Family Based Association Tests Using the fbat package

Weiliang Qiu
email: weiliang.qiu@gmail.com
Ross Lazarus
email: ross.lazarus@channing.harvard.edu

April 25, 2007

Contents

1 Introduction

The R package fbat can be used to test the following null hypotheses for each marker based on family pedigrees:

H_{01} : the marker has no association and no linkage with the trait;
H_{02} : the marker has no association with the trait in the presence of linkage.

We assume that

- the families are **nuclear** families
- there are no missing genotypes and phenotypes for children

- markers are bi-allelic.

A more general software `FBAT` is available as a stand-alone executable with documentation and example files from `http://www.biostat.harvard.edu/~fbat/fbat.htm`. While this `R` package has some important limitations as present, these will be addressed in further versions.

2 Pedigree data file format

All fields are separated by whitespace (e.g. one or more spaces).

First line : names of all markers in the sequence of the genotype data. For example, marker_1, marker_2, ..., marker_m.

Remaining lines: The remaining lines contain only non-negative integers and have the same format:

family	pid	father	mother	sex	affection	$\text{marker}_{1.1}$	$\text{marker}_{1.2}$	$\cdots$	$\text{marker}_{m.1}$	$\text{marker}_{m.2}$

where

family: family id

pid: patient id

father: father id.

Use 0 (zero) for founders or marry-ins (parents not specified) in a pedigree. A **founder** in a pedigree is an individual who is not a child of any individuals in the pedigree.

mother: mother id.

Use 0 (zero) for founders or marry-ins (parents not specified) in a pedigree. A **founder** in a pedigree is an individual who is not a child of any individuals in the pedigree.

sex: 1 – male; 2 – female;

affection: affection status (i.e., trait)

2 – affected; 1 – unaffected; 0 – unknown

$\text{marker}_{i.j}$: allele j of marker i, $j = 1, 2$; $i = 1, 2, \ldots, m$.

non-missing Alleles are represented by positive integers. Missing alleles are represented by zero (0).

3 Examples

To call the functions in the R package fbat, we first need to load it into R:

To read the pedigree file CAMP.ped into R, we use the function readGenes.ped in the R package GeneticsBase:

```
library(GeneticsBase)
gSet<-readGenes.ped(filename="CAMP.ped",
                    columns = c("family", "pid", "father", "mother",
                                "sex", "affection"))
```

The function readGenes.ped returns back an object of the R class *geneSet*.

Before we apply family based association tests, it would be good practice to check Hardy-Weinberg equilibrium for each marker based on parental data. We can use the function pedHardyWeinberg to do this.

```
> data(CAMP)

Reading 8 markers and 2011 subjects from ` CAMP.ped ' ...
generating 'geneSet' object...

Successfully read the pedigree file ` CAMP.ped '.

Number of Markers:  8
Number of Subjects: 2011
Number of Families: 651

> ch <- pedHardyWeinberg(CAMP)

     nInfoInd nGenotype nHET nHOM nAllele nMissing  chi2 df p-value
m709     1263         3    4 1259       2       40 0.003  1   0.955
m654     1256         3  546  710       2       47 4.532  1   0.033
m47      1241         3  557  684       2       62 1.669  1   0.196
p46      1249         3  577  672       2       54 1.939  1   0.164
p79      1237         3  545  692       2       66 4.064  1   0.044
p252     1171         3  391  780       2      132 2.393  1   0.122
p491     1259         3   28 1231       2       44 0.159  1   0.690
p523     1275         3  399  876       2       28 0.542  1   0.462
```

The column nInfoInd means the number of informative individuals, i.e. individuals whose genotypes contain no missing alleles for the specified marker; the column nGenotype means number of possible genotypes; the column nHET means number of heterozygous genotypes; the column nHOM means number of homozygous genotypes; the column nAllele means number of alleles; the column nMissing means number of missing alleles; the column chi2 means chi square test statistic; the column means df means

degree of freedom of the chi square test statistic under the null hypothesis that Hardy-Weinberg condition holds; and the column `p-value` means pvalue of the test.

