Estimating Genetic Components of Variance for Quantitative Traits in Family Studies using the MULTIC routines

Mariza de Andrade
Elizabeth J. Atkinson
Eric Lunde
Christopher I. Amos *
Jianfang Chen *

June 9, 2006

Technical Report #78

* Department of Epidemiology, U.T. MD Anderson Cancer Center Houston, Texas

Contents

1 Introduction

One of the primary goals of human genetics research is the elucidation of the underlying genetic architecture of complex disorders or phenotypes such as hypertension. Hypertension is a trait that relies on the underlying quantitative measure of systolic blood pressure ($\geq$ 140 mmg) and/or diastolic blood pressure ($\geq$ 90 mmg). In genetic studies the quantitative measure such as systolic blood pressure of a complex disorder (hypertension) will provide more information than its categorized measure (e.g., yes/no). Thus, our goal is the linkage analysis of quantitative phenotypes or traits.

In contrast to simple Mendelian disorders that result from a mutation in a single gene (Huntington's Disease), complex phenotypes are the product of the action of multiple genes and environmental factors as well as their interactions. If genetic factors influence inter-individual variability of a trait, identifying the specific factors involved is a major goal of most current genetic studies. Although some environmental factors can be observed directly and modeled as fixed effects, genetic effects are typically unobservable. To identify the total proportion of variance that can be attributed to genetic factors, mixed linear models, path analysis, and other methodologies have been developed and applied, as discussed by Thompson and Shaw [47] and Rao et al. [39]. Although effects from specific alleles at identified genetic loci are usually not available for the study of quantitative traits, a dense map of microsatellite markers and single nucleotide polymorphisms (SNP) is currently available for human genome-wide linkage analysis. These markers usually do not have a direct effect on trait variability, but if they are linked, co-segregation of the linked markers with a trait locus can be used to partition inter-individual variability into linked and unlinked components of variance [6].

Some environmental effects can be attributed to known and measurable factors, and a fixed effects model is appropriate to represent these sources of variability. Other sources of variability, such as measurement error, reflect an inherently variable process and are specified by a random effects model. In some cases unmeasurable environmental effects due to factors such as shared household effect can be modeled as a random effect. Genetic sources of variability can be modeled as either fixed or random components of variance. Observed effects from specific alleles at a locus, which are believed to directly affect trait variability, are modeled as fixed effects. Polygenic effects, which arise from effects of many unlinked loci with unmeasurably small effects, induce a correlation structure among relatives [26] and are modeled as a random effect. The effect from a major locus, which by definition has an estimably large effect, can be modeled as a fixed or random component. If the number of alleles at a locus is known, the unobservable genetic effects from the locus could be modeled as a fixed effect. Standard models of qualitative traits, such as being affected by a disease, assume a simple biallelic genetic model. For many of the quantitative traits that have been characterized at the molecular level, such as Lp(a), α1 - antitrypsin, and galactosemia, this assumption seems to be invalid, because a large number of variant alleles are typically observed in those cases. Thus, a random effects model may be more appropriate for assessing the effects from a genetic factor linked to a marker. If the genetic effect actually results from a single locus that has two alleles, specifying a random effects model has been shown in simulation studies to lead to unbiased estimates of the variance components provided the sample size is large enough and families are not selected through extreme individuals [9].

For the preliminary evaluation of genetic linkage using quantitative traits, numerous strategies have been developed. The Haseman-Elston (H-E) method is the simplest of these [27]. In this procedure, the squared difference between trait values from two siblings is regressed upon the estimated proportion of genes "identical by descent" (IBD) at a marker locus. In the absence

of linkage, there is no relationship between IBD at the marker locus and the squared pair differences. In the presence of linkage, pairs of sibs who share two genes IBD are concordant for the genetic factor that influences variation in trait levels. Therefore, they show a smaller squared pair difference than pairs that share no alleles IBD. Thus, a simple test for linkage can be developed by testing for a negative regression of squared pair differences onto IBD. The H-E method is easy to apply, has been shown to detect linkage in the presence of non-normally distributed residual non-genetic variance [11], and has been extended to provide a test for linkage using multivariate data [8]. Recently, the H-E method was revisited to incorporate the trait covariance between the two sibling pairs [25]. However, even the "revisited" method, which uses the cross-product–rather than the squared difference–in sib trait values, is, in general, less powerful than VC models [44]. To provide a simpler framework for modeling data from sibships and extended families and for inclusion of covariates, Hopper and Mathews [28] and Amos [6] suggested the use of a components of variance approach. The obvious approach of simply applying standard likelihood theory and assuming a multivariate normal distribution to model the residual distribution of the phenotypes (after conditioning on fixed effects such as measurable covariates) in families may not be appropriate if unobservable major genes are segregating, because this segregation introduces platykurtosis and may skew the distribution of trait values. Extensive simulations by Amos et al. [9] failed to document significant bias attributable to the moderate kurtosis introduced by major gene effects, but robust estimation procedures were preferable when the nongenetic variation was markedly non-normal.

For correlated traits, such as those predicting cardiovascular disease risk, multivariate approaches for genetic linkage can increase the power and precision of estimators for genetic effects [15, 40]. For traits influenced by several genetic factors, the specific genetic loci may induce distinct correlation structures among the measures, so that one can separate the effects of each genetic locus by multivariate analysis, even though this might not be possible with simple univariate analyses. Finally, multivariate analysis provides a statistically efficient mechanism for controlling the analysis-wide significance level when there are multiple trait observations for each subject. In multivariate analysis of quantitative traits, it is not always apparent whether a variable should be treated as a covariate or as an outcome. For example, in analysis of blood pressure, body-mass index (BMI), which is a measure of obesity, is often treated as a covariate. However, if a genetic factor influences both BMI and blood pressure, then adjusting blood pressure for BMI would reduce the effects from the major-genetic locus. Therefore, using methods that can analyze several traits jointly is essential. Genetic model-free methods [28, 21, 3] are more easily applied than full likelihood methods, which require modeling the prevalences of genetic factors along with the parameters to describe the genotype specific phenotype distributions. In a 2002 article [18], de Andrade and Amos provided an overview of the methodology currently available for multivariate linkage analysis. Modeling becomes more complex when observations are recorded over time. Studies in which one family member is observed over a specific time and variations in outcomes are sought are commonly called longitudinal. Several authors have developed and reviewed statistical methods for longitudinal cohort studies (see [23]). For genetic analysis, Province and Rao [37, 38] used path analysis to estimate the genetic and environmental effects in families in the presence of temporal trends or time effect, but did not include variance components (VC) to measure effects from specific genes. Models using structural equations have also been developed and widely applied in the field of behavioral genetics to twin studies [36], but these models, primarily directed to the estimation of polygenic and environmental effects, are difficult to use for studies of large families or extended pedigrees. de Andrade et al. [18] extended the general linear model framework for longitudinal studies of families observed over

a specific period of time and for which the familial correlation interacts with the temporal correlation.

This technical report attempts to provide an overview of the theory behind the variance components approach for analyzing one or more quantitative traits in the face of familial correlation. It also provides an introduction to the S-Plus/R *multic* library which contains software to carry out this analysis. Details of the theory, examples, and further directions are combined together in the rest of this report.

2 Software

2.1 Overview of primary functions

In 1996 `MULTIC` (*Multi*variate Analysis for *C*omplex Traits) was created [19] to analyze quantitative traits in familial data using a variance components approach. Unlike other software, it allows the user to look at multivariate traits and longitudinal data. We have created an S-Plus/R interface to the original `MULTIC` code and have added in user-friendly plotting functions, summarizing routines, and diagnostic tools.

The main S-Plus/R function is `multic` which is the modeling function for univariate, multivariate, or longitudinal phenotypes, allowing for multiple environmental covariates and the identity-by-descent (IBD) data. The function `multic` assumes that multipoint or singlepoint IBD information has already been assessed, and that this information is stored in a certain format. There are several software packages that can obtain this information. Currently we have written the utility function `solar2multic` for using IBD files that are in the `Solar` format [12, 13, 4] and the utility function `sw2mloci` for using IBD files that are in the `Simwalk` format [45]. In addition to `multic`, there are a number of supporting functions in the library. These are briefly described below.

- `polygene`: Prints descriptive statistics for the traits and covariates, plus covariate information and variance components for the polygenic and sporadic model. This can be applied to any multic object.
- `print.multic`: Summarizes basic information about the families and the maximum lod score in the `multic` object.
- `summary.multic`: Provides a short summary of the `multic` object listing the maximum lod score, it's location, and the top 5 families that contribute to this lod score (if the user has specified calc.fam.log.liks=TRUE).
- `mlociCut`: Subsets the `multic` IBD file to a smaller region.
- `plot.multic`: Plots the LOD scores from the specified `multic` object.
- `plot.family.lods`: Plots the LOD scores for the individual families that contribute the most to the overall peak LOD score.
- `residuals.multic`: This is an extractor function which allows the user to obtain several types of residuals from the `multic` object. In order to extract this information, you will need to fit `multic` with the `calc.residuals` option set to TRUE (the default is FALSE to save space).
- `fitted.multic`: This is an extractor function which allows the user to obtain the fitted values from the `multic` object.

- `clean`: `multic` creates a number of temporary files while it's running. Use `clean` if your call to multic is interrupted and you want to get rid of these temporary files.
- `expand.multic` and `expand.data`: These functions are used for bootstrapping when you are sampling families with replacement and you need to include a family more than once into the various files.
- `addGE`: This function uses input from more than one `multic` object and estimates a multivariate LOD score from these univariate `multic` objects.

There are also a large number of supporting routines which will rarely if ever be called directly by the user. A listing of these routines are found in the appendix. We have not ported the software to the Windows environment although we expect others will. This is not an argument against Windows, however it is not an environment that we routinely use.

2.1.1 Converting Solar IBDs

When converting IBD information that is in the `Solar` format, `solar2multic` utilizes the following files:

- phi2 file: The phi2.gz file contains the ID for two related individuals and the two times the kinship coefficient matrix.
- ped file: The pedigree file consists of one record for each individual in the data set. Each record must include the following fields: individual ID, father ID, mother ID, sex, famid. The file must be comma delimited.
- pedindex files: There are two pedindex files: pedindex.out and pedindex.cde. The file pedindex.out associates each ID in your pedigree file with a sequential ID used by all SOLAR files. The file pedindex.cde identifies the fields in the fixed width pedindex.out file.
- Directory where the ibd files are stored: Two kinds of IBD files can be provided (two-point or multipoint).

More details about the files can be found in SOLAR user's guide (`http://www.sfbr.org/solar/doc/00.contents.html`).

2.1.2 Converting Simwalk IBDs

When converting IBD information that is in the `Simwalk` format, `sw2mloci` utilizes the following files:

- ibd files: one for each family.
- map file: an optional map file that incorporates genetic distance.

Note that you need to have access to the S-Plus/R `kinship` library if you use the Simwalk IBDs.

2.2 Example: Data preparation

Our examples use data from the Rochester Family Heart Study (RFHS), a community-based cross-sectional study of the genetic epidemiology of atherosclerotic coronary artery disease and essential hypertension in Rochester, MN [50]. Between 1984 and 1991, 3978 members from 601 households underwent standardized medical interviews, physical examinations, and blood sampling at the Mayo Clinic. Subjects were ascertained through households with two or more children enrolled in the schools of Rochester, MN, in 1983. Additional details of the sampling design, recruitment, and study protocols have been previously published [48, 49]. Genotype and phenotype measurements were available for 2135 RFHS participants from 279 multigenerational pedigrees.

This first example illustrates how files from `Solar` or `Simwalk` can be translated to the right format for `multic`. The function `solar2multic` takes the `Solar` files and creates `mloci.out.gz` and `share.out.gz` in the directory *multicInput* as specified. The function `sw2mloci` takes the `Simwalk` files and creates `mloci.out.gz`. A dataframe called `d10` is including the phenotype and family relationship data is also created.

```
# Attach the multic library
library(multic)

# Translate the Solar files to the correct format for multic
# This creates the files share.out and mloci.out in the directory  multicInput

solar2multic(phi2='solar/phi2', pedigree='solar/roch.ped',
             pedindex.out='solar/pedindex.out',
             pedindex.cde='solar/pedindex.cde',
             ibd='solar/ibd10',output='multicInput')

# OR #

# Translate Simwalk files to the correct format for multic
# This creates the file mloci.out in the directory  multicInput.
# The information for share.out is calculated using the kinship library.

sw2mloci(directory='simwalk', map='simwalk/c10.map', output='multicInput')


# Read in the phenotype and family relationship data and
# create the dataframe d10

ped <- read.table("solar/roch.ped", header = T, sep = ',')
phen <- read.table("solar/roch.phen", header = T, sep = ',')
d10 <- merge(ped, phen)
d10 <- d10[order(d10$famid, d10$id), ] ## data needs to be sorted by famid
```

3 Polygenic and Sporadic Models

3.1 Theory

As was mentioned before, consider the case in which a particular trait, such as systolic blood pressure, is observed for families (or clusters of related individuals). Under a sporadic model (H_O: there is no genetic effect), the observed values of the trait for the members of the ith family can be represented by

$$\mathbf{y}_i = \boldsymbol{\mu} + \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\epsilon_i}. \tag{1}$$

It is important to note that the model described above is the same as the standard linear model.

Under a polygenic model (H_1: there is a genetic effect) the observed values of the trait for the members of the ith family can be represented by

$$\mathbf{y}_i = \boldsymbol{\mu} + \mathbf{X}_i\boldsymbol{\beta} + \mathbf{a}_i + \boldsymbol{\epsilon_i}, \tag{2}$$

where $\mathbf{y}_i$ is the vector of the observed values of the trait (or phenotype) for the ith family, $\boldsymbol{\mu}$ is the vector of overall mean, $\mathbf{a}_i$ is the unobservable vector of the additive random genetic effects for the ith family, $\mathbf{X}_i$ is the matrix of observable covariates, $\boldsymbol{\beta}$ is the vector of fixed covariate effects uncorrelated with the additive genetic effects and environmental effects, and $\boldsymbol{\epsilon}_i$ is the vector of environmental effects for the ith family, $\forall i, i = 1, 2, \ldots, k$.

