The hapassoc Package

September 23, 2007

Version 1.1-1

Title Likelihood inference of trait associations with SNP haplotypes and other attributes using the EM Algorithm

Author K. Burkett <kburkett@sfu.ca>, B. McNeney <mcneney@sfu.ca>, J. Graham <jgraham@stat.sfu.ca>

Maintainer K. Burkett <kburkett@sfu.ca>

Depends R (>= 2.0.0), stats

Description The following R functions are used for likelihood inference of trait associations with haplotypes and other covariates in generalized linear models. The functions accommodate uncertain haplotype phase and can handle missing genotypes at some SNPs.

License GPL version 2

URL http://stat-db.stat.sfu.ca:8080/statgen/research/hapassoc

R topics documented:

pre.hapassoc ........................................... 2
anova.hapassoc ...................................... 4
hapassoc ............................................. 5
hypoDat ............................................... 8
hypoDatGeno ......................................... 8
logLik.hapassoc .................................... 9
summary.hapassoc .................................. 10

Index 12
Description

This function takes as an argument a data frame with non-SNP and SNP data and converts the genotype data at single SNPs (the single-locus genotypes) into haplotype data. The rows of the input data frame should correspond to subjects. Single-locus SNP genotypes may be specified in one of two ways: (i) as pairs of columns, with one column for each allele of the single-locus genotypes (“allelic format”), or (ii) as columns of two-character genotypes (“genotypic format”). The SNP data should comprise the last 2*numSNPs columns (allelic format) or the last numSNPs columns (genotypic format) of the data frame.

If the haplotypes for a subject cannot be inferred from his or her genotype data, “pseudo-individuals” representing all possible haplotype combinations consistent with the single-locus genotypes are considered. Missing single-locus genotypes, up to a maximum of maxMissingGenos (see below), are allowed, but subjects with missing data in more than maxMissingGenos, or with missing non-SNP data, are removed. Initial estimates of haplotype frequencies are then obtained using the EM algorithm applied to the genotype data. Haplotypes with frequencies below a user-specified tolerance (zero.tol) are assumed not to exist and are removed from further consideration. (Pseudo-individuals having haplotypes of negligible frequency are deleted and the column in the design matrix corresponding to that haplotype is deleted.) For the remaining haplotypes, those with non-negligible frequency below a user-defined pooling tolerance (pooling.tol) are pooled into a single category called “pooled” in the design matrix for the risk model. However, the frequencies of each of these pooled haplotypes are still calculated separately.

Usage

pre.hapassoc(dat,numSNPs,maxMissingGenos=1,pooling.tol = 0.05,
zerotol = 1/(2 * nrow(dat) * 10), allelic=TRUE, verbose=TRUE)

Arguments

dat the non-SNP and SNP data as a data frame. The SNP data should comprise the last 2*numSNPs columns (allelic format) or last numSNPs columns (genotypic format). Missing allelic data should be coded as NA or "" and missing genotypic data should be coded as, e.g., "A" if one allele is missing and "" if both alleles are missing.

numSNPs number of SNPs per haplotype

maxMissingGenos maximum number of single-locus genotypes with missing data to allow for each subject. (Subjects with more missing data, or with missing non-SNP data are removed.) The default is 1.

pooling.tol pooling tolerance – by default set to 0.05

zerotol tolerance for haplotype frequencies below which haplotypes are assumed not to exist – by default set to $\frac{1}{2N+10}$ where N is the number of subjects
pre.hapassoc

allelic

TRUE if single-locus SNP genotypes are in allelic format and FALSE if in genotypic format; default is TRUE.

verbose

indicates whether or not a list of the genotype variables used to form haplotypes and a list of other non-genetic variables should be printed; default is TRUE.

Details

See the hapassoc vignette, of the same name as the package, for details.

