Package 'PLIS'

October 12, 2022

Type Package

Title Multiplicity Control using Pooled LIS Statistic

Version 1.2

Date 2022-10-01

Author Zhi Wei & Wenguang Sun

Maintainer Zhi Wei <zhiwei04@gmail.com>

Description A multiple testing procedure for testing several groups of hypotheses is implemented. Linear dependency among the hypotheses within the same group is modeled by using hidden Markov Models. It is noted that a smaller p value does not necessarily imply more significance due to the dependency. A typical application is to analyze genome wide association studies datasets, where SNPs from the same chromosome are treated as a group and exhibit strong linear genomic dependency. See Wei Z, Sun W, Wang K, Hakonarson H (2009) <doi:10.1093/bioinformatics/btp476> for more details.

License GPL-3

Repository CRAN

Date/Publication 2022-10-02 15:40:02 UTC

NeedsCompilation no

R topics documented:

PLIS-package *PLIS*

Description

PLIS is a multiple testing procedure for testing several groups of hypotheses. Linear dependency is expected from the hypotheses within the same group and is modeled by hidden Markov Models. It is noted that, for PLIS, a smaller p value does not necessarily imply more significance because of dependency among the hypotheses. A typical applicaiton of PLIS is to analyze genome wide association studies datasets, where SNPs from the same chromosome are treated as a group and exhibit strong linear genomic dependency.

Details

Package:	PLIS
Type:	Package
Version:	1.0
Date:	2012-08-08
License:	GPL-3
LazyLoad:	yes

main functions: em.hmm & plis

Author(s)

Wei Z, Sun W, Wang K and Hakonarson H
Maintainer: Zhi Wei <zhiwei04@gmail.com>

References

Wei Z, Sun W, Wang K and Hakonarson H, Multiple Testing in Genome-Wide Association Studies via Hidden Markov Models, Bioinformatics, 2009

See Also

p.adjust(), in which the traditional procedures are implemented. The adjustment made by p.adjust will not change the original ranking based on the given p values. However, taking into account dependency, PLIS may generate a ranking different from that by p value.

bwfw.hmm *backward and forward inferences*

Description

When L>1, calculate values for backward, forward variables, probabilities of hidden states. A supporting function called by em.hmm.

Usage

```
bwfw.hmm(x, pii, A, pc, f0, f1)
```

Arguments

x	the observed Z values
pii	(prob. of being 0, prob. of being 1), the initial state distribution
A	A=(a00 a01\\ a10 a11), transition matrix
pc	(c[1], ..., c[L])–the probability weights in the mixture for each component
f0	(mu, sigma), the parameters for null distribution
f1	(mu[1], sigma[1]\\...\\mu[L], sigma[L])–an L by 2 matrix, the parameter set for the non-null distribution

Details

calculates values for backward, forward variables, probabilities of hidden states,
–the lfdr variables and etc.
–using the forward-backward procedure (Rabiner 89)
–based on a sequence of observations for a given hidden markov model M=(pii, A, f)
–see Sun and Cai (2009) for a detailed instruction on the coding of this algorithm

Value

alpha	rescaled backward variables
beta	rescaled forward variables
lfdr	lfdr variables
gamma	probabilities of hidden states
dgamma	rescaled transition variables
omega	rescaled weight variables

Author(s)

Wei Z, Sun W, Wang K and Hakonarson H

References

Multiple Testing in Genome-Wide Association Studies via Hidden Markov Models, Bioinformatics, 2009
Large-scale multiple testing under dependence, Sun W and Cai T (2009), JRSSB, 71, 393-424
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Rabiner L (1989), Procedings of the IEEE, 77, 257-286.

`bwfw1.hmm` *backward and forward inferences*

Description

When L=1, calculate values for backward, forward variables, probabilities of hidden states. A supporting function called by em.hmm.