We assume that the additive genetic effect for the ith family is $\mathbf{a}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{R})$, where $\mathbf{R}$ is the matrix of coefficients of relationship between pairs of related individuals (the same as twice the kinship coefficient matrix), for example, $\frac{1}{2}$ for full sibs and parent-offspring; the environmental effect for the ith family is $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$, where $\mathbf{I}$ is the identity matrix; and the additive genetic effect is uncorrelated with the environmental effect. Then $\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{X}_i\boldsymbol{\beta}, \mathbf{V})$, where $\mathbf{V} = \sigma^2\mathbf{R} + \tau^2\mathbf{I}$.

3.2 Estimation Methods

To estimate the VCs, σ^2 and τ^2, we can use the E-M algorithm, Newton-Raphson (N-R) method, or the Fisher Scoring method. These methods are well described in the literature for the case in which a design matrix can be specified for each VC [42]. For most genetic problems in humans, however, no design matrix can be specified a priori because the structure of the matrix will vary from family to family and markers are usually less than fully informative, and few reports concern the estimation of VC in such a case. The E-M algorithm is based on sufficient statistics of the complete data, which consist of the observed and unobserved data. Observed data are the observed quantitative traits or phenotypes, and unobserved data are the random components, in our case, the genetic components. The N-R and scoring methods are based on the derivatives of the log likelihood function. Each method has its own advantages and disadvantages.

The E-M algorithm generally takes longer to converge but does not produce negative estimates of the variance components, whereas the N-R and scoring methods converge faster, but can produce negative estimates unless boundary constraints are imposed. The E-M algorithm can yield the breeding values, an important measure in animal breeding, but it does not provide an information matrix of the estimates. To produce this matrix, we need to apply an additional step [34, 35]. Conversely, the scoring and N-R methods automatically provide this information matrix. As a note, we use maximum likelihood estimates and not restricted

maximum likelihood (REML) estimates. The first two methods have not yet been implemented in `multic`, though we have plans to do this. Details about all these methods can be found in Searle et. al [42].

The Expectation-Maximization Algorithm

The basic form of E-M equations for the above model are well known [32, 22]. The natural sufficient statistics for the "complete data" ($\mathbf{y}$, $\mathbf{a}$) when there are no fixed effects except for the overall mean $\boldsymbol{\mu}$, are $k^{-1}\mathbf{1}^t(\mathbf{y}-\mathbf{a})$, $k^{-1}\mathbf{a}^t\mathbf{R}^{-1}\mathbf{a}$, and $k^{-1}\boldsymbol{\epsilon}^t\boldsymbol{\epsilon}$, whose unconditioned expectations are $\boldsymbol{\mu}, \sigma^2$, and τ^2, respectively. Thus, the iterative equations are formed by setting new values for these parameters equal to the conditional expectations of the statistics taken at current parameter values

$$\mu^{(+)} = k^{-1}\mathbf{E}(\mathbf{1}^t(\mathbf{y}-\mathbf{a}) \mid \mathbf{x}, \boldsymbol{\mu}, \sigma^2, \tau^2) = k^{-1}\mathbf{1}'(\mathbf{y}-\boldsymbol{\eta}), \tag{3}$$

$$\sigma^{2(+)} = k^{-1}\mathbf{E}(\mathbf{a}^t\mathbf{R}^{-1}\mathbf{a} \mid \mathbf{y}, \boldsymbol{\mu}, \sigma^2, \tau^2) = k^{-1}[tr(\mathbf{R}^{-1}\Sigma) + \boldsymbol{\eta}^t\mathbf{R}^{-1}\boldsymbol{\eta}], \tag{4}$$

$$\tau^{2(+)} = k^{-1}\mathbf{E}(\boldsymbol{\epsilon}^t\ \boldsymbol{\epsilon} \mid \mathbf{x}, \boldsymbol{\mu}, \sigma^2, \tau^2) = k^{-1}[tr(\Sigma) + \boldsymbol{\epsilon}^t\boldsymbol{\epsilon}], \tag{5}$$

where

$$\boldsymbol{\eta} = \mathbf{E}(\mathbf{a} \mid \mathbf{x}, \boldsymbol{\mu}, \sigma^2, \tau^2), \qquad \boldsymbol{\epsilon} = \mathbf{y} - \boldsymbol{\mu} - \boldsymbol{\eta},$$

$$\Sigma = \sigma^2\mathbf{R}(\mathbf{I} - \sigma^2\mathbf{V}^{-1}\mathbf{R}) = \sigma^2\tau^2\mathbf{R}\mathbf{V}^{-1},$$

$$tr(\mathbf{R}^{-1}\Sigma) = \sigma^2\tau^2 tr(\mathbf{V}^{-1}),$$

$$tr(\Sigma) = \sigma^2\tau^2 tr(\mathbf{R}\mathbf{V}^{-1}) = \tau^2 tr(\mathbf{I} - \tau^2\mathbf{V}^{-1}).$$

The problem with these E-M equations is that they all require computation of $\mathbf{V}^{-1}$ at each iteration. For large pedigree, the computation of $\mathbf{V}^{-1}$ at each iteration can become computationally intensive. To deal with this issue, Thompson and Shaw [47] solved the above E-M equations without determining $\mathbf{V}^{-1}$. Instead, only the eigenvalues of $\mathbf{R}$ needed to be determined, which provide the eigenvalues of $\mathbf{V}$, hence, $\mathbf{V}^{-1}$. Thus the trace terms of equations (4) and (5) are easily computed.

The Newton-Raphson Method

In the VC estimation problem, we can estimate the set of parameters denoted by $\boldsymbol{\theta}$, which consists of the fixed effects coefficients and the VC parameters, by the N-R iterations

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - (\mathbf{H}^{(m)})^{-1}\nabla\mathbf{L}^{(m)}, \tag{6}$$

where $\mathbf{L}$ indicates the log likelihood function, $\mathbf{H}^{(m)}$ the Hessian matrix (second-derivative matrix), and $\nabla\mathbf{L}^{(m)}$ the gradient vector, with $\boldsymbol{\theta}$ replaced by $\boldsymbol{\theta}^{(m)}$.

The Fisher Scoring Method

This method uses an iteration scheme and replaces the Hessian matrix with the information matrix, which is the negative expectation of the Hessian matrix. By doing so, the information

matrix need only be calculated once, thus, avoiding the computational burden of iteratively calculating the Hessian. Then, the scoring method uses the following iteration scheme:

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} + (\mathbf{I}^{(m)})^{-1}\nabla\mathbf{L}^{(m)}, \tag{7}$$

where $\mathbf{I}^{(m)}$ is the information matrix. Currently this is the only method implemented in `multic`.

In the VC estimation problem, we can estimate the set of parameters denoted by $\boldsymbol{\theta}$, which consists of the fixed effects coefficients and the VC parameters, by the N-R iterations

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - (\mathbf{H}^{(m)})^{-1}\nabla\mathbf{L}^{(m)}, \tag{8}$$

where $\mathbf{L}$ indicates the log likelihood function, $\mathbf{H}^{(m)}$ the Hessian matrix (second-derivative matrix), and $\nabla\mathbf{L}^{(m)}$ the gradient vector, with $\boldsymbol{\theta}$ replaced by $\boldsymbol{\theta}^{(m)}$.

3.3 Example: The `polygene` function

As an initial step, it is important to look at the data and determine if there is an indication of a genetic component in the overall trait variance. It is also necessary to view the trait distribution and make sure the overall model distributions are met. Influential points and highly skewed data can have a negative impact on the overall analysis.

The `multic` function requires the user to supply family variables, a formula specifying the traits and covariates, if any, and the name of the dataset. All other arguments are optional. If the kinship matrix is not supplied with a `share.out` file, this information is automatically calculated using a function available from the `kinship` library. The formula follows standard S-Plus rules with a trait (or traits), a tilde, and covariates separated by a plus-sign (`+`). A one (`1`) is listed on the right-hand side of the function to specify a model with no covariates. The `control` option allows the user to specify initial values and convergence criteria, and to request that a shared environmental component be calculated.

```
##############################################################################
## The polygenic model is fit when no marker data is specified
##############################################################################

## Fit with the share.out file
> fit1a <- multic(formula=sys.avg  ~ male + agexam + agexam^2,
                data=d10, famid, id, fa, mo, sex, share='multicInput/share.out.gz')

## OR

## Fit without the share.out file
> fit1b <- multic(formula=sys.avg ~ male + agexam + agexam^2,
                data=d10,famid, id, fa, mo, sex )

##############################################################################

Call:
multic(formula = sys.avg ~ male + agexam + agexam^2, data = d10, famid = famid,
        id = id, dadid = fa, momid = mo, sex = sex)

Fitting traits without covariates...
```

```
   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,
  11,  12,  13,  14,  15,  16,  17,  18,  19,  20,
  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,
  31,  32,  33,  34,  35,  36,  37,  38,  39,  40,
  41,  42,  43,  44,  45,  46

Calculating likelihoods under null hypothesis on Thu Feb 23 13:52:34 2006
Iteration Number:
   1,   2,   3,   4,   5,   6,   7,   8,   9,  10

Loading (fixed, random) effects from file...
Loading inverse of the expected second derivative from file...
Loading variance-covariance sandwich matrix from file..
Calculating heritability values...
Calculating correlation values...
```

The `polygene` function provides basic summaries of the cohort and allows the user to check for evidence of a genetic component in the trait (or traits) of interest (e.g. heritability). The heritability estimate ($h^2 = \sigma_R^2/\sigma_T^2$) in this example is 0.393, with an associated p-value of $<$ 0.001, confirming that there is evidence that this trait has a genetic component. The proportion of the total variance ($\sigma_T^2 = \sigma_R^2 + \tau^2$) due to the covariates is 0.49.

```
> polygene(fit1b)

  Pedigree Information
  --------------------
 Pedigrees People Females Males Probands
       279   2494    1243  1251        0

  Complete-Count Information
  --------------------------
 People Females Males
   2114    1101  1013

  Other Information
  -----------------
 Traits Covariates Locations
      1          3         0

  Descriptive Statistics for the Variables
  ----------------------------------------
              n      Mean   Std.Dev Minimum Maximum Kurtosis Skewness
    sys.avg 2114  114.3000   21.4900  69.670  272.70   2.9760  1.28200
       male 2114    0.4792    0.4997   0.000    1.00  -1.9940  0.08327
      agexam 2114   38.4200   23.2400   5.158   90.27  -1.2600  0.28350
I(agexam^2) 2114 2016.0000 1988.0000  26.610 8149.00  -0.4756  0.84210

  Covariate coefficients
  ----------------------
Trait 1 ( sys.avg ):
             Estimate Std.err  t.value p.value
(Intercept)  1.003e+02 1.10800  90.5000  0.0000
       male  6.009e-01 0.64920   0.9257  0.3547
```

```
     agexam -1.113e-01 0.05670  -1.9640  0.0497
i(agexam^2)  8.902e-03 0.00067  13.2900  0.0000

  Variance Components
  -------------------
Polygenic:
 Estimate Std.err  Wald W.p.value    h^2  se.h^2 h.p.value
    92.74   11.19 68.67         0 0.3929 0.03982         0

Environmental (non-genetic component):
 Estimate Std.err Wald W.p.value
    143.3   9.367  234         0

  Proportion of Variance due to the Covariates
  --------------------------------------------
R.sq: 0.4917
```

4 Major Gene or QTL Analysis

4.1 Theory

Advances in molecular biology enable us to evaluate the association and genetic linkage of markers with quantitative traits. Let us assume that in addition to the polygenic additive effect, a major gene is responsible for the trait and a marker locus is linked with this major gene. Then equation (2) can be rewritten as

$$\mathbf{y}_i = \boldsymbol{\mu} + \mathbf{X}_i\boldsymbol{\beta} + \mathbf{a}_i + \mathbf{g}_i + \boldsymbol{\epsilon_i}, \tag{9}$$

where $\mathbf{g}_i$ is the unobservable major gene effect for the ith family. Amos [1994] showed that $\mathbf{g}_i \sim (\mathbf{0}, \sigma_g^2\mathbf{F}_i)$, where $\mathbf{F}_i = (f(\theta, \pi_{ijl}))$, θ is the recombination fraction between the major gene and the marker locus, and π_{ijl} is the IBD sharing assessed by marker typings for pairs of individuals (j, l) at the ith family. The values for $f(\theta, \pi_{ijl})$ for several relationships are given by Amos [6].

We assume that the dominance component of variance for the major gene effect is negligible. Amos [5] showed that, except for unusual situations, the additive component of variance usually dominates the dominance variance. For most linkage testing situations, the additive assumption is reasonable. The dominance variance can become appreciable for recessive traits. When the recessive disease allele frequency is less than 1/3, the dominance variance is greater than the additive variance. Consequently, here σ_g^2 refers to the additive major gene component of variance. The value of π_{ijl} can assume only 3 values: 0, $\frac{1}{2}$, or 1. For incomplete marker data estimates of π_{ijl} can be obtained using different algorithms [27]; [45].

Under tight linkage, we assume that $\theta = 0$. Then $f(\theta, \pi_{ijl}) = \pi_{ijl}$, and consequently, $\mathbf{g}_i \sim (\mathbf{0}, \sigma_g^2\boldsymbol{\Pi}_i)$, where $\boldsymbol{\Pi}_i = (\pi_{ijl})$. In this case, the variance-covariance matrix for each family $\mathbf{y}_i$ is $\mathbf{V} = \sigma^2\mathbf{R} + \sigma_g^2\boldsymbol{\Pi}_i + \tau^2\mathbf{I}$. Here we assume that $\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{X}_i\boldsymbol{\beta}, \mathbf{V})$, despite the fact this assumption may not be correct because the major gene effect does not necessarily follow a normal distribution.

To estimate the VC of the major gene effect, one extra parameter is added to the N-R and scoring equations (8) and (7), respectively. These equations can produce negative estimates of the VC parameters, which is an inadmissible solution [42]. We applied the step-halving method suggested by Jennrich and Schluchter [31] to avoid this problem whenever a

parameter was estimated outside the admissible region. The E-M algorithm has the advantage to accommodate the parameter constaints. However, it requires to invert $\mathbf{\Pi}_i$ for each family i. For some families, this matrix may be singular . To overcome this problem, the computation of the spectral values decomposition of $\mathbf{\Pi}_i$ was proposed by Iturria et al. [30].
Quasi-likelihood methods are also available for QTL analysis but these two approaches are not currently available in `multic`.

