

Package ‘haplo.ccs’

April 17, 2009

Type Package

Title Estimate Haplotype Relative Risks in Case-Control Data

Version 1.3

Date 2007-08-06

Author Benjamin French and Thomas Lumley

Depends R (>= 2.0.1), haplo.stats, survival

Maintainer Benjamin French <bcf@u.washington.edu>

Description ‘haplo.ccs’ estimates haplotype and covariate relative risks in case-control data by weighted logistic regression. Diplootype probabilities, which are estimated by EM computation with progressive insertion of loci, are utilized as weights.

License GPL (>= 2)

Repository CRAN

Date/Publication 2007-08-09 08:02:01

R topics documented:

haplo	2
haplo.ccs	3
haplo.hpp	7
renin	8
sandcov	9

Index	11
--------------	-----------

haplo	<i>Specify an Allele Matrix, Inheritance Mode, and Grouping for Rare Haplotypes</i>
-------	---

Description

'haplo' specifies an allele matrix and an inheritance mode for the 'haplo.ccs' model formula. 'haplo' also specifies preferences for grouping rare haplotypes.

Usage

```
haplo(..., mode, group.rare=TRUE, rare.freq=0.02)
```

Arguments

...	a matrix of alleles or list of columns of an allele matrix. Each locus on the chromosome has a pair of adjacent columns in the allele matrix, so that the number of columns of the allele matrix is twice the number of loci on the chromosome. The order of the columns corresponds to the order of the loci on the chromosome, and each row corresponds to the alleles for each subject. The alleles should be numerically coded, i.e., 1, 2, 3, or 4 for 'A', 'C', 'G', or 'T'.
mode	the inheritance mode, either 'additive', 'dominant', or 'recessive'. Note that the default inheritance mode is 'additive'.
group.rare	a logical value indicating whether rare haplotypes should be grouped in the 'haplo.ccs' model. Note that the default is to group rare haplotypes.
rare.freq	the population haplotype frequency to define rare haplotypes. If 'group.rare=TRUE', then haplotypes with an estimated population frequency less than or equal to 'rare.freq' are grouped in the 'haplo.ccs' model. Note that the default haplotype frequency is 0.02. 'rare.freq' is automatically set to 0 if 'group.rare=FALSE'.

Value

A matrix of alleles with mode, group.rare, and rare.freq assigned as attributes.

Author(s)

Benjamin French and Thomas Lumley, University of Washington

References

French B, Lumley T, Monks SA, Rice KM, Hindorff LA, Reiner AP, Psaty BM. Simple estimates of haplotype relative risks in case-control data. *Genetic Epidemiology* 2006; 30(6):485-494.

See Also[haplo.ccs](#)**Examples**

```
data(renin)

## Specify an allele matrix in a model fit by 'haplo.ccs'.
haplo.ccs(case ~ haplo(geno))

## Specify dominant inheritance and define rare haplotypes.
haplo.ccs(case ~ haplo(geno, mode="dominant", rare.freq=0.01))

## Specify the allele matrix without grouping rare haplotypes.
haplo.ccs(case ~ haplo(geno, group.rare=FALSE))
```

`haplo.ccs`*Estimate Haplotype Relative Risks in Case-Control Data*

Description

'haplo.ccs' estimates haplotype and covariate relative risks in case-control data by weighted logistic regression. Diplotype probabilities, which are estimated by EM computation with progressive insertion of loci, are utilized as weights. The model is specified by a symbolic description of the linear predictor, which includes specification of an allele matrix, inheritance mode, and preferences for rare haplotypes using 'haplo'. Note that use of this function requires installation of the 'haplo.stats' and 'survival' packages. See 'haplo.em' for a description of EM computation of diplotype probabilities. Currently missing genotype information is not allowed.

Usage

```
haplo.ccs(formula, data=NULL, ...)

haplo.ccs.fit(y, x, int, geno, inherit.mode, group.rare, rare.freq,
              referent, names.x, names.int, ...)
```

Arguments

`formula` a symbolic description of the model to be fit, which requires specification of an allele matrix and inheritance mode using 'haplo'. Note that 'additive' is the default inheritance mode for 'haplo'. Preferences for grouping rare haplotypes

are also specified using 'haplo'. Note that by default 'haplo' groups haplotypes with an estimated population frequency less than 0.02. More details on model formulae are given below.