Value

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>haplotest</td>
<td>logical, TRUE if some haplotypes were pooled in the risk model</td>
</tr>
<tr>
<td>initFreq</td>
<td>initial estimates of haplotype frequencies</td>
</tr>
<tr>
<td>zeroFreqHaplos</td>
<td>list of haplotypes assumed not to exist</td>
</tr>
<tr>
<td>pooledHaplos</td>
<td>list of haplotypes pooled into a single category in the design matrix</td>
</tr>
<tr>
<td>haploDM</td>
<td>Haplotype portion of the data frame augmented with pseudo-individuals. Has (2^{\text{numSNPs}}) columns scoring number of copies of each haplotype for each pseudo-individual</td>
</tr>
<tr>
<td>nonHaploDM</td>
<td>non-haplotype portion of the data frame augmented with pseudo-individuals</td>
</tr>
<tr>
<td>haploMat</td>
<td>matrix with 2 columns listing haplotype labels for each pseudo-individual</td>
</tr>
<tr>
<td>wt</td>
<td>vector giving initial weights for each pseudo-individual for the EM algorithm</td>
</tr>
<tr>
<td>ID</td>
<td>index for each individual in the original data frame. Note that all pseudo-individuals have the same ID value</td>
</tr>
</tbody>
</table>

References


See Also

hapassoc, summary.hapassoc.

Examples

#First example data set has single-locus genotypes in "allelic format"
data(hypoDat)
example.pre.hapassoc<-pre.hapassoc(hypoDat, numSNPs=3)

# To get the initial haplotype frequencies:
example.pre.hapassoc$initFreq

# h000    h001    h010    h011    h100    h101    h110
# 0.25179111 0.26050418 0.23606001 0.09164470 0.10133627 0.02636844 0.01081260
#  h111
The '001' haplotype is estimated to be the most frequent.

These haplotypes are to be pooled in the design matrix for the risk model.

The 'hAAC' haplotype is estimated to be the most frequent.

These haplotypes are to be pooled in the design matrix for the risk model.

---

### Description

This function returns the likelihood ratio test statistic comparing two nested models fit with `hapassoc`.

### Usage

```r
## S3 method for class 'hapassoc':
anova(object, redfit, display=TRUE, ...)
```

### Arguments

- `object`: a list of class `hapassoc` output by the `hapassoc` function
- `redfit`: A `hapassoc` object resulting from fitting a reduced model
- `display`: An indicator to suppress output displayed on screen
- `...`: additional arguments to the summary function currently unused
Details

See the hapassoc vignette, of the same name as the package, for details.

Value

- LRTstat: The likelihood ratio statistic comparing the two models
- df: Degrees of freedom of the likelihood ratio statistic
- pvalue: The p-value of the test

References


See Also

pre.hapassoc, hapassoc, summary.hapassoc.

Examples

data(hypoDatGeno)
example2.pre.hapassoc<-pre.hapassoc(hypoDatGeno, numSNPs=3, allelic=FALSE)
example2.regr <- hapassoc(affected ~ attr + hAAA+ hACA + hACC + hCAA + pooled, example2.pre.hapassoc, family=binomial())
example2.regr2 <- hapassoc(affected ~ attr + hAAA, example2.pre.hapassoc, family=binomial())
anova(example2.regr,example2.regr2)

# Returns:
# hapassoc: likelihood ratio test

#Full model: affected ~ attr + hAAA + hACA + hACC + hCAA + pooled
#Reduced model: affected ~ attr + hAAA

#LR statistic = 1.5433 , df = 4 , p-value = 0.8189

hapassoc

EM algorithm to fit maximum likelihood estimates of trait associations with SNP haplotypes

Description

This function takes a dataset of haplotypes in which rows for individuals of uncertain phase have been augmented by “pseudo-individuals” who carry the possible multilocus genotypes consistent with the single-locus phenotypes. The EM algorithm is used to find MLE’s for trait associations with covariates in generalized linear models.
Usage

hapassoc(form, haplos.list, baseline = "missing", family = binomial(), freq = NULL, maxit = 50, tol = 0.001, start = NULL, verbose=FALSE)

Arguments

form                       model equation in usual R format
haplos.list                list of haplotype data from pre.hapassoc
baseline                   optional, haplotype to be used for baseline coding. Default is the most frequent haplotype according to the initial haplotype frequency estimates returned by pre.hapassoc.
family                     binomial, poisson, gaussian or freq are supported, default=binomial
freq                       initial estimates of haplotype frequencies, default values are calculated in pre.hapassoc using standard haplotype-counting (i.e. EM algorithm without adjustment for non-haplotype covariates)
maxit                      maximum number of iterations of the EM algorithm; default=50
tol                        convergence tolerance in terms of either the maximum difference in parameter estimates between iterations or the maximum relative difference in parameter estimates between iterations, which ever is larger.
start                      starting values for parameter estimates in the risk model
verbose                    should the iteration number and value of the convergence criterion be printed at each iteration of the EM algorithm? Default=FALSE

Details

See the hapassoc vignette, of the same name as the package, for details.