To view the statistics for individual markers, we can use the function `viewHW`. For example,

```
> viewHW(ch, "p79")

number of possible genotypes for marker  p79  >>
[1] 3
genotype frequency >>
     p79.1 p79.2 freq
[1,]     1     1  504
[2,]     1     2  545
[3,]     2     2  188
allele frequency >>
    1     2
0.628 0.372
 nInfoInd nGenotype      nHET      nHOM   nAllele  nMissing      chi2        df
 1237.000     3.000   545.000   692.000     2.000    66.000     4.064     1.000
  p-value
    0.044
```

To get the family based association test statistics, we use the function `fbat`:

```
> res <- fbat(CAMP)

converting geneSet object to numerical matrix...
fbating...
```

The usage of the function `fbat` is

```
fbat(geneSetObject, model="a", traitMethod=3, traitOffset=0, quiet=TRUE)
```

The function argument `model` specifies the genotype codings.

By default, we use the additive model (`model`="a"). Other available models include dominant (`model`="d"), recessive (`model`="r"), and genotype (`model`="g") models.

The function argument `traitMethod` indicates the trait coding method. If `traitMethod` is equal to 1, then the trait is represented by `trait-offset` where `trait` is the sixth column (i.e., `affection` status) of the pedigree matrix and the value of `offset` is provided by the argument `traitOffset`. If the argument `traitMethod` takes value other than 1, then the trait is set to be 1 if the sixth column of the pedigree matrix takes value 2 and the trait is set to be 0 if the sixth column of the pedigree matrix takes value 1.

The function `fbat` returns a list. To summarize the values, degrees of freedom, and p-values of the test statistics for the markers, we can use the function `summaryPvalue`:

```
> summaryPvalue(res)

**************************
          chisq rank    pvalue
m709 1.80000000    1 0.1797125
m654 0.08828829    1 0.7663646
m47  0.02846975    1 0.8660092
p46  0.16835017    1 0.6815822
p79  0.14808044    1 0.7003766
p252 1.24225352    1 0.2650372
p491 0.53333333    1 0.4652088
p523 2.19512195    1 0.1384483
**************************
```

To adjust multiple comparisons, we can use the function `p.adjust` in the R package base to adjust the p-values. For example,

```
> pvals <- res$statPvalue[, 3]
> p.adjust.M <- p.adjust.methods
> p.adj <- sapply(p.adjust.M, function(meth) p.adjust(pvals, meth))
> noquote(apply(p.adj, 2, format.pval, digits = 3))

     holm hochberg hommel bonferroni BH    BY fdr   none
[1,] 1    0.866    0.866  1          0.707 1  0.707 0.180
[2,] 1    0.866    0.866  1          0.866 1  0.866 0.766
[3,] 1    0.866    0.866  1          0.866 1  0.866 0.866
[4,] 1    0.866    0.866  1          0.866 1  0.866 0.682
[5,] 1    0.866    0.866  1          0.866 1  0.866 0.700
[6,] 1    0.866    0.866  1          0.707 1  0.707 0.265
[7,] 1    0.866    0.866  1          0.866 1  0.866 0.465
[8,] 1    0.866    0.866  1          0.707 1  0.707 0.138
```

To view summary statistics of individual marker, we can use the function `viewstat`. For example,

```
> viewstat(res, "p79")

****************************************
651  pedigree  2011  persons
393  informative families at marker  p79
The alleles of marker  p79  >>
[1] 1 2
Score for marker  p79  >>
[1] 473 363
```

```
Expected score for marker  p79  >>
[1] 477.5 358.5
Covariance matrix of the score for marker  p79  >>
        [,1]    [,2]
[1,]  136.75 -136.75
[2,] -136.75  136.75
Moore-Penrose generalized inverse of covariance matrix
             [,1]         [,2]
[1,]  0.001828154 -0.001828154
[2,] -0.001828154  0.001828154
test statistics for marker  p79  >>
    chisq      rank    pvalue
0.1480804 1.0000000 0.7003766
****************************************
```

Note that if the covariance matrix of the S score vector is singular, the Moore-Penrose generalized inverse is used.