4.2 Statistical Tests

The hypothesis test for univariate data is $H_0 : \sigma^2_{mg} = 0$ versus $H_A : \sigma^2_{mg} \geq 0$. To test whether this hypothesis is true, we use the likelihood ratio test (LRT) and the Wald test (WT). The Wald test is constructed by dividing the parameter estimate σ^2_g by its standard error, and it has t distribution. The LRT is distributed as a mixture of χ^2's distributions as suggested by Self and Liang [43] when the true parameter value is on the boundary of the parameter space. For the univariate case, the mixture is a $1/2\ \chi^2_0 + 1/2\ \chi^2_1$.

4.3 Example: Whole Chromosome

Typically you will want to look at multiple markers within a chromosome to determine the maximum lod score. By simply adding in the IBD information using the `mloci.out` option, the analysis is run for each marker location that is specificied in `mloci.out`. Note that this analysis may take awhile. For this example 185 mibd positions were studied and the analysis took approximately 80 minutes to run.
The `print` of this `multic` object gives you basic information about how many subjects were used in this analysis and where the maximum lod score was located. In this example the maximum lod score was 1.45, located around 177 cM.

```
## Note: mloci.out and share.out were calculated in the Software section

> mult10 <- multic(sys.avg ~ sex + agexam + agexam^2, data=d10,
                   famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                   mloci='multicInput/mloci.out.gz',
                   share='multicInput/share.out.gz')

> print(mult10)
Call:
multic(formula =  sys.avg ~ sex + agexam + agexam^2, data = d10,
        famid = famid, id = id, dadid = fa, momid = mo, sex = sex,
        mloci.out = "multicInput/mloci.out.gz",
        share.out = "multicInput/share.out.gz", calc.fam.log.liks = T)

Multivariate analysis counts:

  Pedigree Information
  --------------------
 Pedigrees People Females Males Probands
       279   2494    1243  1251        0

  Complete Phenotype Count Information
  ------------------------------------
 People Females Males
```

```
   2114    1101  1013

  Other Information
  -----------------
 Traits Covariates Locations
      1          3       185

Maximum lod score: 1.4458
         at locus: mibd.10.177
at positions (cM): 177
```

The `summary.multic` function provides a few more details, listing a range surrounding the maximum lod score. It also provides the top 5 families that contributed to the maximum if you originally ran multic with the option `calc.fam.log.liks=TRUE`. Using the default plot function (see figure 1) provides a quick look at the overall results. Additional information can be found in the multic object; `mult10$log.liks` provides the loglikelihood, lod score value and the corresponding p-value.

```
> summary(mult10)
Call:
multic(formula =  sys.avg ~ sex + agexam + agexam^2, data = d10,
        famid = famid, id = id, dadid = fa, momid = mo, sex = sex,
        mloci.out = "multicInput/mloci.out.gz",
        share.out = "multicInput/share.out.gz", calc.fam.log.liks = T)

Maximum lod score: 1.4458
         at locus: mibd.10.177
at positions (cM): 177

The top 5 families and their lod score contributions
to the maximum lod score are:
      80     249     163     102     45
 0.41722 0.24083 0.22126 0.20274 0.1943

The minimum and maximum positions (cM) that produced a lod score
greater than the maximum - 1 ( 1.4458 - 1 )
and are contiguous to mibd.10.177 are:
133 184

> plot(mult10)
```

4.4 Example: Subset marker region or subjects

In some circumstances you may want to run models focusing on a certain region of the chromosome or only using a subset of the families. In order to analyse a subset of the chromosome, you first need to create a subset of your mloci object using the `mlociCut` function. Subsetting subjects is possible using the `subset` option within `multic`. In this particular example, the largest families (those with at least 10 members) are identified with a TRUE/FALSE variable. For this subset analysis, only those markers between 120 and 180 cM are of interest. In the print summary 61 locations are examined instead of 185 and 560 people were analysed instead of 2114 (see complete phenotype count information). Using only the largest families in this subsetted region, the lod score was reduced from 1.45 to only 0.639.

Figure 1: *Default plot of a multic object. The trait is systolic blood pressure and the covariates are sex, age at exam, and age at exam*^{2}.

```
### Cut mloci.out in order to only look at a certain region
> mlociCut('multicInput/mloci.out', c(120,180), 'multicInput/mloci.cut')

## identify families that have at least 10 members, create T/F flag if large
> tmp <- rle(d10$famid)
> large <- !is.na(match(d10$famid, tmp$values[tmp$lengths>10]))

> sub10 <- multic(sys.avg ~ sex + agexam + agexam^2, data=d10, subset=large,
                famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                mloci='multicInput/mloci.cut.gz', share='multicInput/share.out.gz')

> sub10
Call:
multic(formula =
        sys.avg ~ sex + agexam + agexam^2, data = d10,
        famid = famid, id = id, dadid = fa, momid = mo, sex = sex,
        mloci.out = "multicInput/mloci.cut.gz",
        share.out = "multicInput/share.out.gz", subset = large)

Multivariate analysis counts:

  Pedigree Information
  --------------------
 Pedigrees People Females Males Probands
       279   2494    1243  1251        0

  Complete Phenotype Count Information
```

```
 ------------------------------------
People Females Males
   560     294   266

 Other Information
 -----------------
Traits Covariates Locations
     1          3        61

Maximum lod score: 0.63919
         at locus: mibd.10.176
at positions (cM): 176
```

5 Multivariate Traits

5.1 Theory

The multivariate variance components (MVC) approach is an extension of the univariate approach described by

$$\mathbf{Y}_j|\mathbf{X}_j = \boldsymbol{\mu} + \mathbf{X}_j\boldsymbol{\beta} + \mathbf{a}_j + \mathbf{g}_j + \mathbf{e}_j \tag{10}$$

where $\mathbf{Y}_j$ is a vector of dimension $N_j * m$, for family j, where N_j is the number of individuals for family j and m is the number of traits; $\boldsymbol{\mu}$ is the overall mean vector of dimension N_j for family j, $\mathbf{X}_j$ is an $N_j \times p$ matrix of covariates, $\boldsymbol{\beta}$ is a p-vector of regression coefficients, $\mathbf{g}_j$ is a N_j-vector of genetic effects by which the major locus affects the trait values for family i, $\mathbf{a}_j$ is a 1 x N_j vector expressing how the additive polygenic factor affects the trait values for family j, and $\mathbf{e}_j$ is residual variation (or environmental effects) from the model. For more details see Amos [6] and de Andrade et al. [16]. The MVC approach is also a model-free approach, and it has advantages over model-dependent approaches. To simply describe these models we use the vec transformation [3, 21, 7], to string out the observations as a single vector and then allow elements of this vector to be correlated, according to the model proposed by the equation 10. Let $\mathbf{Y}_j = (Y_{11}, ..., Y_{1N_j}, ..., Y_{mN_j})'$ be a vector of m multivariate trait values for N_j members of the j^{th} family. Let N be the total number of families, $\boldsymbol{\beta}$ a vector of dimension mp of the regression coefficients for the p covariates (including a vector of 1's corresponding to the overall mean), $\mathbf{X}_j = \mathbf{I}_m \otimes \mathbf{X}_{N_j,p}$ an $mN_j \times mp$ known matrix of covariate values for the j^{th} family, where $\otimes$ is the Kronecker product. Then, the variance-covariance matrix of the m traits, $\mathbf{V}_j$, with dimension $mN_j \times mN_j$ is

$$\mathbf{V}_j = \mathbf{A} \otimes \mathbf{R}_j + \mathbf{B} \otimes \boldsymbol{\pi}_j + \mathbf{C} \otimes \mathbf{I}_j, \tag{11}$$

where, $\mathbf{R}_j$ is the $N_j \times N_j$ matrix of the coefficients of relationship for the j^{th} family; $\boldsymbol{\pi}_j$ an $N_j \times N_j$ matrix of estimated proportion of alleles identical by descent (IBD) for pairs of related individuals for the j^{th} family; $\mathbf{I}_j$ is the $N_j \times N_j$ identity matrix; and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are, respectively, polygenic, major-gene, and residual variance-covariance matrices each of dimension $m \times m$. A fourth term to measure dominance components can be added. Because the dominance component of variance is usually much smaller it is ignored here, but can be modeled by including by modeling increased covariance among pairs sharing 2 alleles IBD.

Similarly, additional terms to model shared environment can be added. When longitudinal data are considered, the error variance structure can be modified to take account serial correlation among the observations [18]. A special approach can be taken for discrete/quantitative traits. In this approach, a decomposition is effected in which the quantitative trait is first conditioned on the discrete trait [51].

The Multivariate Variance Components Test

To test for genetic linkage, we construct a likelihood ratio test. Under the null hypothesis, the major gene parameter(s) $\mathbf{B}$ of equation 11 are constrained to $\mathbf{0}$. For simplicity, let us consider bivariate traits. For bivariate linkage analysis of an additive genetic effect, the parameters are σ^2_{a1}, σ^2_{a2} , and $\sigma_{a1,a2}$ where the first two components measure the major-genetic variance of the traits and the third component measures the major-gene covariance for the traits. We also usually constrain the major-gene variances to be positive so that they fall in the admissible part of the parameter space. As a result, the distribution of the bivariate test that the major-gene components and covariance are zero is a mixture of 1/4 χ^2_0, 1/2 χ^2_1, and 1/4 χ^2_3 as suggested by Self and Liang [43]. This follows because for one-quarter of the parameter space, both genetic variance parameters are estimated to be positive and hence lead to a chi-squared test having 3 degrees of freedom; for one-half of the parameter space, one of the genetic variances is constrained to be 0 and hence the major gene covariance is 0 so that the chi-squared distribution has 1 degree of freedom, while for the remaining one-quarter of the parameter space, both genetic variances are constrained to be zero, resulting in a degenerate distribution of a point mass at 0.
Because the same major gene alleles are assumed to be determining the two traits, it is logical to consider imposing the constraint $\sigma_{a1,a2} = \pm\sigma_{a1}\sigma_{a2}$, which is always satisfied whenever there is a single genetic factor in a region and the dominance components of variance affecting each trait is 0. As discussed by Almasy et al. [3], the observed correlation attributable to a locus may not be one if there are multiple loci affecting both traits in a region. Therefore, they proposed testing the hypotheses of pleiotropy, which presumes that the trait(s) are influenced by the same gene versus coincident linkage, which presumes that there are two or more linked loci that separately influence the traits. If the covariance is constrained to be the product of the square root of the variances, then the hypothesis test of linkage for either of the traits becomes a mixture of 1/4 χ^2_0, 1/2 χ^2_1, and 1/4 χ^2_2. In this case, the covariance is no longer a parameter to be estimated. Amos et al. [7] compared the efficacy of fitting data either with or without this constraint on the covariance. They found rather similar power for either the unconstrained or constrained tests when the empirically derived critical values were used.

5.2 Example: look at three traits

As was mentioned earlier, hypertension is a trait that relies on the underlying quantitative measure of systolic blood pressure and/or diastolic blood pressure. As has been shown earlier, we can easily study the genetic influence of genetic markers on an individual trait, but it may be that the genetic factor influences both BMI and blood pressure. Therefore, we will look at the influence of the genetic information on systolic blood pressure, diastolic blood pressure, and BMI. To do this, we simply modify our endpoint in the previous call to multic. Note that we used the subsetted marker region `mloci.cut.gz` to save analysis time. The maximum lod score is now 5.02, up from 1.45 when we examined sys.avg alone. The position has also changed and is now 145 when it was 177.

```
> bsd10 <- multic(cbind(bmi,dia.avg,sys.avg) ~ sex + agexam + agexam^2,
```

Figure 2: *Multivariate analysis combining bmi, diastolic blood pressure, and systolic blood pressure. Analysis was run in a limited portion of chromosome 10.*

```
               data=d10, famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
               mloci='multicInput/mloci.cut.gz',
               share='multicInput/share.out.gz', max.iterations=40)

> bsd10

Call:
multic(formula =
       cbind(sys.avg, bmi, dia.avg) ~ sex + agexam + agexam^2,
       data = d10, famid = famid, id = id, dadid = fa, momid = mo, sex = sex,
       mloci.out = "multicInput/mloci.cut.gz", share.out =
       "multicInput/share.out.gz", max.iterations = 40)

Multivariate analysis counts:

  Pedigree Information
  --------------------
 Pedigrees People Females Males Probands
       279   2494    1243  1251        0

  Complete Phenotype Count Information
  ------------------------------------
 People Females Males
   2081    1084   997

  Other Information
  -----------------
```

Figure 3: *Is it worthwhile to run a multivariate analysis? Determine an estimate of how much you'll actually gain when combining the three traits.*

```
 Traits Covariates Locations
      3          3        61

Maximum lod score: 5.0241
         at locus: mibd.10.145
at positions (cM): 145
```

This analysis analysis is useful, but it can take quite awhile to run. If you've already run univariate analyses on your traits, there is a quick way to see if it's even worthwhile to look at a multivariate analysis (see figure 3). The function `addGE` will take several univariate trait multic objects and provide an estimate of the maximum multivariate LOD score [20].

```
> add2 <- addGE(list(bmi10, dia10, sys10), combine = 2, plotit = T, ylim=c(0,8), legend=F)
> add3 <- addGE(list(bmi10, dia10, sys10), combine = 3, plotit = F)
> lines(add3$cM, add3$lod, col=4, lwd=2, lty=4)
> key(corner=c(0,1), lines=list(lwd=2, col=1:4, lty=1:4),
      text=list(c('BMI-Dia','BMI-Sys','Dia-Sys','BMI-Dia-Sys'), col=1:4))

> add2[1:4, ]

  comb   statChi       pChi  cM       lod
1 1, 2 2.1323231 0.07211102 120 1.1419984
2 1, 3 0.5145535 0.23658714 120 0.6260089
3 2, 3 1.1492481 0.14185331 120 0.8481605
4 1, 2 2.1762557 0.07007754 121 1.1544212
```

Figure 4: *Combine standard tools such as Principal Components Analysis and Factor Analysis with multic when combining multiple traits.*

5.3 Example: usage of Principal Components and Factor Analysis to combine traits

Multivariate analysis is really only practical for two or three traits at a time. The `multic` function has a limit of five traits that can be examined together and that analysis would take multiple days to run. Principal Components Analysis (PCA) and Factor Analysis (FA) are both dimension-reduction techniques, in the sense that they can be used to replace a large set of observed variables with a smaller set of new variables. The two methods are different in their goals and in their underlying models, but they provide another tool in multivariate analysis. A general rule is that you should use Principal Components Analysis when you are interested in summarizing your traits using fewer dimensions, whereas Factor Analysis is used more when you need an explanatory model for the correlations among the traits.