Value

it                         number of iterations of the EM algorithm
beta                       estimated regression coefficients
freq                       estimated haplotype frequencies
fits                       fitted values of the trait
wts                        final weights calculated in last iteration of the EM algorithm. These are estimates of the conditional probabilities of each multilocus genotype given the observed single-locus genotypes.
var                        joint variance-covariance matrix of the estimated regression coefficients and the estimated haplotype frequencies
dispersionML               maximum likelihood estimate of dispersion parameter (to get the moment estimate, use summary.hapassoc)
family                     family of the generalized linear model (e.g. binomial, gaussian, etc.)
response                   trait value
converged                  TRUE/FALSE indicator of convergence. If the algorithm fails to converge, only the converged indicator is returned.
model equation

loglik the log-likelihood evaluated at the maximum likelihood estimates of all parameters

call the function call

References


See Also

pre.hapassoc, summary.hapassoc, glm, family.

Examples

data(hypoDat)
examp.pre.hapassoc<-pre.hapassoc(hypoDat, 3)

example.pre.hapassoc$initFreq # look at initial haplotype frequencies
# h000 h001 h010 h011 h100 h101 h110 h111
# 0.25179111 0.26050418 0.23606001 0.09164470 0.10133627 0.02636844 0.01081260
# 0.02148268

names(example.pre.hapassoc$haploDM)
# "h000" "h001" "h010" "h011" "h100" "pooled"

# Columns of the matrix haploDM score the number of copies of each haplotype
# for each pseudo-individual.

# Logistic regression for a multiplicative odds model having as the baseline
# group homozygotes '001/001' for the most common haplotype

example.regr <- hapassoc(affected ~ attr + h000 + h010 + h011 + h100 + pooled, example.pre.hapassoc, family=binomial())

# Logistic regression with separate effects for 000 homozygotes, 001 homozygotes
# and 000/001 heterozygotes

example2.regr <- hapassoc(affected ~ attr + I(h000==2) + I(h001==2) + I(h000==1 & h001==1), example.pre.hapassoc, family=binomial())
hypoDat

**Simulated data for a hypothetical binary trait**

**Description**

Simulated binary trait data used to illustrate the hapassoc package.

**Usage**

```r
data(hypoDat)
```

**Format**

Matrix with columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ,1]</td>
<td>numeric</td>
<td>affection status (1=yes, 0=no)</td>
</tr>
<tr>
<td>[ ,3]</td>
<td>numeric</td>
<td>simulated quantitative attribute</td>
</tr>
<tr>
<td>[ ,5]</td>
<td>numeric</td>
<td>the first allele of hypothetical SNP M1</td>
</tr>
<tr>
<td>[ ,6]</td>
<td>numeric</td>
<td>the second allele of hypothetical SNP M1</td>
</tr>
<tr>
<td>[ ,5]</td>
<td>numeric</td>
<td>the first allele of hypothetical SNP M2</td>
</tr>
<tr>
<td>[ ,6]</td>
<td>numeric</td>
<td>the second allele of hypothetical SNP M2</td>
</tr>
<tr>
<td>[ ,7]</td>
<td>numeric</td>
<td>the first allele of hypothetical SNP M3</td>
</tr>
<tr>
<td>[ ,8]</td>
<td>numeric</td>
<td>the second allele of hypothetical SNP M3</td>
</tr>
</tbody>
</table>

hypoDatGeno

**Simulated data for a hypothetical genetic SNPs**

**Description**

Simulated genetic SNPs data used to illustrate the hapassoc package.