Sometimes the user might want to know if a genotype a homozygous or heterozygous. The function pedFlagHomo can provide those information. For example,

```
> res.f <- pedFlagHomo(CAMP)

dim(flagHomoMat)= 1303 8
length(ped[,2])= 1303
numHomo -- number of homozygous genotypes
numHetero -- number of homozygous genotypes
numMiss1 -- number of genotypes containing one missing allele
numMiss2 -- number of genotypes containing two missing alleles
counts>>>
     numHomo numHetero numMiss1 numMiss2
m709    1259         4        0       40
m654     710       546        0       47
m47      684       557        0       62
p46      672       577        0       54
p79      692       545        0       66
p252     780       391        0      132
p491    1231        28        0       44
p523     876       399        0       28
```

The function pedGFreq gets genotype frequencies and percentages. For example,

```
> res <- pedGFreq(CAMP)
```

```
genotype counts>>>
      00 01 02   11  12  22
m709  40  0  0 1259   4   0
m654  47  0  0  527 546 183
m47   62  0  0  180 557 504
p46   54  0  0  214 577 458
p79   66  0  0  504 545 188
p252 132  0  0   69 391 711
p491  44  0  0 1231  28   0
p523  28  0  0  821 399  55
```

The function `pedAFreq` gets allele frequencies and percentages. For example,

```
> res <- pedAFreq(CAMP)

allele frequencies and percentages>>>
       0    1    2     0     1     2
m709  80 2522    4 0.031 0.968 0.002
m654  94 1600  912 0.036 0.614 0.350
m47  124  917 1565 0.048 0.352 0.601
p46  108 1005 1493 0.041 0.386 0.573
p79  132 1553  921 0.051 0.596 0.353
p252 264  529 1813 0.101 0.203 0.696
p491  88 2490   28 0.034 0.955 0.011
p523  56 2041  509 0.021 0.783 0.195
```

The package `fbat` also provides a function `geneSet2Ped` to convert a *geneSet* object to a pedigree matrix. The functions `fbat`, `pedHardyWeinberg`, `pedFlagHomo`, `pedGFreq`, and `pedAFreq` have default forms (`fbat.default`, `pedHardyWeinberg.default`, `pedFlagHomo.default`, `pedGFreq.default`, and `pedAFreq.default`) that use a pedigree matrix as input.

Appendix

A Notation

For a given marker,

- Y_{ij} — Observed trait of the j-th offspring in family i.
- T_{ij} — A function of Y_{ij}.

  $$T_{ij} = T(Y_{ij}).$$

  For example

  $$T_{ij} = T(Y_{ij}) = Y_{ij} - \mu_{ij},$$

  where μ_{ij} is an offset.

- g_{ij} — Genotype of the j-th offspring in family i;
- X_{ij} — A function of g_{ij}.
$$X_{ij} = X(g_{ij}).$$
- S score:
$$S = \sum_{ij} T_{ij} X_{ij} = \sum_{ij} T(Y_{ij}) X(g_{ij}).$$
- test statistic:
$$U = S - \mathrm{E}[S|H_0, \mathcal{C}],$$
where $\mathcal{C}$ is a condition set. When parental genotypes are complete, the condition set $\mathcal{C} = \mathcal{T} \cup \mathcal{G}$, where $\mathcal{T}$ is the observed traits in all family members and $\mathcal{G}$ is the parental genotypes. When parental genotypes are incomplete, the condition set $\mathcal{C} = \mathcal{T} \cup \mathcal{G}^* \cup \mathcal{G}_{\text{offspring}}$, $\mathcal{G}^*$ is the partially observed parental genotypes and $\mathcal{G}_{\text{offspring}}$ is the set of offspring genotypes (i.e., the offspring genotype configuration).
- V – variance or covariance matrix of U under the null hypothesis H_0. I.e.,
$$V = \mathrm{Cov}(U|H_0, \mathcal{C}) = \mathrm{Cov}\,(S|H_0, \mathcal{C}).$$
- For the univariate case,
$$Z = \frac{U}{\sqrt{V}}\Bigg| H_0, \mathcal{C} \dot{\to} \mathrm{N}\,(0, 1).$$
- For the multivariate case,
$$\chi^2 = U' V^{-1} U \big| H_0, \mathcal{C} \dot{\to} \chi^2_r,$$
where $r = \mathrm{rank}\,(V)$.