The following example looks at multiple traits that are often indicators of cardiac problems (bmi, waist size, waist to hip ratio, systolic blood pressure, diastolic blood pressure, triglycerides, hdl, ldl, glucose, and insulin). In analysis not shown, both the PCA and FA suggest using only the first element of each analysis. These new summary measures are then used in `multic` as a new trait. The plot of the results indicate that neither methodology fully captures the information from these traits with a maximum lod scores of 1.43 using PCA and 2.14 using FA.

```
> traits <- cbind(bmi, e.waist, invwhr, sys.avg, dia.avg, k.trig, k.hdl,
                  k.ldl, gluc.avg, insulin)
## determine which rows have no missing data
> traits.notna <- rowSums(is.na(traits))==0

## run the principal components and factor analysis
```

```
> traits.pc <- princomp(traits, cor=T, na.action=na.omit)
> traits.f <- factanal(traits, factors=4, na.action=na.omit)

## create matrices of the components that has the same length
## as the original data
> pcscores <- matrix(NA,nrow=nrow(d10), ncol=10)
> pcscores[traits.notna,] <- predict(traits.pc)

> factorscores <- matrix(NA,nrow=nrow(d10), ncol=4)
> factorscores[traits.notna,] <- predict(traits.f)

## save the first component in the original data frame for further analysis
> d10$pc1 <- pcscores[,1]
> d10$fac1 <- factorscores[,1]

## run the multic analysis
> fitpc1 <- multic(pc1 ~ sex + agexam + agexam^2, data=d10,
                famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                mloci='multicInput/mloci.cut.gz', share='multicInput/share.out.gz')

> fitfac1 <- multic(fac1 ~ sex + agexam + agexam^2, data=d10,
                famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                mloci='multicInput/mloci.cut.gz', share='multicInput/share.out.gz')

## plot results
> plot(fitfac1, col=2, lty=2, lwd=2, ylim=c(-1,8))
> par(new=T)
> plot(fitpc1, col=1, lty=1, lwd=2, ylim=c(-1,8))
> key(corner=c(0,1), lines=list(lty=1:2, col=1:2), lwd=2,
      text=list(c('Principal Components','Factor Analysis'), col=1:2))
```

6 Longitudinal Data

6.1 Theory

We extended the VC approach proposed by Hopper and Mathews [28] and Amos [6] to accomodate longitudinal familial data. The VC approach is described in detail for observations at a single time point for two-point analysis [6, 16], for multipoint analysis [4], and for multivariate traits [21, 30, 7].

For longitudinal familial data, suppose we observe a quantitative trait, such as systolic blood pressure, in families. Then, the model for T time points of observed values of the trait for the j^{th} relative in a family can be written as

$$\mathbf{Y}_j = \boldsymbol{\mu} + \mathbf{X}_j\boldsymbol{\beta} + \mathbf{a}_j + \mathbf{g}_j + \mathbf{s}_j + \boldsymbol{\epsilon}_j \tag{12}$$

where $\mathbf{Y}_j$ is the vector having T time points of observed values of the quantitative trait. $\boldsymbol{\mu}$ is the vector of overall mean for the T time points. $\mathbf{X}_j$ is the $T \times p$ matrix of observed covariates where p is the number of covariates. $\boldsymbol{\beta}$ is the vector of covariate effects. $\mathbf{a}_j$ is the vector having T time points of unobserved additive polygenic effects with $a_{jt} \sim \mathcal{N}(0, \sigma^2_{a.t})$ for all

time point t. $\mathbf{g}_j$ is the vector having T time points of unobserved additive major gene effects with $E(g_{jt}) = 0$ and $\text{var}(g_{jt}) = \sigma^2_{g.t}$ for all time point t [6]. Typically, the number of alleles in $\mathbf{g}_j$ and the degree of their effects are not known. The methods we propose do not require that the number of alleles be known. $\mathbf{s}_j$ is the vector of common shared environmental effects with $s_{jt} \sim \mathcal{N}(0, \sigma^2_{s.t})$ for all time point t. The $\mathbf{s}_j$ can be partitioned by sets of dummy variables representing the common shared environment for sibships, parents-offspring, and spouses. $\boldsymbol{\epsilon}_j$ is the vector of random environmental effects with $\epsilon_{jt} \sim \mathcal{N}(0, \tau^2_t)$ for all time point t. We assume the random effects, $\mathbf{a}_j$, $\mathbf{g}_j$, $\mathbf{s}_j$, $\boldsymbol{\epsilon}_j$, are uncorrelated with each other, and with the covariate effects, and $\mathbf{Y}_j$ follows a multivariate normal distribution. This assumption is not critical for estimation [29] but violations of multivariate normality can influence the accuracy of hypothesis tests [1]. Although an extra term for measuring dominance components can be added, we did not do so because the dominance component of variance is usually much smaller than other components.

The covariances of the traits for individuals j and l at times t and $t^{'}$ are

$$\text{cov}(y_{jt}, y_{lt'}) = \begin{cases} \sigma^2_{a.t} + \sigma^2_{g.t} + \sigma^2_{s.t} + \tau^2_t & j = l, t = t' \\ \sigma_{a.tt'} + \sigma_{g.tt'} + \sigma_{s.tt'} + \tau_{tt'} & j = l, t \neq t' \\ \delta_{jl}\sigma_{a.t} + \pi_{jl}\sigma_{g.t} & j \neq l, t = t' \\ \delta_{jl}\sigma_{a.tt'} + \pi_{jl}\sigma_{g.tt'} & j \neq l, t \neq t' \end{cases}$$

To express $\mathbf{V}_i$ (the variance-covariance matrix of $\mathbf{Y}_i$) in a simpler way, we considered the situation in which all the family members have already attained adulthood at the time of the first measure. An individual was considered an adult when he or she is older than 16 years or has reached puberty.

To describe these models simply, we use the vec transformation [21], to string out the observations as a single vector and allow elements of this vector to be correlated, according to the model proposed in equation 12. Let $\mathbf{Y}_i = (\mathbf{Y}^{'}_{i1}, \ldots, \mathbf{Y}^{'}_{iT})^{'}$ be a vector of T time point trait values for k_i members of the i^{th} family, where $\mathbf{Y}^{'}_{it} = (Y_{i1t}, \ldots, Y_{ijt}, \ldots, Y_{ik_it})^{'}$ for t=1,...,T. Let $E(\mathbf{Y}_i) = \mu + \mathbf{X}_i\boldsymbol{\beta}$, and

$$\mathbf{V}_i = \mathbf{A} \otimes \mathbf{R}_i + \mathbf{B} \otimes \mathbf{\Pi}_i + \mathbf{C} \otimes \mathbf{S}_i + \mathbf{D} \otimes \mathbf{I}_i,$$

where $\otimes$ defines the direct product of two matrices [41]; $\mathbf{R}_i$ is a $k_i \times k_i$ matrix of coefficient of relationship between pairs of relatives; $\mathbf{\Pi}_i$ is a $k_i \times k_i$ matrix of IBD values for the i^{th} family; $\mathbf{S}_i$ is a $k_i \times k_i$ matrix of indicator values in which, for sibships, $S_{i,j,l} = 1$ if $j = l$ or j and l belongs to the same sibship and $= 0$ otherwise (alternatively the matrix $\mathbf{S}$ can be modified to accomodate the common shared environment of parents-offspring and spouses); $\mathbf{I}_i$ is a $k_i \times k_i$ identity matrix; and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are, respectively, polygenic, major gene, common shared, and random environment variance-covariance matrices each of dimension $T \times T$. These matrices are represented by $\mathbf{A} = (\sigma_{a.tt'}), \sigma_{a.tt} = \sigma^2_{a.t}$, $\mathbf{B} = (\sigma_{g.tt'}), \sigma_{g.tt} = \sigma^2_{g.t}$, $\mathbf{C} = (\sigma_{s.tt'}), \sigma_{s.tt} = \sigma^2_{s.t}$, and $\mathbf{D} = (\tau_{tt'}), \tau_{tt} = \tau^2_t$, with their typical elements in the parentheses.

We applied the scoring algorithm to estimate the fixed effects and the variance components parameters [19].

6.2 Longitudinal Heritability Measure

One way to measure polygenic and major gene heritabilities for longitudinal data is to extend the standard heritability measure to incorporate the serial VC components. Thus, the polygenic and major gene heritabilities can be expressed as

$$h^2 = \frac{\sum_{t=1}^{T} \sigma^2_{a.t} + \sum_{t<t'} \sigma_{a.tt'}}{\sum_{t=1}^{T} \left(\sigma^2_{a.t} + \sigma^2_{g.t} + \sigma^2_{s.t} + \tau^2_t\right) + \sum_{t<t'} \left(\sigma_{a.tt'} + \sigma_{g.tt'} + \sigma_{s.tt'} + \tau_{tt'}\right)} \tag{13}$$

and

$$h^2_g = \frac{\sum_{t=1}^{T} \sigma^2_{g.t} + \sum_{t<t'} \sigma_{g.tt'}}{\sum_{t=1}^{T} \left(\sigma^2_{a.t} + \sigma^2_{g.t} + \sigma^2_{s.t} + \tau^2_t\right) + \sum_{t<t'} \left(\sigma_{a.tt'} + \sigma_{g.tt'} + \sigma_{s.tt'} + \tau_{tt'}\right)}. \tag{14}$$

6.3 Longitudinal Statistical Tests

Hypothesis tests for longitudinal data can be constructed to test whether any of the VCs differ from hypothesized values. For genetic linkage analysis, one would test $H_0 : \sigma^2_{g.t} = 0$ for all time points t against $H_1 : \sigma^2_{g.t} > 0$ for some time point t. To test whether this hypothesis is true, we can use the likelihood ratio test (LRT) or a Wald-type score test. We define the LRT as -2(log likelihood under H_0 - log likelihood under H_1). For the linkage test, the LRT is often preferable to a Wald test for two reasons. First, the parameters modeling the major gene and polygenic components are highly correlated. Because of this correlation, the variance of the components may not be well estimated, so Wald tests for the linkage parameter may not be reliable. Second, the information for linkage depends upon the number of informative meioses, which may be limited. Because the LRT is more robust for small samples than Wald tests, use of the LRT is probably preferable. Because the major gene covariance ($\sigma_{g.12}$) is not defined when one of the major-gene VC is zero, the distribution of the hypothesis test that the major-gene components are zero may not converge to a limiting chi-squared distribution. The LRT is distributed as a mixture of χ^2 distributions as suggested by Self and Liang [43] when the true parameter value is on the boundary of the parameter space. For longitudinal data, the mixture of χ^2s will depend on the number of time points. For instance, for two time points, H_0: $\boldsymbol{\sigma}_g = (\sigma^2_{g.1}, \sigma_{g.12}, \sigma^2_{g.2})' = \mathbf{0}$ against H_1: $\sigma^2_{g.t} > 0$ for some t. Thus, the LRT is distributed as a mixture of χ^2 distributions, $1/4\ \chi^2_0 + 1/2\ \chi^2_1 + 1/4\ \chi^2_3$. We and others performed simulation studies to confirm that the likelihood ratio test is distributed as a mixture of χ^2's under H_0 [7, 30].

For interaction between genes and time, one would test whether the genetic factors are the same at different time points by introducing constraints that they are equal: $H_0 : \sum_{i=1}^{T} c_t\, \sigma^2_{g.t} = 0$, for $\sum_{i=1}^{T} c_t = 0$ against $H_A : \sigma^2_{g.t} \neq \sigma^2_{g.t'}$ for some time points t, t'. To test whether this hypothesis is true, we can use a Wald test, W. For instance, for some two time points, $H_0 : \eta_0 = \sigma^2_{g.t} - \sigma^2_{g.t'} = 0$ against $H_A : \sigma^2_{g.t} \neq \sigma^2_{g.t'}$,

$$W = \left(\frac{\hat{\eta} - \eta_0}{\sqrt{var(\hat{\eta})}}\right)^2 \tag{15}$$

where $\hat{\eta} = \widehat{\sigma^2_{g.t}} - \widehat{\sigma^2_{g.t'}}$, and $\text{var}(\hat{\eta}) = \text{cov}(\widehat{\sigma^2_{g.t}}, \widehat{\sigma^2_{g.t'}})$, with W asymptoticaly distributed as χ^2_1.

6.4 Example: look at three time-points

For this example we used one of the simulated datasets from the Genetic Analysis Workshop 13. We've chosen to use 7 markers from chromosome 21, replicate 3, focusing on systolic blood pressure measured at multiple times (visits 10, 11, and 12).

```
> fit2 <- multic(formula=cbind(sbp10,sbp11,sbp12) ~  age10+sex10 + age11+sex11 + age12+sex12,
                 data=d21, famid, id, fa, mo, sex,
                 longitudinal=T,
                 mloci='multicInputLong/mloci.cut.gz',
                 share='multicInputLong/share.out.gz')


> summary(fit2)
Call:
multic( formula =
        cbind(sbp10, sbp11, sbp12) ~ age10 + sex10 + age11 + sex11 + age12 +
        sex12, data = d21, famid = famid, id = id, dadid = fa, momid = mo,
        sex = sex, mloci.out = "multicInputLong/mloci.cut.gz",
        share.out = "multicInputLong/share.out.gz", longitudinal = T)

Maximum lod score: 1.2772
         at locus: mibd.21.29.8
at positions (cM): 29.8

Since multic was run with calc.fam.log.liks = F (default),
the top families and their lod scores have not been calculated.

The minimum and maximum positions (cM) that produced a lod score
greater than the maximum - 1 ( 1.2772 - 1 )
and are contiguous to mibd.21.29.8 are:
15.4 29.8
```

7 Diagnostics

7.1 Theory

7.1.1 Testing for Normality

Since the polygenic model assumes that the trait values for each family ($\vec{Y_i}$) follow a multivariate normal distribution with mean $\mu_i = \vec{\mu} + X_i\beta$ and covariance matrix $\vec{V_i} = \sigma^2\vec{G_i} + \tau^2\vec{I}$, the normality assumption and presence of outliers need to be checked. Note that the proposed tests do not justify the normality assumption; they only detect significant departures from it.

