Package 'maew'

April 17, 2017

Type Package

Title Multiplicity-Adjusted Evidence Weights

Version 0.6

Author Wenjian Bi <wenjian.bi@stjude.org>, Guolian Kang <guolian.kang@stjude.org>, Stan Pounds <stanley.pounds@stjude.org>.

Maintainer Wenjian Bi <wenjian.bi@stjude.org>

Depends R (>= 2.15.1), survival

Date 2017-04-16

Description Traditional Bayesian information criterion (BIC) and evidence weights are very common in model selection.
As for GWAS or NGS, we adjust the weights for multiplicity based on EBP values to capture information about the most appropriate model.
The adjusted evidence weights can be employed to select the most appropriate model for designing validation analysis.

License GPL-2

LazyData yes

LazyLoad yes

R topics documented:

maew-package 1
compute.BIC 2
compute.maew 4
pval.ebp 5
revise.maew 6

Index **8**

`maew-package` *An R Package to Compute Multiplicity-Ajusted Evidence Weights for Model Selection*

Description

Traditional Bayesian information criterion (BIC) and evidence weights are very common in model selection. As for GWAS or NGS, we adjust the weights for multiplicity based on EBP values to capture information about the most appropriate model. The adjusted evidence can be employed to select the most appropriate model for designing validation analysis.

Details

Package:	maew
Type:	Package
Version:	0.6
Date:	2017-04-16
License:	GPL(>=2)

Author(s)

Wenjian Bi <wenjian.bi@stjude.org> Guolian Kang <guolian.kang@stjude.org> Stan Pounds <stanley.pounds@stjude.org>

Maintainer: Wenjian Bi <wenjian.bi@stjude.org>

References

Bi WJ, Kang GL, Pounds SB. (2017) Multiplicity Adjusted Evidence Weights for Associating Phenotypes with Genotypes (Manuscript).

`compute.BIC` *Compute Bayesion information criterion*

Description

For each genomic feature, compute Bayesian information criterions (BICs) for each candidate model. Candidate models include null, additive, dominant, recessive, and co-dominant models.

Usage

```
compute.BIC (Gtype, Ptype, Covars = NULL, model.string = c("glm",
    "coxph"), IC = c("BIC", "AIC"), n.cmds = 5, ...)
```

Arguments

`Gtype`	numeric matrix of genotype with one row per subject, one column per SNP locus. No copy, one copy, two copies of the minor alleles should be assigned as 0, 1, 2, respectively. Rownames should be subj.id, and colnames should be snp.id. Missing data should be assigned as NA.
`Ptype`	numeric vector of observed phenoytpe with names of subj.id

Covars	numeric matrix of covariates with one row per subject, one column per covariate. Rownames should be subj.id, and colnames should be covar.id. Missing data should be assigned as NA. The default is NULL, i.e., there is no covariate to be adjusted for.
model.string	character string specifying the method to compute BICs/AICs. "glm" is for continuous or binary phenotype, "coxph" is for survivial phenotype.
IC	character string specifying model selection method. Two options include "BIC" (default) and "AIC".
n.cmds	an odd number (greater than or equal to 5) to specify candidate models number. Default setting is 5 candidate models including null, additive, dominant, recessive, and co-dominant model. More information can be seen in 'Details'.
...	Further auguments to be passed to glm or coxph. If phenotype is binary, family=binomial is needed.

Details

Parameter 'n.cmds' is an odd number (greater than or equal to 5) to specify candidate models number. If n.cmds=5, we consider 5 candidate models including null model, co-dominant model, additive model(0), dominant model(1) and recessive model(-1). If model type is "codom"(Non-specific alternative model), we regard different genotypes as unordered category factors. For other models, we give different definition of design matrix based on candidate model parameter "x". For each feature, no copy, one copy and two copies of the minor allels is given -1, x, and 1 in design matrix, respectively. Hence, x=-1 indicate recessive model, x=0 indicates additive model, and x=1 indicates dominant model.

Value

An integrated matrix with each row for one feature. Columns contain the following components:

pval.xxx	p-values following model xxx. pval.-1 is p-value of recessive model, pval.0 is p-value of additive model, pval.1 is p-value of dominant model. Please see details for definiation of other candidate models.
BIC.xxx	BIC values for model xxx.
coef.xxx	estimated coefficients for model xxx.

Note

For each feature, if no copy of the minor alleles in the cohort, the corresponding row of the output matrix is NA. If no subjects with two copies of the minor alleles in the cohort, the corresponding row of the output matrix has NA of value "coef.-1" (coefficient of recessive model).