Usage

```
bwfw1.hmm(x, pii, A, f0, f1)
```

Arguments

`x`	the observed Z values
`pii`	(prob. of being 0, prob. of being 1), the initial state distribution
`A`	A=(a00 a01\\a10 a11), transition matrix
`f0`	(mu, sigma), the parameters for null distribution
`f1`	(mu[1], sigma[1]\\...\\mu[L], sigma[L])–an L by 2 matrix, the parameter set for the non-null distribution

Details

calculates values for backward, forward variables, probabilities of hidden states,
–the lfdr variables and etc.
–using the forward-backward procedure (Rabiner 89)
–based on a sequence of observations for a given hidden markov model M=(pii, A, f)
–see Sun and Cai (2009) for a detailed instruction on the coding of this algorithm

Value

`alpha`	rescaled backward variables
`beta`	rescaled forward variables
`lfdr`	lfdr variables
`gamma`	probabilities of hidden states
`dgamma`	rescaled transition variables
`omega`	rescaled weight variables

Author(s)

Wei Z, Sun W, Wang K and Hakonarson H

References

Multiple Testing in Genome-Wide Association Studies via Hidden Markov Models, Bioinformatics, 2009
Large-scale multiple testing under dependence, Sun W and Cai T (2009), JRSSB, 71, 393-424
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Rabiner L (1989), Procedings of the IEEE, 77, 257-286.

`em.hmm` *EM algorithm for HMM to estimate LIS statistic*

Description

em.hmm calculates the MLE for a HMM model with hidden states being 0/1. the distribution of observed Z values given state 0 is assumed to be normal and gvien state 1, is assumed to be a normal mixture with L components

Usage

```
em.hmm(x, L=2, maxiter = 1000, est.null = FALSE)
```

Arguments

`x`	the observed Z values
`L`	the number of components in the non-null mixture, default value=2
`maxiter`	the maximum number of iterations, default value=1000
`est.null`	logical. If FALSE (the default) set the null distribution as N(0,1), otherwise will estimate the null distribution.

Details

None.

Value

`pii`	the initial state distribution, pii=(prob. of being 0, prob. of being 1)
`A`	transition matrix, A=(a00 a01\ a10 a11)
`f0`	the null distribution
`pc`	probability weights of each component in the non-null mixture
`f1`	an L by 2 matrix, specifying the dist. of each component in the non-null mixture
`LIS`	the LIS statistics

ni	the number of iterations excecuted
logL	log likelihood
BIC	BIC score for the estimated model
converged	Logic, Convergence indicator of the EM procedure

Author(s)

Wei Z, Sun W, Wang K and Hakonarson H

References

Multiple Testing in Genome-Wide Association Studies via Hidden Markov Models, Bioinformatics, 2009