Define $\vec{\widehat{Z}_i} = (\widehat{Z}_1, \widehat{Z}_2, ..., \widehat{Z}_{n_i})^T$ to be the standardized residual trait values

$$\widehat{Z_i} = \mathbf{V_i^{-1/\vec{2}}}(\mathbf{Y_i} - \widehat{\mu_i}),$$

where $\vec{\widehat{V_i}}$ the MLE of the covariance matrix for family i of size n_i,i = 1,2,..,k, and $\widehat{\mu_i} = \vec{\widehat{\mu}} + X_i\widehat{\beta}$ is the MLE of the mean. A normal probability plot can be constructed to visually check the normality assumption, and a residual plot can be constructed in an effort to identify outliers. This can be easily been done in `multic`.jj

Kurtosis can also be used as a measure of departure from the normality assumption. Allison et al.[2] showed that leptokurtic distributions (with kurtosis value (κ) > 0) have increased

Figure 5: *Default plot of a multic object. The trait is systolic blood pressure measured at three time points and the covariates are sex, age at exam, and age at exam^2 (measured at the three time points)*

false positive rates compared with the nominal values expected for normally distributed traits using sibpair data. However, Blangero et al. [14] suggested that distributions with $\kappa < 2$ could be reasonably analyzed under the assumption of a multivariate normal distribution for pedigree trait values. When $\kappa \geq 2$, it was recommended to apply a transformation to the trait, which is a common statistical procedure when the variable is not normally distributed, using a power transformation [Box and Cox, 1982; Cook and Wang, 1983]. These methods are readily available in Splus and R.

7.1.2 Empirical Normal Quantile Transformation

To reduce the impact of non-normality, one transformation to consider applying is the empirical normal quantile transformation. This transformation consists of the following steps:

1. Consider the traits values of j^{th} subjects in i^{th} families, y_{ij} , i=1,..,k; j=1,..,n_i.
2. Sort the y's and rank them (r_{ij}).
3. The transformation of y_{ij} is

$$y^*{}_{ij} = \Phi^{-1}\left(\frac{r_{ij}}{(1+\sum_{i=1}^{k} n_i)}\right)$$

where Φ^{-1} is the inverse of the cumulative function of the standard normal distribution. The value $(1 + \sum n_i)$ in the denominator is to assure that $y^*{}_{ij}$ is $< \infty$.

This transformation is also known as the van der Waerden normal scores. One way to think as an application to the QTL is that the correlation coefficient calculated using this

Figure 6: *Box-Cox transformation for triglycerides, using the covariates sex, age at exam, and age at* $exam^2$

transformation measures now the measure the similarity between phenotypic traits given their IBD status and not the ordinary linear correlation, i.e., $\rho(IBD) = \rho + \gamma(IBD - 1)$, IBD = 0,1,2. The null hypothesis is $\gamma = 0$. [33, 10].

7.1.3 Influence of outliers

We are not only interested in individual outliers or deviations from the normality assumption, but also whether one family is influencing the linkage analysis due to either family outliers or the lack of normality. To address the issue of family outliers, possible diagnostic tools include the family mean residuals and quadratic forms over families in polygenic models and lod scores over families in major gene models.

Diagnostic tools for the major gene model measure a family's likelihood contribution (L_i) to the total likelihood (L). These diagnostic measures include the family likelihood (L_i), the family log-likelihood ($log(L_i)$), and the family lod scores. Plots of these values for each family may indicate whether one or more families are greatly contributing to the total likelihood and thus warrants further investigation. If family i contains multiple outliers, L_i will be large and thus inflate the total likelihood value or lod score, resulting in misleading conclusions. Details about outliers can be found in de Andrade et al. [17]

7.2 Example

7.2.1 Normality and Outliers

In this example we examine four different traits: systolic blood pressure, diastolic blood pressure, BMI, and triglycerides. The first analysis step for model diagnostics is a visual examination of the data. Boxplots of the traits versus age groups shows why age^2 is

Figure 7: *Histogram of triglyceride using two different transformations*

important in the models (see figure 8). Note that there are several points that appear to be outliers. The kurtosis of triglycerides $\gg 2$, indicating that a transformation is necessary [14]. Using the `boxcox` function, $1/\sqrt{k.trig}$ is the appropriate transformation (see figure 6). Note that the Empirical Normal Quantile Transformation also does a good job at normalizing the data (figure 7).

```
## Box-Cox transformation for triglycerides

> library(MASS)
> boxcox(k.trig ~ male + agexam + agexam^2, data=d10, plotit=T)

## Look at the differences of the transformations
> par(mfrow=c(1,2))
> hist(1/sqrt(d10$k.trig))
> hist(t.rank(d10$k.trig))


## create age groupings from the variable agexam
> d10$agegp <- cut(d10$agexam, c(0,20,40,60,110), left.include=T,
                 labels=c('0-19','20-39','40-59','60+'))

par(mfrow=c(2,2)) ## plot using 2 rows and 2 columns
boxplot(split(d10$sys.avg, d10$agegp), ylab='Systolic Blood Pressure',
        xlab='Age Group',style.bxp='old')
boxplot(split(d10$dia.avg, d10$agegp), ylab='Diastolic Blood Pressure',
        xlab='Age Group',style.bxp='old')
boxplot(split(d10$bmi, d10$agegp), ylab='BMI', xlab='Age Group', style.bxp='old')
boxplot(split(d10$k.trig, d10$agegp), ylab='Triglycerides',
```

Figure 8: *Boxplots of the traits systolic blood pressure, diastolic blood pressure, BMI, and triglycerides for four age groups.*

```
        xlab='Age Group', style.bxp='old')

## calculate kurtosis for k.trig
> kurtosis(d10$k.trig, na.rm=T)
[1] 670.7128
```

Often we also want to look at the normality assumption of the residuals, which we can easily do using a quantile-quantile (QQ) plot. If the distribution of the residuals is the same as it would be for a standard normal, then the plot is approximately a straight line. In this particular example there are several extreme points and a heavier right tail that would be expected.

```
##
## First remember to fit the polygenic model with the option calc.residuals=T
##

> trig10 <- multic(famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                 k.trig ~ sex + agexam + agexam^2, data=d10, calc.residuals=T)

## Now extract the standardized residuals (average per family)

> tmp <- qqnorm(resid(trig10, type='standard'), ylab='Standardized Residuals')
> qqline(resid(trig10, type='standard'))

## define an outlier as being greater that 2 or less than -2
outliers <- abs(tmp$y) > 2
```

Figure 9: *Quantile-Quantile plot for the Triglyceride model.*

```
outliers[is.na(outliers)] <- F

## -- stick FAMID values on the plot
uni.famid <- unique(d10$famid)
text(x=1.01*tmp$x[outliers], y=tmp$y[outliers], uni.famid[outliers],
     cex=.7, adj=0)
```

Note that family 32 appears as on outlier in both the standardized residuals and Q-Q plots.

```
## Now look at the fitted values (average per family) and the Q1 residuals

> plot(fitted(trig10), resid(trig10, type='Q1'),
       xlab='Ave Fitted: k.trig ~ sex + agexam + agexam^2', ylab='Q1')


## This looks at the standardized residuals and fitted vaules
## without collapsing (i.e. 1 obs/person)

> sr.trig10 <- resid(trig10, type='standard', collapse.family=F)
> fit.trig10 <- fitted(trig10, collapse.family=F)

> plot(fit.trig10, sr.trig10, xlab='Fitted: k.trig ~ sex + agexam + agexam^2',
       ylab='Standardized Residual')
> abline(h=0)

## Again define an outlier as being further than 10 times the IQR (6 obs)
> outliers <- abs(sr.trig10) > 10*(quantile(sr.trig10, .75, na.rm=T) -
```

Figure 10: *Fitted values, averaged for each family, versus Q1 values for triglycerides using a model with the covariates sex, agexam, and* $agexam^2$.

```
                                    quantile(sr.trig10, .25, na.rm=T))
> outliers[is.na(outliers)] <- F

## Stick the family ID values on the plot
> text(x=1.01*fit.trig10[outliers], y=.98*sr.trig10[outliers], d10$famid[outliers],
       cex=.7, adj=0)
```

Now look at family 32 and see why there may be a problem.

```
####################
### PLOT FAMILY 32
####################

> library(kinship)

## create a pedigree object for family 32
> ok <- d10$famid==32
> ped32 <- pedigree(id=d10$id[ok], dadid=d10$fa[ok], momid=d10$mo[ok],
                    sex=d10$sex[ok])

# -- indicate those with high triglyceride levels in red
# -- place the triclyceride level underneath the subject ID

> high.trig <- ifelse(is.na(d10$k.trig) | d10$k.trig < 200, 1, 2)
> plot(ped32, id=paste(d10$id[ok], 'backslash n' , d10$k.trig[ok]), col= high.trig[ok])
```

Figure 11: *Individual fitted values versus standardized residuals for triglycerides using a model with the covariates sex, agexam, and* $agexam^2$.

Figure 12: *Pedigree plot of family 32 with triglyceride levels underneath the subject ID, with outliers of 1972 and 5173.*

Figure 13: *LOD Scores for the 5 families that contribute most to the peak LOD score when looking at triglycerides with the covariates sex, agexam and agexam*^{2}.

7.2.2 Influential Families

It is often of interest to determin the influence of individual families on the total LOD score. The `plot.family.lods` function selects the top (default 5) families that contribute to the overall lod score and plots what the lod score would be just using these families. It is important to remember to fit your multic model using the `calc.fam.log.liks=T` option.

```
> get.top.n.families(mult10, n=5)
     log.lik lod.score
 80 -59.6111 0.4172174
249 -29.5070 0.2408261
163 -27.3228 0.2212609
102 -46.6765 0.2027391
 45 -77.2895 0.1943043

> plot.family.lods(mult10, type="total")
[1] "80: 29%"  "249: 17%" "163: 15%" "102: 14%" "45: 13%"
```

8 Validation of Results

The jackknife and bootstrap methods are two approaches that can be used to validate linkage findings and are easy to run using functions available in S-Plusand R [24]. Both techniques involve sampling from the existing data and recalculating a statistic in order to obtain an estimate of the variability of the statistic. The jackknife is an approximation of the bootstrap, but is computationally less intense. In our setting, sampling is done on the family unit instead

of the individual and we're most interested in the estimate of the overall lod score at a specific location.

The jackknife technique removes a subset of the families and calculates the statistic. This process is repeated for each subset (often removing 1 family at a time). The bootstrap technique samples with replacement from the existing families and calculates the statistic. Each subsample typically includes roughly two-thirds of the families. This step is typically repeated some large number of times (100 or 500). Below are examples showing how to implement these concepts. The examples focus on chromosome 10 where a lod score of 1.45 was found at 177 cM. We can obtain estimates of the standard error of the lod score using either approaches.

8.1 Jackknife

In our jackknife example we looked at a small region of chromosome 10 ranging from 175-179 cM. We removed one family at a time and ran multic using the remaining families.

```
##############################################################################
##             Jackknife example
##############################################################################

## cut the IBD file (mloci.out) so that only a small region is used
> mlociCut('multicInput/mloci.out', 175:179,'multicInput/mloci.cut177')

## create a blank matrix to store results
jacklods <- matrix(NA, nrow=length(unique(d10$famid)), ncol=5,
                   dimnames = list(1:length(unique(d10$famid)),
                   c('lod175','lod176','lod177','lod178','lod179'))

## remove one family at a time and rerun the results
> for(i in 1:length(unique(d10$famid)))

  fam.del <- d10$famid != unique(d10$famid)[i]
  fit <- multic(sys.avg ~ sex + agexam + agexam^2, data=d10,
                famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                mloci='multicInput/mloci.cut177.gz', share='multicInput/share.out.gz',
                max.iterations=30, subset=fam.del)
  jacklods[i,1:4] <- fit$log.liks[-1, 4]



## Plot the results
> matplot(175:179, t(jacklods), type='l', xlab='cM',
          ylab='LOD, Jackknife Analysis', xlim=c(175,179), ylim=c(0,2))
> plot(sys10, xlim=c(175,179), ylim=c(0,2), ylab='LOD, Original Analysis')
> hist(jacklods[,3], 'fd', xlab='LOD, 177 cM')
> abline(v=sys10$log.liks$lod.score[sys10$log.liks$distance==177], lwd=2)
```

Figure 14: *Jackknife results for the model sys.avg∼sex + agexam + agexam^2 focusing on the region 175 to 179 cM. Each one of the 179 families was removed from the data, one at a time, and the model was refit. The upper left panel includes the lod results for each deletion. The upper right panel shows the results from the original model. The lower left figure shows a histogram for position 177 cM, with a vertical line indicating the result when using all the families at once.*

8.2 Bootstrap

We used two different approaches to obtain bootstrapped results. The first approach bootstraps the families and runs multic on each of these sets of families to obtain the overall lod. The second approach fits `multic` once using the option `calc.fam.log.liks = T`, samples from the family lods obtained from that one call and adds up the family lods to obtain an overall lod score for each sample.

The two approaches produce different distributions, as can be seen in figures 15 and 16. In the second approach where the bootstrap is performed using the lod for each family at a particular position, the estimated parameters values are the same for each family and for each bootstrap run. Thus the bootstrap distribution of the lod values will follow a normal distribution according to the bootstrap theory. However, when the bootstrap is performed by random selecting families and then calculating the lod at a particular position, the estimated parameter values will change for each bootstrap run. Thus, the distribution of the lod will follow a mixture of $1/2\ (\chi^2_0 + \chi^2_1)$ as expected but the distribution of the parameters estimates will follow a normal distribution. The empirical distribution of these two boostrap approaches are similar but slightly different at the mean due to skewness difference of these two distributions (1.64 for method 1; 1.44 for method 2). The medians (1.43 for method 1 and 1.42 for method 2) are much more similar. Using these two approaches, it is possible to better understand different parts of the variation within the modeling process.