<code>data</code>	an optional data frame, list, or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in 'data', the variables are taken from 'environment(formula)', typically the environment from which 'haplo.ccs' is called.
<code>referent</code>	a character string representing the haplotype to be used as the referent. The haplotype with the highest estimated population frequency is the default referent.
<code>...</code>	optional model-fitting arguments to be passed to 'glm'.
<code>y</code>	a vector of observations.
<code>x</code>	the design matrix for environmental covariates.
<code>int</code>	the design matrix for haplotype-environment interaction.
<code>geno</code>	the allele matrix.
<code>inherit.mode</code>	the inheritance mode specified by 'haplo'.
<code>group.rare</code>	a logical value indicating whether rare haplotypes should be grouped, specified by 'haplo'.
<code>rare.freq</code>	the population haplotype frequency used to define the rare haplotypes, specified by 'haplo'.
<code>names.x</code>	the column names of the design matrix for covariates.
<code>names.int</code>	the column names of the design matrix for haplotype-environment interaction.

Details

A formula has the form 'y ~ terms' where 'y' is a numeric vector indicating case-control status and 'terms' is a series of terms which specifies a linear predictor for 'y'. A terms specification of the form 'first + second' indicates all the terms in 'first' together with all the terms in 'second' with duplicates removed. The terms in the formula will be re-ordered so that main effects come first, followed by the interactions, all second-order, all third-order and so on. The specification 'first*second' indicates the cross of 'first' and 'second'.

Note that 'haplo.ccs.fit' is the workhorse function. The inputs 'y', 'x', 'geno', and 'int' represent case-control status, the matrix of covariates, the matrix of alleles, and the matrix of terms that have interaction with the haplotypes to be estimated from the alleles. The argument 'inherit.mode' corresponds to the inheritance mode specified by 'haplo', and the arguments 'group.rare' and 'rare.freq' correspond to the preferences for grouping rare haplotypes specified by 'haplo'. 'names.x' and 'names.int' correspond to the column names of 'x' and 'int', respectively. The background functions 'one', 'count.haps', and 'return.haps' are used in specifying the model terms and neatly packaging the results.

Value

'haplo.ccs' returns an object of class inheriting from "'haplo.ccs'". More details appear later in this section. The function 'summary' (i.e., 'summary.haplo.ccs') obtains or prints a summary of the results, which include haplotype and covariate relative risks, robust standard error estimates, and estimated haplotype frequencies. The generic accessory functions 'coefficients', 'fitted.values', and

'residuals' extract corresponding features of the object returned by 'haplo.ccs'. The function 'vcov' (i.e., 'vcov.haplo.ccs') returns sandwich variance-covariance estimates. The function 'haplo.freq' extracts information returned by the EM computation of haplotype frequencies. Note that if rare haplotypes are grouped, then their individual estimated frequencies are summed. An object of class "'haplo.ccs'" is a list containing at least the following components:

formula	the formula supplied.
call	the matched call.
coefficients	a named vector of coefficients.
covariance	a named matrix of sandwich variance-covariance estimates, computed using 'sandcov'.
residuals	the working residuals, i.e., the residuals from the final iteration of the IWLS fit.
fitted.values	the fitted mean values, obtained by transforming the linear predictors by the expit function.
linear.predictors	the linear fit on the logit scale.
df	the model degrees of freedom.
rank	the numeric rank of the fitted model.
family	the family object, in this case, quasibinomial.
iter	the number of iterations of IWLS used.
weights	the working weights, i.e., the weights from the final iteration of the IWLS fit.
prior.weights	the weights initially supplied, in this case, the diplotype probabilities estimated by the EM computation.
y	a vector indicating case-control status, expanded for each subject by the number of plausible diplotypes for that subject.
id	the numeric vector used to identify subjects, expanded for each subject by the number of plausible diplotypes for that subject.
converged	a logical indicating whether the IWLS fit converged.
boundary	a logical indicating whether the fitted values are on the boundary of the attainable values.
model	the model matrix used.
terms	the terms object used.
offset	the offset vector used.
contrasts	the contrasts used.
xlevels	a record of the levels of the factors used in fitting.
inheritance.mode	the method of inheritance.
rare.freq	the value used to define the rare haplotypes.
em.lnlike	the value of the log likelihood at the last EM iteration.