Examples

```
## Simulate 100 subjects with 10 SNPs and 2 Covariates. Minor allele frequency is 0.3.
## (1) Analyze continuous phenotype
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- rnorm(100);
compute.BIC(Gtype,Ptype,Covars,model.string="glm");
## (2) Analyze binary phenotype
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- rbinom(100,size=1,prob=0.5);
```

```
compute.BIC(Gtype,Ptype,Covars,model.string="glm",family=binomial)
## (3) Analyze survival phenotype
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- Surv(rexp(100),rbinom(100,size=1,prob=0.5)) # survival phenotype
compute.BIC(Gtype,Ptype,Covars,model.string="coxph")
```

`compute.maew` *Compute MAEW Values*

Description

For each genomic feature, compute multiplicity adjusted evidence weights (MAEWs) for each candidate model based on the output of compute.BIC

Usage

```
compute.maew (BIC.res, cmd.para = "codom", no.false.brks = 0.95,
    large.p = 0.8)
```

Arguments

`BIC.res`	an object as returned by the compute.BIC function. See 'Value of help(compute.BIC)'.
`cmd.para`	character of "codom" (default) or a number with range from -1 to 1 to characterize specific model to compute EBP q-value. See 'Details of help(compute.BIC)'
`no.false.brks`	select cut-off for log-likelihood to refine bins with probability of false breakpoints controlled below this value, 0.95 or 0.99.
`large.p`	do not create new bins with breakpoints above this p-value

Value

An integrated matrix with each row for one feature. Columns contain the following components:

`p`	p-values for model of 'cmd.para' in column 1
`q`	EBP q-values in column 2
`maew.xxx`	MAEW value for model xxx. See 'Value of help(compute.BIC)'.
`coef.xxx`	estimated coefficient for model xxx. See 'Value of help(compute.BIC)'.

Examples

```
## Simulate 100 subjects with 10 SNPs and 2 Covariates. Minor allele frequency is 0.3.
## (1) Assume the phenotype is continuous
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- rnorm(100);
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="glm");
compute.maew(res.aics);
## (2) Assume the phenotype is binary
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
```

```
Ptype <- rbinom(100,size=1,prob=0.5);
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="glm",family=binomial)
compute.maew(res.aics);
## (3) Assume the phenotype is survival
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- Surv(rexp(100),rbinom(100,size=1,prob=0.5)) # survival phenotype
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="coxph")
compute.maew(res.aics);
```

`pval.ebp` *EBP Computation*

Description

Use the p-value histogram to estimate the empirical bayesian probability (EBP) of each genomic feature.

Usage

```
pval.ebp (p, no.false.brks = 0.95, large.p = 0.8)
```

Arguments

`p`	a numeric vector of p-values
`no.false.brks`	select cut-off for log-likelihood to refine bins with probability of false breakpoints controlled below this value, 0.95 or 0.99.
`large.p`	do not create new bins with breakpoints above this p-value

Value

a numeric matrix with p-values in column 1, EBP in column 2

References

Pounds SB, Gao CL, Zhang H (2012). Empirical Bayesian Selection of Hypothesis Testing Procedures for Analysis of Sequence Count Expression Data. Statistical Applications in Genetics and Molecular Biology, 11: 5.

Examples

```
p <- runif(1000,0,1)
res <- pval.ebp(p)
EBP <- res[,"EBP"]
```

`revise.maew` *Revise multiplicity adjusted evidence weights (MAEWs) based on customized p-values*

Description

Besides AIC values, the computaion of MAEW relies on p-values. In the default setting, p-values are computed based on the same model as the computation of AICs. revise.maew can revise the MAEWs based on customized p-values provided by users.

Usage

```
revise.maew (p, BIC.res)
```

Arguments

`p`	a numeric vector of p-values
`BIC.res`	an object as returned by the compute.BIC function. See 'Value of help(compute.BIC)'.

Value

An integrated matrix with the same form as the output of compute.maew

Examples

```
## Simulate 100 subjects with 10 SNPs and 2 Covariates. Minor allele frequency is 0.3.
## (1) Assume the phenotype is continuous, we use Kruskal-Wallis test instead of linear regression method.
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- rnorm(100);
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="glm");
p.kw <- apply(Gtype,2,FUN=function(gtype){res.kw<-kruskal.test(Ptype~gtype);return(res.kw$p.value)})
revise.maew(p.kw,res.aics);
## (2) Assume the phenotype is binary, we use chi-square test instead of logistic regression method.
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- rbinom(100,size=1,prob=0.5);
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="glm",family=binomial)
p.chisq <- apply(Gtype,2,FUN=function(gtype){res.chisq<-chisq.test(gtype,Ptype);return(res.chisq$p.value)}
revise.maew(p.chisq,res.aics);
## (3) Assume the phenotype is binary, we use fisher exact test instead of logistic regression method.
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- rbinom(100,size=1,prob=0.5);
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="glm",family=binomial)
p.fisher <- apply(Gtype,2,FUN=function(gtype){res.fisher<-fisher.test(gtype,Ptype);return(res.fisher$p.val
revise.maew(p.fisher,res.aics);
## (4) Assume the phenotype is survival
Gtype <- matrix(rbinom(n=1000,size=2,prob=0.3),nrow=100,ncol=10);
Covars <- matrix(rnorm(200),nrow=100,ncol=2);
Ptype <- Surv(rexp(100),rbinom(100,size=1,prob=0.5)) # survival phenotype
res.aics <- compute.BIC(Gtype,Ptype,Covars,model.string="coxph")
p.new <- runif(10);
```

```
revise.maew(p.new,res.aics);
```

Index

compute.BIC, 2
compute.maew, 4

maew *(*maew-package*)*, 1
maew-package, 1

pval.ebp, 5

revise.maew, 6