See Also

plis

Examples

```
##(1) Example for analyzing simulated data
grp1.nonNull.loci=c(21:30, 51:60); grp2.nonNull.loci=c(41:60)
grp1.theta<-grp2.theta<-rep(0,200)
grp1.theta[grp1.nonNull.loci]=2; grp2.theta[grp2.nonNull.loci]=2

grp1.zval=rnorm(n=length(grp1.theta),mean=grp1.theta)
grp2.zval=rnorm(n=length(grp2.theta),mean=grp2.theta)
##Group 1
#Use default L=2
grp1.L2rlts=em.hmm(grp1.zval)
#Use true value L=1
grp1.L1rlts=em.hmm(grp1.zval,L=1)
#Choose L by BIC criteria
grp1.Allrlts=sapply(1:3, function(k) em.hmm(grp1.zval,L=k))
BICs=c()
for(i in 1:3) {
  BICs=c(BICs,grp1.Allrlts[[i]]$BIC)
}
grp1.BICrlts=grp1.Allrlts[[which(BICs==max(BICs))]]

rank(grp1.BICrlts$LIS)[grp1.nonNull.loci]
rank(-abs(grp1.zval))[grp1.nonNull.loci]

##Group 2
grp2.Allrlts=sapply(1:3, function(k) em.hmm(grp2.zval,L=k))
BICs=c()
for(i in 1:3) {
  BICs=c(BICs,grp2.Allrlts[[i]]$BIC)
}
grp2.BICrlts=grp2.Allrlts[[which(BICs==max(BICs))]]
```

```
rank(grp2.BICrlts$LIS)[grp2.nonNull.loci]
rank(-abs(grp2.zval))[grp2.nonNull.loci]

##PLIS: control global FDR
states=plis(c(grp1.BICrlts$LIS,grp2.BICrlts$LIS),fdr=0.1,adjust=FALSE)$States
#0 accept; 1 reject under fdr level 0.1

##(2) Example for analyzing Genome-Wide Association Studies (GWAS) data
#Information in GWAS.SampleData can be obtained by using PLINK
#http://pngu.mgh.harvard.edu/~purcell/plink/

#not running
#please uncomment to run
#
#data(GWAS.SampleData)
#
#chr1.data=GWAS.SampleData[which(GWAS.SampleData[,"CHR"]==1),]
#chr6.data=GWAS.SampleData[which(GWAS.SampleData[,"CHR"]==6),]
#
##Make sure SNPs in the linear physical order
#chr1.data<-chr1.data[order(chr1.data[,"BP"]),]
#chr6.data<-chr6.data[order(chr6.data[,"BP"]),]
#
##convert p values by chi_sq test to z values; odds ratio (OR) is needed.
#chr1.zval<-rep(0, nrow(chr1.data))
#chr1.ors=(chr1.data[,"OR"]>1)
#chr1.zval[chr1.ors]<-qnorm(chr1.data[chr1.ors, "P"]/2, 0, 1, lower.tail=FALSE)
#chr1.zval[!chr1.ors]<-qnorm(chr1.data[!chr1.ors, "P"]/2, 0, 1, lower.tail=TRUE)
#chr1.L2rlts=em.hmm(chr1.zval)
#
#chr6.zval<-rep(0, nrow(chr6.data))
#chr6.ors=(chr6.data[,"OR"]>1)
#chr6.zval[chr6.ors]<-qnorm(chr6.data[chr6.ors, "P"]/2, 0, 1, lower.tail=FALSE)
#chr6.zval[!chr6.ors]<-qnorm(chr6.data[!chr6.ors, "P"]/2, 0, 1, lower.tail=TRUE)
#chr6.L2rlts=em.hmm(chr6.zval)
#
##Note that for analyzing a chromosome in real GWAS dataset, em.hmm can take as long as 10+ hrs
##L=2 or 3 is recommended for GWAS based on our experience
##em.hmm can be run in parallel for different chromosomes before applying the PLIS procedure
#plis.rlts=plis(c(chr1.L2rlts$LIS,chr6.L2rlts$LIS),fdr=0.01)
#all.Rlts=cbind(rbind(chr1.data,chr6.data), LIS=c(chr1.L2rlts$LIS,chr6.L2rlts$LIS),
#gFDR=plis.rlts$aLIS, fdr001state=plis.rlts$States)
#all.Rlts[order(all.Rlts[,"LIS"])[1:10],]
```

GWAS.SampleData *Sample GWAS Dataset*

Description

Sample GWAS Dataset with 400 SNPs from Chromosome 1 and 6 (200 SNPs each).

Usage

```
data(GWAS.SampleData)
```

Format

A data frame with 400 observations on the following 6 variables.

`CHR` Chromosome ID

`SNP` rs Id

`BP` Phisical Position

`OR` Odds Ratio

`CHISQ` 1 d.f. Chi Square test Statistic

`P` P value of 1 d.f. Chi Square test Statistic

Details

The required values (Odds ratio and P value) can be calculated by using PLINK

References

Supplementary Material of Multiple Testing in Genome-Wide Association Studies via Hidden Markov Models, Bioinformatics, 2009

Examples

```
data(GWAS.SampleData)
```

---

plis *A multiple testing procedure based on pooled LIS statistics*

Description

It controls the global FDR for the pooled hypotheses from different groups

Usage

```
plis(lis, fdr = 0.001, adjust = TRUE)
```

Arguments

`lis`	pooled LIS statistics estimated from different groups
`fdr`	nominal fdr level you want to control
`adjust`	logical. If TRUE (the default), will calculate and return "adjusted" LIS value– the corresponding global FDR if using the LIS statistic as the significance cutoff. It may take hours if you have hundreds of thousands LISs to adjust.

Value

States	state sequence indicating if the hypotheses should be rejected or not: 0 accepted , 1 rejected
aLIS	the corresponding global FDR if using the LIS statistic as the significance cutoff

Author(s)

Wei Z, Sun W, Wang K and Hakonarson H

References

Multiple Testing in Genome-Wide Association Studies via Hidden Markov Models, Bioinformatics, 2009

See Also

see em.hmm for examples

Index