<code>em.lr</code>	the likelihood ratio statistic used to test the assumed model against the model that assumes complete linkage equilibrium among all loci.
<code>em.df.lr</code>	the degrees of freedom for the likelihood ratio statistic.
<code>em.nreps</code>	the count of haplotype pairs that map to each subject's marker genotypes.
<code>hap1</code>	character strings representing the possible first haplotype for each subject.
<code>hap2</code>	character strings representing the possible second haplotype for each subject.
<code>hap.names</code>	character strings representing the unique haplotypes.
<code>hap.probs</code>	the estimated frequency of each unique haplotype. Note that if rare haplotypes are grouped, then their individual estimated frequencies are summed.
<code>em.converged</code>	a logical indicating whether the EM computation converged.
<code>em.nreps</code>	the number of haplotype pairs that map to the marker genotypes for each subject.
<code>em.max.pairs</code>	the maximum number of pairs of haplotypes per subject that are consistent with their marker data.
<code>em.control</code>	a list of control parameters for the EM computation.

Note

The functions 'anova', 'logLik', and 'AIC' are not appropriate for models of class "'haplo.ccs'", because 'haplo.ccs' does not fit by maximum likelihood. Accordingly, model and null deviance are not reported.

Author(s)

Benjamin French and Thomas Lumley, University of Washington

References

French B, Lumley T, Monks SA, Rice KM, Hindorff LA, Reiner AP, Psaty BM. Simple estimates of haplotype relative risks in case-control data. *Genetic Epidemiology* 2006; 30(6):485-494.

The help files for 'glm', 'haplo.em', and 'haplo.glm' were instrumental in creating this help file.

See Also

[glm](#), [haplo](#), [haplo.em](#), [haplo.glm](#), [sandcov](#)

Examples

```
data(renin)

## Fit a model for haplotype effects.

haplo.ccs(case ~ haplo(geno))

## Fit a model for haplotype and covariate effects.

haplo.ccs(case ~ gender + age + factor(race) + haplo(geno))
```

```
## Fit a model for haplotype interaction with gender.  
haplo.ccs(case ~ age + factor(race) + gender*haplo(geno))
```

`haplo.hpp`*Obtain Summaries of Phase Ambiguity*

Description

'haplo.hpp' obtains summary statistics of phase ambiguity. The proportion of subjects whose highest posterior diplotype probability is greater than or equal to a specified probability is reported.

Usage

```
haplo.hpp(model, prob=0.95)
```

Arguments

<code>model</code>	a fitted model of class "haplo.ccs".
<code>prob</code>	the probability to which to compare the highest posterior diplotype probability for each subject. Note that the default probability is 0.95. Note also that either a single probability or a list of probabilities may be specified.

Value

The proportion of subjects whose highest posterior diplotype probability is greater than or equal to the specified probability or probabilities.

Author(s)

Benjamin French and Thomas Lumley, University of Washington

References

French B, Lumley T, Monks SA, Rice KM, Hindorff LA, Reiner AP, Psaty BM. Simple estimates of haplotype relative risks in case-control data. *Genetic Epidemiology* 2006; 30(6):485-494.

See Also

[haplo](#), [haplo.ccs](#)

Examples

```
data(renin)

haplo.hpp(model=haplo.ccs(case ~ haplo(geno)), prob=c(0.90, 0.95))
```

renin

Example Dataset for 'haplo.ccs'

Description

This dataset serves as the example dataset for 'haplo.ccs'. The genotypes in this dataset were generated from haplotype frequency data for renin, one of the genes in the renin-angiotensin system.

Usage

```
data(renin)
```

Format

All variables are in numeric format.