##

Figure 15: *Bootstrap results for the model* $sys.avg \sim sex + agexam + agexam^2$ *focusing on the location 177 cM. In this approach, families were sampled with replacement, and multic was run 1,000 times.*

Figure 16: *Bootstrap results for the model* $sys.avg \sim sex + agexam + agexam^2$ *focusing on the location 177 cM, where the lod scores were saved for each family. In this second approach, family lod scores were sampled with replacement, and the overall lod score was recalculated 10,000 times.*

```
## Subset IBD to one location
> mlociCut('multicInput/mloci.out', c(177,177), 'multicInput/mloci.cut177')


###########################################################################
### APPROACH NUMBER 1 - TRADITIONAL BOOTSTRAP SAMPLING OF THE FAMILIES

## Create a function that is executed for each bootstrap sample.
> tfun <- function(newids)

    expand.multic(newids, mloci.out='multicInput/mloci.cut177',
                          share.out='multicInput/share.out')
    newdata <- expand.data(newids,d10)

    tmpfit <- multic(sys.avg ~ sex + agexam + agexam^2, data=newdata,
                     famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                     mloci='mloci.cut177.expanded',
                     share='share.out.expanded', max.iterations=100)
    ## store lod, major gene, polygene, and environmental variance estimates
    ans <- cbind(tmpfit$log.liks[2,4], tmpfit$major.gene1[1,1,2],
                 tmpfit$polygenic[1,1,2], tmpfit$environmental[1,1,2])
    return(ans)


## Set up an empty matrix to store the results
> set.seed(1001)
> Nboot <- 1000
> bootresults <- matrix(NULL, nrow=Nboot, ncol=4,
                  dimnames=list(1:Nboot, c('LOD','mg1','poly','env')))

## Loop through the multiple calls to tfun
> for(i in 1:Nboot)
     newids <- sample(unique(d10$famid), length(unique(d10$famid)), replace=T)
     bootresults[i,] <- tfun(newids)


## Plot the bootstrap results
> bandwidth <- function(x)
     r <- quantile(x, c(.25, .75))
     h <- (r[2] - r[1])/1.34
     return(4*1.06*min(sqrt(var(x,na.method='omit')), h) * length(x)^-1/5)


> hist(bootresults[,1], 'fd',xlab='Overall LOD', ylab='Density',
       xlim=c(-2,6), ylim=c(0,.5), prob=T)
> lines(density(bootresults[,1], width=bandwidth(bootresults[,1])))
> abline(v=sum(lods), lwd=2)  ## observed
> abline(v=mean(bootresults[,1]), lty=2, lwd=2) ## mean
> key(corner=c(1,1), lines=list(lty=1:2), text=list(c('Observed','Mean')))

## Summarize the results
> quantile(bootresults[,1],c(.025,.975))
```

```
  2.5%  97.5%
0.1708 4.1071
```

For 1000 samples, this first approach took 4 1/2 hours to run. The second approach, shown below, took only a few minutes.

```
##############################################################################
### APPROACH NUMBER 2 - BOOTSTRAP THE FAMILY LODS AND SUM TOTAL LOD

> origfit <- multic(sys.avg ~ sex + agexam + agexam^2, data=d10,
                    famid=famid, id=id, dadid=fa, momid=mo, sex=sex,
                    calc.fam.log.liks = TRUE,
                    mloci='multicInput/mloci.cut177',
                    share='multicInput/share.out')

> lods <- origfit$fam.log.liks[,2,2]

## run 10,000 bootstraps for the family lod values

> library(resample,first=T)
> boot.orig <- bootstrap(lods, sum, B=10000)
> plot(boot.orig, main=' ', xlab='Overall LOD')
> summary(boot.orig)

Call:
bootstrap(data = lods, statistic = sum, B = 10000)

Number of Replications: 10000

Summary Statistics:
    Observed  Mean      Bias   SE
sum    1.446 1.439 -0.007091 1.03

Percentiles:
       2.5%      5%   95% 97.5%
sum -0.5044 -0.2274 3.176 3.515

BCa Confidence Intervals:
       2.5%      5%   95% 97.5%
sum -0.4744 -0.1958 3.211 3.554
```

9 Time tests

We designed a series of tests to compare the performace of multic using Splus and R and using two platforms, a Unix Sunfire V440 with four 1.593GHz UltraSparceIIi processors, 8G of memory, and 2 73G ultra320 SCSI disks versus a Mac OS X version 10.3.9 with a 1.5 GHz PowerPC G4 processor and 512 MB DDR DSRAM of memory. Several conditions were chosen for two datasets including:

1. GAW13 simulated data, Chromosome 21, replicate 003
    - Large (35-84) member families, 11 families

N.traits	Region	Covar.Used	Families	Dataset	cpu	ratio
1	Small	No	Large	GAW14	56	1.0
2	Small	No	Large	GAW14	837	14.9
3	Small	No	Large	GAW14	7382	131.8
1	Small	Yes	Large	GAW14	44	0.8
2	Small	Yes	Large	GAW14	950	17.0
3	Small	Yes	Large	GAW14	5861	104.7
3	Small	Longitudinal	Large	GAW14	4949	88.4
1	Small	Yes	Small	Genoa	86	1.5
2	Small	Yes	Small	Genoa	1548	27.6
3	Small	Yes	Small	Genoa	10796	192.8
1	Large	No	Large	GAW14	582	10.4
2	Large	No	Large	GAW14	3025	54.0
3	Large	No	Large	GAW14	22267	397.6
1	Large	Yes	Large	GAW14	635	11.3
2	Large	Yes	Large	GAW14	4218	75.3
3	Large	Yes	Large	GAW14	20357	363.5
1	Large	Yes	Small	Genoa	850	15.2
2	Large	Yes	Small	Genoa	12598	225.0
3	Large	Yes	Small	Genoa	98647	1761.6

Table 1: *Time test results using Splus on the Solaris system. Results using R on the Solaris system were similar and results on the Mac were approximately 35% less. The column marked ratio provides some indication of the relative time comparable to the first line in the table.*

- Moderate (10-20) member families, 142 families
- Small region: 15 - 30 cM (7 locations)
- Large region: 1 - 58 cM (27 locations, 3.9 as large as the small region)

2. GENOA data, Chromosome 1

   - Small (3-16) sibling families, 400 families
   - Small region: 80 - 100 cM (24 locations)
   - Large region: 1 - 275 cM (328 locations, 13.7 as large as the small region)

Our general conclusions were that platform is the most important factor. Multic run via Splus or R on the Unix system took a similar amount of CPU. Multic run on the Mac took $\approx$ 35% less time than on our Unix system. The time for 502 subjects from 11 large families was equivalent to the time for 2382 subjects from 142 medium families. It also appears that running the correct model with covariates is slightly faster (6%) than a model without covariates. Two traits took approximately twelve times (6-15 IQR) the CPU that was used for one trait. Three traits were took approximately 90 times (35-115 IQR) more CPU than one trait. Finally, the time for 3-trait longitudinal analysis is approximately 85% the time for running a 3 trait multivariate analysis.

Selected results using Splus are presented in table 1. The final column indicates a ratio of time compared with the model with no covariates using one trait for a small region with large families.

10 Future Directions

As is true with most software development, the package is never really completely finished. We have already started a list of enhancements that we intend to add to the multic library. There have been several requests to add in two other estimation methods including the Newton-Raphson Method (using Maxfun already available in the stand-alone version of multic) [46] and the Expectation-Maximization Algorithm (using code from the EMVC package) [30]. We are also planning to make the package more flexible by directly importing IBDs calculated from Merlin, Genehunter, and others. We will investigate the possibility of directly calculating the IBDs using Splus. In addition to adding in connections to and from other packages, we are planning to continue working on model diagnostics and helpful summary functions. Other requests include the ability to specify the model covariance structure for longitudinal data. Finally, we will investigate the possibility of using parallel processing to improve the analysis speed.

11 Function helpfiles

`multic` *Create a multic object*

Description

Calculate the polygenic and major gene models for quantitative trait linkage analysis using variance components approach.

Usage

```
multic(formula, data = sys.parent(), famid, id, dadid, momid, sex,
       mloci.out = NULL, share.out = "kinship", longitudinal = FALSE,
       subset = NULL, ascertainment = NULL,
       control = multic.control(...), ...)
```

Arguments

`formula`	a formula object, with the traits on the left of a `~` (tilde) operator and the covariates, separated by `+` operators, on the right. The traits may be a single numeric vector or a matrix. Commonly, traits are aggragated together using the `cbind` command. See the Examples section for examples.

`famid` integer, numeric, or character vector specifying each individual's family identifier. Members of the same family must have the same `famid` and each family must have a unique `famid`. Any missing data will result in an error message and the termination of `multic`.

`id` integer, numeric, or character vector specifying each individual's identifier. Members of the same family must have a unique `id` within the family. `id` does not have to be universally unique among all individuals. Any missing data will result in an error message and the termination of `multic`.

`dadid` integer, numeric, or character vector specifying each individual's father identifier. This father identifier must have the same `famid` as the individual. Any missing data will result in an error message and the termination of `multic`.

`momid` integer, numeric, or character vector specifying each individual's mother identifier. This mother identifier must have the same `famid` as the individual. Any missing data will result in an error message and the termination of `multic`.

`sex` integer, numeric, or character vector specifying each individual's sex. Acceptable forms of sex-coding are `"M"`, `"m"`, or `1` for male and `"F"`, `"f"`, or `2` for female. Any missing data will result in an error message and the termination of `multic`.

`data` `data.frame` in which to interpret the variables named in `formula`, `famid`, `id`, `dadid`, `momid`, `sex`, `subset`, and `ascertainment` arguments. If `data` is missing, the variables in `formula` should be in the search path.

`mloci.out` a character value specifying a path to an mloci.out (or similarly formatted) file. Specifying a non-empty mloci.out file will allow multic to calculate sporadic models using the ibd (identity by decent) information in the mloci.out file. Due to the general size of mloci.out, it is often stored in .gz format. `multic` will manage this for the user. Whether the user specifies an mloci.out file with a .gz suffix or not will not effect how `multic` operates on the file. See `solar2mloci` for more details.

`share.out` a character value specifying a path to a share.out (or similarly formatted) file. This file contains the amount of genetic material shared between each family member pair based on family structure only. It also contains boolean values to indicate whether two familiy members have a sibling-sibling, parent-parent, or parent-offspring relationships. Due to the general size of share.out, it is often stored in .gz format. `multic` will manage this for the user. Whether the user specifies the file wiht a .gz suffix format or not will not effect how `multic` operates on the file. See `phi2share` for more details.

`longitudinal` logical flag: if `TRUE`, then fomula will be interpreted as a longitudinal model. In this case, the fomula argument requires special formatting as described. The number of traits on the left side of the `~` (tilde) is the number of time-points for `multic` to analyze. The number of covariates on the right side of the `~` (tilde) must be a multiple of the number of traits on the left side. That multiple is the amount of covariates to analyze at each

	time-point. All covariates for the first time point must be specified before any of the second, all second before any third, etc. See the Examples section for examples.
`subset`	a logical vector specifying which subset of the rows in `data` to use in the fit.
`ascertainment`	vector specifying each individual's ascertainment (effected) status. Acceptable forms of ascertainment are `T`, `TRUE`, or `1` for a proband (effected) and `F`, `FALSE`, or `0` for a non-proband (non-effected person).
`control`	`list` of iteration and algorithmic constants. See `multic.control` for their names and default values. These can also be given directly as arguments to `multic` itself, instead of through a `multic.control` object. If control is specified, the arguments specified in `...` will not be used.
`...`	further arguments passed to `multic.control` to alter `multic`'s default behavior.

Details

See the technical report.

Value

an object of class `"multic"`. See `multic.object` for more details.

Side Effects

Many temporary files are created during `multic`'s execution. These files are deleted afterwards (by default). If they are not deleted (due to a crash or some other unexpected action), use the included function clean() to delete them. Also, `multic` copies, gunzip's, and removes the copies of share.out and mloci.out (if specified).

References

Amos, C. I. (1994). "Robust variance-components approach for assessing genetic linkage in pedigrees." American Journal of Human Genetics 54(3): 535-543.

Almasy, L. and J. Blangero (1998). "Multipoint quantitative-trait linkage analysis in general pedigrees." American Journal of Human Genetics 62(5): 1198-1211.

See Also

`multic.object`, `multic.control`, `phi2share`, `solar2mloci`, `solar2multic`, `sw2mloci`

Examples

```
# Call multic with a univariate formula with two covariates and no
# markers (no mloci.out argument).
fit.ibd.uni <- multic(k.trig ~ sex.x + agexam,
                      data = ped.phen.data,
                      famid, id, fa, mo, sex.x,
                      share.out = 'multicInput/share.out')
```

```
# Call multic with a bivariate formula with three covariates, no
# markers (no mloci.out argument), and calculate the family log
# likelihoods.
fit.ibd.bi <- multic(cbind(k.trig, k.chol) ~ sex.x + agexam + agexam2,
                     data = ped.phen.data,
                     famid, id, fa, mo, sex.x,
                     share.out = 'multicInput/share.out',
                     calc.fam.log.liks = TRUE)

# Call multic with a longitudinal formula with six covariates letting
# the kinship library calculate the share.out argument.
long.fit <- multic(cbind(sbpA, sbpB, sbpC) ~
                     sexA + ageA + bmiA + generA + ageAg + smkA +
                     sexB + ageB + bmiB + generB + ageBg + smkB +
                     sexC + ageC + bmiC + generC + ageCg + smkC,
                   data = long.data,
                   famid, id, dadid, momid, sex,
                   longitudinal = TRUE)
```

`multic.object` *a multic object*

Description

Object of class `"multic"` returned from the function `multic`.