B Genotype coding methods

Denote K as the number of all possible different alleles for the locus and X as the vector of genotype coding.

GEN X is a vector with length equal to the number of genotypes that are possible given the parental genotypes in the sample, a maximum of $K(K+1)/2$ genotypes, and with elements equal to 1 or 0 to indicate which of the possible genotypes is equal to the genotype g.

GDOM codes the jth element of the vector X as $x_j = 1$ if genotype g has one or two alleles of type j, otherwise $x_j = 0$. X is a vector of length K.

GREC codes the jth element of the vector X as $x_j = 1$ if genotype g has two alleles of type j, otherwise $x_j = 0$. X is a vector of length K.

GTDT scores the number of alleles of a particular type by coding x_j equal to the number of alleles of type j in the genotype g (i.e., $x_j = 0, 1$, or 2 if g has 0, 1 or 2 alleles of type j). X is a vector of length K.

2-allele case

Example of different marker codings for a marker with $K = 2$ alleles, see Schaid (1996)

genotype	$X(g)$			
g	GEN	GDOM	GREC	GTDT
		(A, a)	(A, a)	(A, a)
AA	(0,0,0)	(1,0)	(1,0)	(2,0)
Aa	(1,0,0)	(1,1)	(0,0)	(1,1)
aa	(0,1,0)	(0,1)	(0,1)	(0,2)

3-allele case

Example of different marker codings for a marker with $K = 3$ alleles, see Schaid (1996) (This table is Table 4 of Horvath et al.'s report for FBAT software)

genotype	$X(g)$			
g	GEN	GDOM	GREC	GTDT
		(A, B, C)	(A, B, C)	(A, B, C)
AA	(0,0,0,0,0)	(1,0,0)	(1,0,0)	(2,0,0)
AB	(1,0,0,0,0)	(1,1,0)	(0,0,0)	(1,1,0)
AC	(0,1,0,0,0)	(1,0,1)	(0,0,0)	(1,0,1)
BB	(0,0,1,0,0)	(0,1,0)	(0,1,0)	(0,2,0)
BC	(0,0,0,1,0)	(0,1,1)	(0,0,0)	(0,1,1)
CC	(0,0,0,0,1)	(0,0,1)	(0,0,1)	(0,0,2)

C Trait coding methods

Denote Y_{ij} as the trait of the j-th child of the ith nuclear family. Y_{ij} can be dichotomous, measured (i.e., continuous?), time-to-onset (i.e., censored?)

The trait coding methods ($T_{ij} = T(Y_{ij})$) are listed below:

- $T_{ij} = 1$ if the jth child is affected; $T_{ij} = 0$ otherwise.

- $T_{ij} = Y_{ij} - \mu_{ij}$, where μ_{ij} is an offset.
- $T_{ij} = Y_{ij} - \mu_{ij}(\boldsymbol{x}'\boldsymbol{\beta})$, where $\mathrm{E}(Y_{ij}|\boldsymbol{x}) = \mu_{ij}(\boldsymbol{x}'\boldsymbol{\beta})$, and $\boldsymbol{x}$ are design matrix of covariates, $\boldsymbol{\beta}$ are unknown parameters.