Details

'case': case-control status (1=Case, 0=Control)

'geno': a matrix of alleles indicating genotype where each locus has a pair of adjacent columns of alleles, and the order of columns corresponds to the order of the loci on the chromosome (1=A, 2=C, 3=T, 4=G)

'gender': gender (1=Male, 2=Female)

'age': age in years

'race': race (1=White, 2=Black, 3=Asian, 4=Other)

Note

Other covariates, such as gender, age, and race were randomly generated. Therefore no scientific inference should be made from these data.

Author(s)

Benjamin French and Thomas Lumley, University of Washington

References

French B, Lumley T, Monks SA, Rice KM, Hindorff LA, Reiner AP, Psaty BM. Simple estimates of haplotype relative risks in case-control data. *Genetic Epidemiology* 2006; 30(6):485-494.

See Also

[haplo.ccs](#)

Examples

```
data(renin)
```

sandcov

Compute Sandwich Variance-Covariance Estimates

Description

'sandcov' computes sandwich variance-covariance estimates for the coefficients of a fitted model. These estimates may be used to calculate robust standard error estimates.

Usage

```
sandcov(model, id)
```

Arguments

model	a fitted model of class "lm" or "glm".
id	the numeric vector used to identify subjects, expanded for each subject by the number of observations for that subject.

Details

For a model of class "haplo.ccs", the sandwich variance-covariance matrix is automatically provided as the object 'covariance', or may be extracted by 'vcov' (i.e., 'vcov.haplo.ccs'). See examples below.

Value

A named matrix for the covariance of the regression coefficients specified in 'model', calculated using the sandwich method.

Author(s)

Benjamin French and Thomas Lumley, University of Washington

References

French B, Lumley T, Monks SA, Rice KM, Hindorff LA, Reiner AP, Psaty BM. Simple estimates of haplotype relative risks in case-control data. *Genetic Epidemiology* 2006; 30(6):485-494.

Liang KY, Zeger SL. Longitudinal data analysis using generalized linear models. *Biometrika* 1986; 73(1):13-22.

See Also

[glm](#), [haplo.ccs](#), [lm](#)

Examples

```
data(renin)

## Fit a model for covariate effects.

m1 <- glm(case ~ age + factor(race) + gender, family=binomial(link=logit))

## Obtain sandwich variance-covariance matrix.

id <- 1:length(case)
v1 <- sandcov(model = m1, id = id)

## Calculate robust standard error estimates.

se1 <- sqrt(diag(v1))

## Fit a model for haplotype and covariate effects.

m2 <- haplo.ccs(case ~ gender + age + factor(race) + haplo(geno))

## Obtain sandwich variance-covariance matrix by one of two methods.

v2 <- m2$covariance
v2 <- vcov(m2)

## Calculate robust standard error estimates.

se2 <- sqrt(diag(v2))
```

Index

*Topic **datasets**

renin, 7

*Topic **models**

haplo, 1

haplo.ccs, 3

haplo.hpp, 6

sandcov, 9

*Topic **regression**

haplo, 1

haplo.ccs, 3

haplo.hpp, 6

sandcov, 9

age(renin), 7

AIC.haplo.ccs(haplo.ccs), 3

anova.haplo.ccs(haplo.ccs), 3

case(renin), 7

coef.haplo.ccs(haplo.ccs), 3

count.haps(haplo.ccs), 3

fitted.haplo.ccs(haplo.ccs), 3

gender(renin), 7

geno(renin), 7

glm, 6, 9

haplo, 1, 6, 7

haplo.ccs, 2, 3, 7–9

haplo.em, 6

haplo.freq(haplo.ccs), 3

haplo.glm, 6

haplo.hpp, 6

lm, 9

logLik.haplo.ccs(haplo.ccs), 3

one(haplo.ccs), 3

print.haplo.ccs(haplo.ccs), 3

print.summary.haplo.ccs
(haplo.ccs), 3

race(renin), 7

renin, 7

residuals.haplo.ccs(haplo.ccs), 3

return.haps(haplo.ccs), 3

sandcov, 6, 9

summary.haplo.ccs(haplo.ccs), 3

vcov.haplo.ccs(haplo.ccs), 3