes and one heterozygous genotype. By default, dist.dep = TRUE.

Value

A list of 8 if ASE is TRUE; and a list of 5 if ASE is FALSE.

allecounts	The original input data.
GenoEsti	A 3 dimentional array with genotype prediction result [SNP, Animal, replicate]. For real data, #replicate=1, so the dimension is [#SNPs, #Animals, 1]. 1 represents homogeneous genotype carrying two alleles form the first allele, 2 represents heterozygous genotype with no ASE, 3 represents heterozygous genotype with first allele ASE, 4 represents heterozygous genotype with second allele ASE, 5 represents homogeneous genotype carrying two alleles form the second allele.
Parameters	A vector of length 31 if dist.dep=TRUE. Estimation of paramter values after convergence. These parameters includes epsilon, alpha1, beta1, alpha2, beta2, rho, and a 5 by 5 transition matrix in a vector form. It is of length 30 if dist.dep=FALSE with the parameter rho deleted.

`Convergence`	Returns TRUE if convergence criterion is met. FALSE if convergence criterion is not met.
`niter`	Number of iterations take.
`ASEest`	Only applies when ASE = TRUE. The ASE ratio estimate for the ASE-High and ASE-Low SNPs.
`PostdenH`	Only applies when ASE = TRUE. The posterior densities of ASE ratio evaluated at a list of ASE ratio from 0.5 to 1, for the ASE-High SNPs.
`PostdenL`	Only applies when ASE = TRUE. The posterior densities of ASE ratio evaluated at a list of ASE ratio from 0 to 0.5, for the ASE-Low SNPs.

Author(s)

J. Steibel, H. Wang, P.S. Zhong

References

J. Steibel, H. Wang, P.S. Zhong (2014). A Hidden Markov Approach for Ascertaining cSNP Genotypes from RNA Sequence Data in Presence of Allelic Imbalance by Exploiting Linkage Disequilibrium.

See Also

See Also as `HMM`

Examples

```
library(HMMASE)
library(R.utils)
data(total)
mbs<-membership(total,1000,2)
reps=1
epsilon0<-0.0001
alpha10=3
beta010=3
alpha20=3
beta020=3
maxiter=20
estimate.alpha=TRUE
common.alpha=TRUE
dist.dep=FALSE
ASE=TRUE

result<-select_run(gpr=5,members=mbs,tcon=total,epsilon0=epsilon0,
alpha10=alpha10, beta010=beta010, alpha20=alpha20, beta020=beta020,
estimate.alpha=estimate.alpha, common.alpha=common.alpha,
dist.dep=dist.dep, ASE=ASE)
```

`total` *An example data set named "total".*

Description

The example data set.

Usage

```
data(total)
```

Format

A data frame with 5364 observations on the following 53 variables.

`V1` a numeric vector, the chromosome ID (13)

`V2` a numeric vector, physical SNP locations

`V3` a factor with levels `A C G T`, for one haplotype

`V4` a factor with levels `A C G T`, for the other haplotype

`V5`**,** `V7`**,** `V9`**, ...,** `V51` The read counts for the first haplotype (V3) of the 24 animals.

`V6`**,** `V8`**,** `V10`**, ...,** `V52` The read counts for the second haplotype (V4) of the 24 animals.

`V53` a logical vector, indicating if the SNP have the chip data information.

Examples

```
data(total)
```

Index