Arguments

`fam.log.liks`	the log likelihoods and lod scores for each family at each marker (including the null hypothesis). `fam.log.liks` is a 3-dimensional matrix. The first dimension is indexed by the family identifiers. The second dimension is indexed by the words `"log.lik"` and `"lod.score"`. The third dimension is indexed by the word `"null"` and the names of the marker file names. To calculate the family log likelihoods, `calc.fam.log.liks = TRUE` must be passed to `multic` via the `...` parameter or a `multic.control` object. If `fam.log.liks` are not calculated, then `fam.log.liks` is a character vector providing instructions how to calculate the values.
`fixed.effects`	the estimate, standard error, t value, and p value of the fixed effects for the traits and covariates for the null hypothesis and each marker. `fixed.effects` is a 3-dimensional matrix. The first dimension is indexed by the trait and covariate names. The second dimension is indexed by the words `"Estmate"`, `"Std.err"`, `"t.value"`, and `"p.value"`. The third dimension is indexed by the word `"null"` and the marker file names.
`polygenic`	the estimate, standard error, Wald score, Wald score P-value, heritabilty estimate, standard error of the heritabilty estimate, and heritably estimate P-value for the variance and covariance of the polygenic effect of

the formula for the null hypothesis and each marker. `polygenic` is a 3-dimensional matrix. The first dimension is indexed by the letter `"s"` followed by a `1`, `2`, etc. for the variance of the first trait, second trait, and so on or `12`, `13`, `23`, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words `"Estimate"`, `"Std.err"`, `"Wald"`, `"W.p.value"`, `"h^2"`, `"se.h^2"`, and `"h.p.value"`. The third dimension is indexed by the word `"null"` and the marker file names.

`major.gene1` the estimate, standard error, Wald score, Wald score P-value, heritabilty estimate, standard error of the heritabilty estimate, and heritably estimate P-value for the variance and covariance of the major gene effect of formula for the null hypothesis and each marker. `major.gene1` is a 3-dimensional matrix. The first dimension is indexed by the letters `"mg"` followed by a `1`, `2`, etc. for the variance of the first trait, second trait, and so on or `12`, `13`, `23`, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words `"Estimate"`, `"Std.err"`, `"Wald"`, `"W.p.value"`, `"h^2"`, `"se.h^2"`, and `"h.p.value"`. The third dimension is indexed by the word `"null"` and the marker file names.

`environmental` the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the environmental effect of formula for the null hypothesis and each marker. environmental is a 3-dimensional matrix. The first dimension is indexed by the letter `"e"` followed by a `1`, `2`, etc. for the variance of the first trait, second trait, and so on or `12`, `13`, `23`, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words `"Estimate"`, `"Std.err"`, `"Wald"`, and `"W.p.value"`. The third dimension is indexed by the word `"null"` and the marker file names.

`sibling.sibling`
the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the sibling to sibling effect of formula for the null hypothesis and each marker. `sibling.sibling` is a 3-dimensional matrix. The first dimension is indexed by the letters `"sib"` followed by a `1`, `2`, etc. for the variance of the first trait, second trait, and so on or `12`, `13`, `23`, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words `"Estimate"`, `"Std.err"`, `"Wald"`, and `"W.p.value"`. The third dimension is indexed by the word `"null"` and the marker file names. To receive valuable data, the 5th member of `constraints` in the `multic.control` object must be set to not `"F"` (fixed).

`parent.parent`
the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the parent to parent effect of formula for the null hypothesis and each marker. `parent.parent` is a 3-dimensional matrix. The first dimension is indexed by the letter `"p"` followed by a `1`, `2`, etc. for the variance of the first trait, second trait, and so on or `12`, `13`, `23`,

etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", and "W.p.value". The third dimension is indexed by the word "null" and the marker file names. To receive valuable data, the 6th member of constraints in the multic.control object must be set to not "F" (fixed).

parent.offspring
: the estimate, standard error, Wald score, and Wald score P-value for the variance and covariance of the parent to offspring effect of formula for the null hypothesis and each marker. parent.offspring is a 3-dimensional matrix. The first dimension is indexed by the letter "q" followed by a 1, 2, etc. for the variance of the first trait, second trait, and so on or 12, 13, 23, etc. for the covariance between the first and second traits, first and third traits, second and third traits, and so on. The second dimension is indexed by the words "Estimate", "Std.err", "Wald", and "W.p.value". The third dimension is indexed by the word "null" and the marker file names. To receive valuable data, the 7th member of constraints in the multic.control object must be set to not "F" (fixed).

log.liks
: the log likelihood, centimorgan distance, log likelihood status, and lod score and P-value for the null hypothesis and each marker. log.liks is a data.frame. The row names are "null" and the markder file names. The column names are "log.likelihood", "distance", "log.lik.status", "lod.score", and "p.value". The log likelihood status represents whether the log likelihood converged before the maximum interations allowed or not and have the values of either "converg" or "non-converg".

var.fixed
: the variance of the fixed effects of the traits and covariates for the null hypothesis and each marker. var.fixed is a 3-dimensional matrix. The first and second dimensions are indexed by the trait and covariate names. The third dimension is indexed by the word "null" and the marker file names.

var.random
: the variance of the polygenic, major gene, and environmental effects for the null hypothesis and each marker. var.random is a 3-dimensional matrix. The first and second dimensions are indexed as described by the polygenic, major.gene1, and environmental components above. The third dimension is indexed by the word "null" and the marker file names.

var.sandwich
: a more precise variance estimator after using a sandwich estimator approach. This is only calculated if the multic object represents a univariate model. var.sandwich is a 3-dimensional matrix. The first and second dimensions are indexed by "s1", "mg1", and "e1". The third dimension is indexed by the word "null" and the marker file names.

cors
: the Pearson, Spearman, genetic, environmental, and phenotypic correlations. cors is a list made up of the components "pearson", "spearman", "genetic", "environment", and "phenotype". Both "pearson" and "spearman" are their respective correlations between the traits and covariates. They are 2-dimensional matrices indexed by the trait and covariate names. "genetic", "environment", and "phenotype" are the respective correlations between the polygenic and environmenal estimates. They

are 2 dimensional matrices. The first dimension is indexed by the word "null" and the marker file names. The second dimension is indexed as described by the covariance portions of the polygenic and environmenal components above.

v.matrices — the variance-covariance matrix of the trait (y) that incorporates the polygenic, major gene, shared common environment, and error matrices. v.matrices is a 2-dimensional matrix. The first dimension is indexed by the family identifier (famid) values. The second dimension is indexed by the word "null" and the marker file names. Currently, there are no individual identifiers on each of the V matrices. If the V matrices are not calculated, then v.matrices is a character vector providing instructions how to calculate the values.

residuals — the observed values minus the fitted values of the trait (y) divided by the square root of the V matrix for each family. If the residuals are not calculated, then residuals is a character vector providing instructions how to calculate the values.

descriptives — the total individuals used, mean, standard deviation, minimum, maximum, kurtosis, and skewness for each trait and covariate.

counts — various counts of the total number of pedigrees, people, females, males, and so on. This is mostly for passing data for print and summary to display and is very likely to be not useful to the user community.

call — how multic was called. call is a call object.

R.sq — the proportion of variance due to the covariates.

metadata — a list of useful data like start.time, finish.time, call, epsilon, trait.count, iterations, null.initial.values, method, etc.

Generation

This class of objects is returned by the multic function to represent a fitted variance components model.

Methods

Objects of this class have methods for the functions polygene, print, plot, fitted, residuals, and summary

See Also

multic

Description

Allows users to alter the default behavior of multic

Usage

```
multic.control(epsilon = 1e-5,
               max.iterations = 50,
               boundary.fix = TRUE,
               constraints = c("E", "E", "E", "E", "F", "F", "F"),
               initial.values = NULL,
               save.output.files = FALSE,
               method = c("multic", "leastsq", "maxfun", "emvc"),
               calc.fam.log.liks = FALSE,
               calc.residuals = FALSE,
               keep.input = calc.residuals)
```

Arguments

`epsilon` a numeric value specifying the convergence threshold. When the difference of an iteration's loglikelihood and the previous iteration's loglikelihood are less than `epsilon`, the value has "converged".

`max.iterations` an integer value specifying the maximum number of iterations `multic` will take to converge during the polygenic and sporadic model calculations.

`boundary.fix` logical flag: if `TRUE`, then the variances generated will be fixed to `0` and no longer estimated when they become less than `0.00001` (`1e-5`).

`constraints` a character vector of length seven (7) specifying the constraints on the random effects variance components. Each value of the vector needs to be either `"E"` - 'E'stimate the variance and covariance, `"C"` - estimate the variance and 'C'onstrain the covariance, or `"F"` - 'F'ix the variance and covariance to `0`. Each index of constraints corresponds to (in this exact order) mu, polygene, major gene, environment, sibling-sibling, parent-parent, and parent-offspring.

`initial.values` numeric vector: use the specified initial values instead of calculating them automatically. This vector has a very specific length and order. If n is the number of traits and m is (n + (n-1) + (n-2) + ... + 1), then the length must be n + 6 * m. So for one trait (univariate), the length must be 7, for two traits (bivariate), 20, and so on. The position of the values in the vector is important as well. The first n terms are the mu starting values. The next starting values come in chunks of m. The next m values are the polygenic starting values, followed by major gene, environmental,

sibling-sibling, parent-parent, and parent-offspring starting values. The metadata$null.initial.values contains the placement of the starting values. You can use this to verify your order is correct.

`save.output.files`
: logical flag: if `TRUE`, then the multiple temporary output files `multic` generates are not removed. This is mostly for debugging purposes and is very likely to be not useful to the user community.

`method`
: a character value specifying the method to use in fitting the model. Possible values include `"multic"` (default), `"leastsq"`, `"maxfun"`, and `"emvc"` (all case insensitive).

`calc.fam.log.liks`
: logical flag: if `TRUE`, then the family log likelihoods will be returned in the `multic` object. WARNING: This significantly increases the size of the returned `multic` object.

`calc.residuals`
: logical flag: if `TRUE`, then the residuals will be calculated and Y beta differences and V matrix data will be returned in the `multic` object. WARNING: This dramatically increases the size of the returned `multic` object.

`keep.input`
: logical flag: if `TRUE`, then the traits and covariates will be saved in the `metdata` list of the `multic` object. Since the input is needed during special residual calculations, its default value is that of `calc.residuals`.

Value

a list that is designed to be supplied as a control argument to `multic`. The values for `multic.control` can be supplied directly in a call to `multic` (via the `...` parameter). These values are then filtered through `multic.control` inside `multic`.

See Also

`multic`, `multic.object`

Examples

```
## The following calls to multic are equivalent
multic(formula, data, control = multic.control(calc.fam.log.liks = TRUE,
                                               calc.residuals = TRUE))
multic(formula, data, calc.fam.log.liks = TRUE, calc.residuals = TRUE)
```

`print.multic` *Use print() on a multic object*

Description

This is a method for the function print() for objects inheriting from class `multic`. See print or print.default for the general behavior of this function and for the interpretation of x.

Usage

```
print.multic(x, ...)
```

Arguments

x	a multic object
...	additional parameters to alter the default behavior of print.multic. Currently ... only exists to pass 'R CMD check' tests.

`plot.multic` *Plot a multic object*

Description

This is a method for the function plot() for objects inheriting from class `multic`. See plot or plot.default for the general behavior of this function and for the interpretation of x.

Usage

```
plot.multic(x, ...)
```

Arguments

x	a `multic` object
...	additional arguments like xlim, ylab, etc.

`solar2multic` *Convert SOLAR-formatted output into multic-formatted mloci.out and share.out*

Description

`solar2multic` is a utility function to convert the ibd and mibd files (identity by descent) files created by SOLAR into the multic input file mloci.out and convert the phi2 created by SOLAR into the multic input file share.out.

Usage

```
solar2multic(phi2, pedigree.file, pedindex.out, pedindex.cde,
             ibd.directory, ibd.dist = NULL, output.directory = ".",
             delete.fixed.dir = TRUE)
```

Arguments

`phi2`	character value specifying a path to a SOLAR-formatted phi2 file. Due to the general size of a typical phi2 file, it is often stored in .gz format. `solar2multic` will manage this for the user. Whether the user specifies the file with a .gz suffix or not will not effect how `solar2multic` operates on the file.
`pedigree.file`	character value specifying a path to a SOLAR-formatted pedigree structure file (.ped). This file must have a header of `famid`, `id`, `fa`, `mo`, and `sex` (case insensitive). The file must also be comma separated.
`pedindex.out`	character value specifying a path to a SOLAR-formatted pedindex.out file. This file must be the same that was output from SOLAR. It provides a mapping between the sequential number system assigned by SOLAR and the original family and individual identifiers.
`pedindex.cde`	character value specifying a path to a SOLAR-formatted pedindex.cde file. This file must be the same that was output from SOLAR. This file describes how pedindex.out is organized. This is necessary to read pedindex.out correctly.
`ibd.directory`	character value specifying a path to a directory containing SOLAR-formatted ibd and/or mibd files.
`ibd.dist`	charater value specifying a path to a SOLAR-formatted .dist file that maps the character marker names to numeric centimorgan values.
`output.directory`	character value specifying a path to a directory that the output files (mloci.out and share.out) will be placed. If any of the specified directory path does not exist, `solar2multic` will create the necessary directories.
`delete.fixed.dir`	logical flag: if `TRUE` (default), then the temporary directory that is created to hold intermediate files is deleted.

Side Effects

Due to write permissions possibly not allowing the user to gunzip and create files in the specified `directory`, `solar2multic` first copies `directory` and `phi2` to the current directory. `solar2multic` then creates a temporary directory to hold the "fixed," intermediate files that will be deleted (by default). Also, `solar2multic` will overwrite mloci.out, mloci.out.gz, share.out and share.out.gz if they exist in `output.directory`.

See Also

`solar2mloci`, `phi2share`

Examples

```
solar2multic(phi2 = "phi2.gz",
             pedigree.file = "chrom18.ped",
             pedindex.out = "pedindex.out",
```

```
            pedindex.cde = "pedindex.cde",
            ibd.directory = "mibddir",
            output.directory = "multicInput",
            delete.fixed.dir = FALSE)

solar2multic("solarOutput/phi2",
            "solarOutput/chrom10.ped",
            "solarOutput/pedindex.out",
            "solarOutput/pedindex.cde",
            "mibds/chrm10"))
```

`phi2share` *Convert a SOLAR-formatted phi2 file into a multic-formatted share.out file.*

Description

`phi2share` reads in the specified files and generates a multic-formatted share.out file. share.out contains unique identifiers, expected shared genetic material, and sibling, spousal, and parent-offspring true/false values.

