The Stata Journal (2002)
2, Number 1, pp. 65–70

Analysis of quantitative traits using regression and log-linear modeling when phase is unknown

A. P. Mander
MRC Biostatistics Unit, Cambridge, UK
adrian.p.mander@gsk.com

Abstract. This function models the relationship between quantitative trait and the genotype of a person. It introduces a new syntax for model specification, which is necessary because when phase is unknown, the explanatory variables for the linear regression are never observed. The data must be a population-based sample because within family effects are not modeled.

Keywords: st0008, haplotype analysis, association studies, phase-unknown, linear regression

1 Syntax

```
qhapipf varlist [using filename] [if exp]  , qt(varname) [ ipf(string)
   regress(string) start display known phase(varname) acc(#) ipfacc(#)
   nolog model(#) lrtest(#,#) convars(varlist) confile(filename) mv
   mvdel hap(string) menu ]
```

To execute the menu interface version of this command type

```
. qhapipf,menu
```

2 Description

This function models the relationship between a normally distributed continuous variable in a population-based random sample and individuals' haplotype. This function uses an EM algorithm to resolve haplotype phase. Covariates are constructed from the haplotype and used in a regression model. Additionally, the EM algorithm handles missing typings assuming MCAR.

There are two distinct models in the log-linear model for haplotype frequencies. Further details of this procedure are found in the Stata function `hapipf` Mander (2001a,c). Haplotype frequencies are estimated under the assumption of Hardy–Weinberg Equilibrium.

The regression model relates the haplotypes to the quantitative trait. This model is specified in `regress()`, with the dependent variable specified by the `qt()` option.

The regression model takes a syntax to specify the dummy variables for the regression model. The syntax can specify within-loci, between-loci, and between-chromosome effects. The theory behind the method is covered in Mander (2001b).

3 Options

`qt(`*varname*`)` specifies the dependent variable in the regression model.

`ipf(`*string*`)` specifies the log-linear model. It requires syntax of the form `l1*l2+l3`. `l1*l2` allows all the interactions between the first two loci, and locus 3 is independent of them. This syntax is used in most books on log-linear modeling.

`regress(`*string*`)` specifies the regression model. The program then creates "dummy" variables for all the effects.

`start` specifies that the starting posterior weights of the EM algorithm are chosen at random.

`display` specifies whether to output parameter estimates.

`known` specifies that phase is known.

`phase(`*varname*`)` specifies a variable that contains 1s where phase is known and 0s where phase is unknown.

`acc(#)` specifies the convergence criteria based on the log likelihood.

`ipfacc(#)` specifies the convergence criteria for the ipf algorithm.

`nolog` suppresses the iteration log.

`model(#)` specifies a label for the log-linear model being fitted. This label is used in the `lrtest()` option.

`lrtest(#,#)` performs a likelihood-ratio test.

`convars(`*varlist*`)` specifies a list of variables in the constraints file.

`confile(`*filename*`)` specifies the name of the constraints file.

`mv` specifies that the algorithm should replace missing locus data (".") with a copy of each of the possible alleles at this locus. This is performed at the same stage as the handling of the missing phase when the dataset is expanded into all possible observations. If this option is not specified but some of the alleles do contain missing data, the algorithm sees the symbol "." as another allele.

`mvdel` specifies that all subjects with missing alleles are deleted.

`hap(`*string*`)` specifies the haplotype of interest. The dummy variables in the regression are all related to this haplotype. If the user does not select a particular haplotype, one is chosen.

`menu` specifies that the command is run through a windowed interface.

4 Regression model syntax

For the examples, I shall assume that there are three loci a, b, and c. The pairs of alleles are contained in the 6 variables a1, a2, b1, b2, c1, and c2. Let the quantitative trait variable be y.

All the models described here assume that the saturated model is fitted for the haplotype frequencies. For a single locus `a`, this saturated model is specified by the option `ipf(l1)`. Given this, the regression models are specified in the `regress()` option, and the more common models are described below. All the regression models assume that there are two alleles per locus, multiple alleles are recoded by the algorithm in terms of an allele of interest, and all the rest are the reference group.

The one parameter constant model is specified by `reg(1)`. To add an additional parameter that is the additive effect of the allele of interest, the model is specified by the option `reg([l1+l1])`, where `l1` represents the first locus in the `varlist`. This is the one-locus single-point additive model (one-locus SAM). The terms between the `[ ]` brackets represent the within locus model, and in the SAM the two chromosomes are independent but have the same parameter for the allele of interest effect. If the allelic effect depended on the chromosome, then there would be two parameters, and this is specified by the option `reg([l1a+l1b])`. The effect of parental imprinting is not additive. Additionally, the within-locus between-chromosome interaction can be included by replacing the + symbol with *. This parameter is usually called the dominance parameter. The two models become `reg([l1*l1])` and `reg([l1a*l1b])`, respectively.