**Usage**

```r
data(hypoDatGeno)
```

**Format**

Matrix with columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ,1]</td>
<td>numeric</td>
<td>affection status (1=yes, 0=no)</td>
</tr>
<tr>
<td>[ ,2]</td>
<td>numeric</td>
<td>simulated quantitative attribute</td>
</tr>
<tr>
<td>[ ,3]</td>
<td>numeric</td>
<td>hypothetical SNP M1</td>
</tr>
<tr>
<td>[ ,4]</td>
<td>numeric</td>
<td>hypothetical SNP M2</td>
</tr>
<tr>
<td>[ ,5]</td>
<td>numeric</td>
<td>hypothetical SNP M3</td>
</tr>
</tbody>
</table>
Description

This function is used to return the log-likelihood at the maximum likelihood estimates computed by `hapassoc` and to return the number of parameters fit by `hapassoc` (i.e. the degrees of freedom in R).

Usage

```r
## S3 method for class 'hapassoc':
logLik(object, ...)
```

Arguments

- `object` a list of class `hapassoc` output by the `hapassoc` function
- `...` additional arguments to the summary function (currently unused)

Details

See the hapassoc vignette, of the same name as the package, for details.

Value

- `logLik` log-likelihood computed at the maximum likelihood estimates
- `df` number of parameters in the model (i.e. regression coefficients, any dispersion parameters and haplotype frequencies). This is not the residual degrees of freedom, which is the number of subjects minus the number of parameters estimated.

References


See Also

`pre.hapassoc`, `hapassoc`, `summary.hapassoc`. 
Examples

data(hypoDatGeno)
example2.pre.hapassoc<-pre.hapassoc(hypoDatGeno, numSNPs=3, allelic=FALSE)
example.regr <- hapassoc(affected ~ attr + hAAA + hACA + hACC + hCAA + pooled, example2.pre.hapassoc, family=binomial())
logLik(example.regr)

# Returns:
# Log Lik: -322.1558 (df=14)

summary.hapassoc  

Summary function for reporting the results of the hapassoc function in a similar style to the lm and glm summaries.

Usage

## S3 method for class 'hapassoc':
summary(object, ...)

Arguments

object  
a list of class hapassoc output by the hapassoc function

...  
additional arguments to the summary function (currently unused)

Details

See the hapassoc vignette, of the same name as the package, for details.

Value

call  
The function call to hapassoc

subjects  
The number of subjects used in the analysis

coefficients  
Table of estimated coefficients, standard errors and Wald tests for each variable

frequencies  
Table of estimated haplotype frequencies and standard errors

dispersion  
Estimate of dispersion parameter (Moment estimator for gamma model)

References


### See Also

`pre.hapassoc`, `hapassoc`.

### Examples

```r
data(hypoDat)
exh <- pre.hapassoc(hypoDat, 3)
exh.regr <- hapassoc(affected ~ attr + h000 + h010 + h011 + h100 + pooled, exh, family=binomial())

# Summarize the results:
summary.hapassoc(exh.regr) # or just summary(exh.regr)

# Results:
#$coefficients
#  Estimate Std. Error z score  Pr(>|z|)
#(Intercept)  -1.24114 0.7820977  -1.587 0.11252606
#attr          0.74037 0.2918205   2.537 0.01117844
#h000       1.14968 0.5942542   1.935 0.05303126
#h010      -0.59319 0.6569672  -0.903 0.36657201
#h011     -0.03615 0.9161959  -0.039 0.96852422
#h100      -0.85329 1.0203105  -0.836 0.40298217
#pooled    0.38517 0.8784283   0.438 0.66104215

#$frequencies
#  Estimate Std. Error
#f.h000   0.2671639 0.03933158
#f.h001   0.2519167 0.03866739
#f.h010   0.2199713 0.03881578
#f.h011   0.1994795 0.02949617
#f.h100   0.0950701 0.02371878
#f.h101   0.0258491 0.01411881
#f.h110   0.0177945 0.01386080
#f.h111   0.0212861 0.01247265

#$dispersion
#[1] 1
```
Index

*T Topic datasets
  hypoDat, 7
  hypoDatGeno, 8

*T Topic methods
  anova.hapassoc, 4
  hapassoc, 5
  logLik.hapassoc, 8
  pre.hapassoc, 1
  summary.hapassoc, 9

anova.hapassoc, 4
family, 7
glm, 7

hapassoc, 3, 4, 5, 5, 9, 10
hypoDat, 7
hypoDatGeno, 8

logLik.hapassoc, 8
pre.hapassoc, 1, 5–7, 9, 10
summary.hapassoc, 3, 5–7, 9, 9