The commands to fit these models are given below.

```
. qhapipf a1 a2, ipf(l1) reg(1) qt(y)
. qhapipf a1 a2, ipf(l1) reg([l1+l1]) qt(y)
. qhapipf a1 a2, ipf(l1) reg([l1a+l1b]) qt(y)
. qhapipf a1 a2, ipf(l1) reg([l1*l1]) qt(y)
. qhapipf a1 a2, ipf(l1) reg([l1a*l1b]) qt(y)
```

To test whether locus a is associated with the quantitative trait, compare the regression models `1` and `[l1+l1]`

```
. qhapipf a1 a2, ipf(l1) reg([l1+l1]) model(0) qt(y)
. qhapipf a1 a2, ipf(l1) reg(1) model(1) lrtest(0,1) qt(y)
```

When modeling more than one locus, there are additional between-loci interaction terms. The within-loci interactions are specified within the `[ ]` brackets, and the between-loci interactions are specified between the `[ ]` brackets. The two-locus SAM now becomes the model `[l1+l1]+[l2+l2]`, where the two loci are independent and are specified by the "+" symbol between the two sets of brackets. An extension of this model would allow one between-loci interaction (or "haplotype" effect); this is the two-locus multipoint additive model (two-locus MAM), and is specified by the option `reg([l1+l1]*[l2+l2])`. The saturated model that ignores parental imprinting is specified by the option `reg([l1*l1]*[l2*l2])`. This model contains between-chromosome interactions. Between-chromosome interactions can be further divided into within-loci

between-chromosome interactions (dominance parameters) and between-loci between-chromosome interactions. The full saturated model including parental imprinting is specified by the option `reg([l1a*l1b]*[l2a*l2b])`.

The commands to fit these models are given below:

```
. qhapipf a1 a2 b1 b2, ipf(l1*l2) reg([l1+l1]+[l2+l2]) qt(y)
. qhapipf a1 a2 b1 b2, ipf(l1*l2) reg([l1+l1]*[l2+l2]) qt(y)
. qhapipf a1 a2 b1 b2, ipf(l1*l2) reg([l1*l1]*[l2*l2]) qt(y)
. qhapipf a1 a2 b1 b2, ipf(l1*l2) reg([l1a*l1b]*[l2a*l2b]) qt(y)
```

The algorithm calculates the haplotype frequencies internally, and the log-linear model option `ipf()` specifies this model. Generally, it is taken to be the saturated model. It may be advantageous to use an intermediate model to reduce the number of parameters in the full joint likelihood. This can also be tested using this command using the likelihood-ratio test.

5 Output

The output from fitting a saturated model on two diallelic loci is given below. There are 16 genotypes, and so there are 16 parameters in the regression model

```
.   qhapipf a1 a2 b1 b2, ipf(l1*l2) reg([l1a*l1b]*[l2a*l2b]) qt(y)

Marker information
------------------

Alleles for l1 are (a1 , a2)
Alleles for l2 are (b1 , b2)

Creating dummy variables
------------------------

The reference haplotype is NOT specified : It will be 2.2
Iteration 1 loglhd = -1310.455115085673
Iteration 2 loglhd = -1309.893539304069
Iteration 3 loglhd = -1309.728576154266
Iteration 4 loglhd = -1309.68123457476
Iteration 5 loglhd = -1309.66781267106
Iteration 6 loglhd = -1309.664049684015
Iteration 7 loglhd = -1309.662986877095
Iteration 8 loglhd = -1309.66271802795
Iteration 9 loglhd = -1309.662636620017
```

(*Continued on next page*)

```
Regression Parameters
```

```
Residual standard deviation is   0.5710
Standard errors are calculated conditional on the weights
```

Var	Coef	SE	Parameter
_d0	1.1452	0.1058	Constant
_d1	1.7256	0.1224	l1a
_d2	2.1411	0.1224	l1b
_d3	-0.0962	0.1224	l1a~=l1b
_d4	2.9005	0.1107	l2a
_d5	2.9005	0.1107	l2b
_d6	-0.2074	0.1107	l2a~=l2b
_d7	0.1229	0.1035	l2a.l1a
_d8	-0.0468	0.1035	l2a.l1b
_d9	0.0873	0.1035	l2a.l1a~=l1b
_d10	0.1229	0.1035	l2b.l1a
_d11	-0.0468	0.1035	l2b.l1b
_d12	0.0873	0.1035	l2b.l1a~=l1b
_d13	0.2825	0.1035	l2a~=l2b.l1a
_d14	-0.0041	0.1035	l2a~=l2b.l1b
_d15	0.0100	0.1035	l2a~=l2b.l1a~=l1b

```
 ~= symbol represents dominance parameters
 &  symbol represents additive parameters.
Quantitative-Haplotype Estimation by EM algorithm
 No. loci          = 2
 Log-Likelihood    = -1309.662636620017
 Tot. parameters   = 20
 No. cells      (from log-linear) = 4
 No. parameters (from log-linear) = 4
 No. parameters (from regression) = 16
```

Note that the parameters of the regression model are specified by dominance terms of the form `l1a~=l1b` and interactions are specified by the "." symbol. Additive terms are indicated by the "&" symbol, so an additive term for the first locus would be `l1&l1`. The standard errors displayed are obtained from the linear regression model conditional on the haplotype frequencies and do not reflect the full model variability. Therefore, they are labeled naive.

6 References

Mander, A. P. 2001a. Haplotype analysis in population-based association studies. *Stata Journal (1)* 1: 58–75.

—. 2001b. Population-based quantitative trait haplotype analysis using an algorithm to resolve phase. Submitted.

—. 2001c. sbe38: Haplotype frequency estimation using an EM algorithm and log-linear modeling. *Stata Technical Bulletin* 57: 5–7. vol. 10, 104–107. College Station, TX: Stata Press.

About the Author

Adrian Mander has recently joined GSK to work in the Clinical Pharmacology Statistics and Data Sciences group after having worked in the MRC biostatistics unit, Cambridge for the last four years. He will continue working in genetics and mainstream statistics.