Usage

```
phi2share(phi2, pedigree.file, pedindex.out, pedindex.cde,
          output.directory=".")
```

Arguments

`phi2`	a character value to specify the location of a SOLAR-formatted phi2 (or similarly formatted) file. Due to the general size of a typical phi2 file, it is often stored in .gz format. `phi2share` will manage this for the user. Whether the user specifies the file with a .gz suffix or not will not effect how `phi2share` operates on the file.
`pedigree.file`	a character value to specify the location of a .ped (or similarly formatted) file. This file must have a header of `famid`, `id`, `fa`, `mo`, and `sex` (case insensitive). The file must also be comma separated.
`pedindex.out`	a character value to specify the location of a pedindex.out file. This file must be the same that was output from SOLAR. It provides a mapping between the sequential number system assigned by SOLAR and the original family and individual identifiers.
`pedindex.cde`	a character value to specify the location of a pedindex.cde file. This file must be the same that was output from SOLAR. This file describes how pedindex.out is organized. This is necessary to read pedindex.out correctly.
`output.directory`	a character value specifying which directory to place the output share.out. If `output.directory` (including supporting path) does not exist yet, it will be created. The default directory is the current directory.

Side Effects

`phi2share` creates a local copy of, gunzip's, and removes the copy of `phi2`. It also will overwrite share.out and share.out.gz if they exist in `output.directory`.

See Also

`solar2mloci`, `solar2multic`

Examples

```
phi2share(phi2 = "phi2",
          pedigree.file = "chrom18.ped",
          pedindex.out = "pedindex.out",
          pedindex.cde = "pedindex.cde",
          output.directory = "multicInput")
phi2share("solarOutput/phi2.gz",
          "solarOutput/chrom18.ped",
          "solarOutput/pedindex.out",
          "solarOutput/pedindex.cde")
```

`solar2mloci` *Convert a directory of SOLAR-formatted ibd and/or mibd files into a multic-formatted mloci.out*

Description

`solar2mloci` reads all of the ibd and mibd files in the given directory, and creates mloci.out.gz in the specified output directory.

Usage

```
solar2mloci(directory, phi2, pedindex.out, pedindex.cde,
          ibd.dist = NULL, output.directory = ".",
          delete.fixed.dir = TRUE)
```

Arguments

`directory`	charater value specifying a path to a directory of SOLAR-formatted ibd and/or mibd files. These files are often kept in .gz format. `solar2mloci` will manage this for the user.
`phi2`	charater value specifying a path to a SOLAR-formatted phi2 file. Due to the general size of a typical phi2 file, it is often stored in .gz format. `solar2mloci` will manage this for the user. Whether the user specifies the file with a .gz suffix or not will not effect how `solar2mloci` operates on the file.

`pedindex.out` character value specifying a path to a SOLAR-formatted pedindex.out file. This must be the same file that was output from SOLAR. It provides a mapping between the sequential number system assigned by SOLAR and the original family and individual identifiers.

`pedindex.cde` character value specifying a path to a SOLAR-formatted pedindex.cde file. This must be the same file that was output from SOLAR. This file describes the format of pedindex.out. This is necessary to read pedindex.out correctly.

`ibd.dist` charater value specifying a path to a SOLAR-formatted .dist file that maps the character marker names of ibd files to their corresponding numeric centimorgan values.

`output.directory` charater value specifying a path to a directory where the output file mloci.out.gz will be created. If the directory (including supporting path) does not exist yet, it will be created. Also, `solar2mloci` will overwrite mloci.out and mloci.out.gz if they exist in `output.directory`.

`delete.fixed.dir` logical flag: if `TRUE`, delete the temporary directory used to "fix" the SOLAR-formatted ibds and mibd files. This is mostly for debugging purposes and is very likely to be not useful to the user community.

Side Effects

Due to write permissions possibly not allowing the user to gunzip and create files in the specified `directory`, `solar2mloci` first copies `directory` and `phi2` to the current directory. `solar2mloci` then creates a temporary directory to hold the "fixed," intermediate files that will be deleted (by default). Also, `solar2mloci` will overwrite mloci.out and mloci.out.gz if they exist in `output.directory`.

See Also

`phi2share`, `solar2multic`

Examples

```
solar2mloci(directory = "ibddir", phi2 = "phi2",
          pedindex.out = "pedindex.out", pedindex.cde = "pedindex.cde",
          ibd.dist = "solar.dist", output.directory = "multicInput",
          delete.fixed.dir = FALSE)

solar2mloci("mibds/chrm10", "solarOutput/phi2.gz",
          "solarOutput/pedindex.out", "solarOutput/pedindex.cde")
```

`sw2mloci` *Convert SimWalk IBD files into a single mloci.out.*

Description

sw2mloci converts all IBD files in a given directory into a sinlge mloci.out needed my multic, altering the centimorgan values if a map argument is provided.

Usage

```
sw2mloci(directory, map="", output.directory=".")
```

Arguments

`directory`	a character object specifying the name of the directory that contains the SimWalk IBD files. This can be an absolute or relative path.
`map`	a character object specifying the name of a .map file to be used to modify the centimorgan values in the mloci.out file. This can be an absolute or relative path and does not have to be in the same directory as the parameter `directory`.
`output.directory`	a character object specifying the name of the directory to put the finished mloci.out.gz. This can be an absolute or relative path. If the directory does not exist, it will be created.

Value

a character object specifying the name of the file created. In general, this will be "mloci.out.gz".

Side Effects

If a file named "mloci.out" or "mloci.out.gz" already exist in the current directory, `sw2mloci` will move them to "mloci.out.before" or "mloci.out.before.gz" respectively before doing any calculations. `sw2mloci` also copies the IBD files and map file (if it is specified) to a temp space. This is done to bypass any write permission issues. This temp space is deleted when the function is finished. It also creates a temp space to hold the intermediate mibd files. These also will be deleted at the end of the function.

See Also

There are similar functions to deal with SOLAR mibds, see `phi2share`, `solar2mloci`, and `solar2multic`.

Examples

```
sw2mloci("../otherInput/sw18", "../otherInput/sw18/c18.map")

sw2mloci("sw18")

sw2mloci(".", "sw18/c18.map", output.directory = "multicInput")
```

`addGE` *Assess combinations of univariate multic objects*

Description

Determine whether there is any evidence that running a multivariate multic model will significantly increase the evidence of a genetic effect.

Usage

```
addGE(multic.objs, combine=2, plotit=FALSE, ibd.dist, statistic=c("lrt",
"wald"), legend=TRUE, ...)
```

Arguments

`multic.objs`	A list of 1-trait multic objects.
`combine`	Indicate how many traits should be examined together. The program will then look at all N traits choose 'combine'.
`plotit`	Logical, default=FALSE. If TRUE, a LOD plot is generated with a separate line for each combination of traits.
`ibd.dist`	The default is to use the distances from the first multic object. This options allows the user to provide a different set of distances.
`statistic`	Character, default="lrt". This determines whether the Wald statistic (MG/SE) or the LRT is used when combining the traits.
`legend`	Logical, default=TRUE. If TRUE and if plotit=TRUE then a legend is automatically provided.
`...`	Allows for graphical parameters to be passed to the plot function (only applicable when plotit=TRUE).

Value

A data frame is returned if the function is assigned to new object. Included are the various combinations (listed in order 1-N), the Chi-square statistic, the p-value, the distance, and the LOD score.

Side Effects

If plotit=T, a plot is generated on the current graphical device.

References

M. de Andrade, C. Olswold, J.P. Slusser, L.A.Tordsen, E.J. Atkinson, K.G. Rabe, and S.L.Slager. Identification of genes involved in alcohol consumption and cigarette smoking. BMC Genetics, 6:S112, 2005.

See Also

multic, gene.eff

Examples

```
add2 <- addGE(list(bmi10, dia10, sys10), combine = 2, plotit = T, ylim=c(0,8), legend=F)
add3 <- addGE(list(bmi10, dia10, sys10), combine = 3, plotit = F)
lines(add3$cM, add3$lod, col=4, lwd=2, lty=4)
key(corner=c(0,1), lines=list(lwd=2, col=1:4, lty=1:4),
    text=list(c('BMI-Dia','BMI-Sys','Dia-Sys','BMI-Dia-Sys'), col=1:4))
```

`expand.multic` *Create "bootstrapped" versions of mloci.out and share.out for multic*

Description

expand.multic is a utility function to create "bootstrap"ed versions of mloci.out and share.out

Usage

```
expand.multic(famids, mloci.out=NULL, share.out=NULL)
```

Arguments

`famids`	famids is a character or integer vector that specifies the family order in a "bootstrapped" fashon. Each index of famids is the famid (family identifier) from the original dataset not the index of the family. An example famids argument would be `famids <- sample(famid, length(unique(famid)), replace = TRUE)`. IMPORTANT NOTE: This sequence of famids must be the same as that passed to expand.data. If they are not, the dataset and the external data will not match.
`mloci.out`	a character value specifying the name of an mloci.out file. This file needs to have the famid portion (i.e., the characters before the hyphen [-]) of the unique id for each entry.
`share.out`	a character value specifying the name of an share.out file. This file needs to have the famid portion (i.e., the characters before the hyphen [-]) of the unique id for each entry.

Value

a list of two elements. The first is the name of the new mloci.out file. The second element is the name of the new share.out. Either element may be `NULL` if the respective input was `NULL`.

Side Effects

the output files are created in the current directory. If either of the input files (mloci.out or share.out) were gzip'ed, expand.multic will gunzip them. Currently, this is done in their own directory. However, in the future, this can be done in a temporary. Also, a directory named "loci" is temporarily craeted to hold split mloci.out file.

See Also

`expand.data`

Examples

```
famids <- sample(famid, length(unique(famid)), replace = TRUE)
new.files <- expand.multic(famids, "input/mloci.out", "input/share.out")
mult.obj <- multic( -- your formula, data, famid, etc. here --
                   mloci.out = new.files$new.mloci.out,
                   share.out = new.files$new.share.out)
```

`expand.data` *Create a "bootstrapped" version of a dataset to be used in multic.*

Description

When using multic to bootstrap over families, an appropriate data set is needed. By providing a random set of famids, expand.data creates such a dataset.

Usage

```
expand.data(famids, d.frame)
```

Arguments

`famids`	famids is a character or integer vector that specifies the family order in a "bootstrapped" fashon. Each index of famids is the famid (family identifier) from the original dataset not the index of the family. An example famids argument would be `famids <- sample(famid, length(unique(famid)), replace = TRUE)`. IMPORTANT NOTE: This sequence of famids must be the same as that passed to expand.multic. If they are not, the dataset and the external data will not match.
`d.frame`	the data.frame that holds the family structure and phenotype data. This should be the dataset that was used to sample famid.

Value

a data.frame that contains the bootstrapped version of the input dataset

See Also

`expand.multic`

Examples

```
famids <- sample(famid, length(unique(famid)), replace = TRUE)
expaned.ped.phen <- expand.data(famids, ped.phen)
```

12 Acknowledgements

We would like to acknowledge the GAW grant, GM31575, for use of the Simulated data from GAW 13.

References

[1] D. B. Allison, M. C. Neale, R. Zannolli, N. J. Schork, C. I. Amos, and J. Blangero. Testing the robustness of the likelihood ratio test in a variance-component quantitative trait loci (qtl) mapping procedure. *Am J Human Genetics*, 65:531–545, 1999.

[2] D. B. Allison, B. Thiel, P. Jean, R. C. Elston, M. C. Infante, and N. J. Schork. Multiple phenotype modeling in gene-mapping studies of quantitative traits: power advantages. *Am J Human Genetics*, 63:1190–1201, 1998.

[3] L. Almasy, , T. D. Dyer, and J. Blangero. Bivariate quantitative trait linkage analysis: Pleiotropy versus co-incident linkages. *Genetic Epidemiology*, 14:953–958, 1997.

[4] L. Almasy and J. Blangero. Multipoint quantitative-trait linkage analysis in general pedigrees. *An J Human Genetics*, 62:1198–1211, 1998.

[5] C. I. Amos. *Robust methods for detection of genetic linkage for data from extended families and pedigrees*. PhD thesis, Louisiana State University, 1988.

[6] C. I. Amos. Robust variance-components approach for assessing genetic linkage in pedigrees. *Am J Human Genetics*, 54:535–543, 1994.

[7] C. I. Amos, M. de Andrade, and D. Zhu. Comparison of multivariate tests for genetic linkage. *Human Heredity*, 51:133–144, 2001.

[8] C. I. Amos, R. C. Elston, G. E. Bonney, B. J. B. Keats, and G. S. Berenson. A multivariate method for detecting genetic linkage with application to the study of a pedigree with an adverse lipoprotein phenotype. *Am J Human Genetics*, 47:247–254, 1990.

[9] C. I. Amos, D. Zhu, and E. Boerwinkle. Assessing genetic linkage and association with robust components of variance approaches. *Annals of Human Genetics*, 60:143–160, 1996.

[10] B. Basrak, C. A. J. Klaassen, M. Beekman, N. G. Martin, and D. I. Boomsma. Copulas in qtl mapping. *Behavior Genetics*, 34:161–172, 2004.

[11] W. C. Blackwelder and R. C. Elston. A comparison of sib-pair linkage tests for disease susceptibility loci. *Genetic Epidemiology*, 2:85–97, 1985.

[12] J. Blangero and L. Almasy. Solar: sequential oligogenic linkage analysis routines. Population Genetics Laboratory 6, Southwest Foundation for Biomedical Research, San Antonio, TX 78228, 1996.

[13] J. Blangero and L. Almasy. Multipoint oligogenic linkage analysis of quantitative traits. *Genetic Epidemiology*, 14:959–964, 1997.

[14] J. Blangero, J. T. Williams, and L. Almasy. Variance components methods for detecting complex trait loci. *Advances in Genetics*, 42:151–181, 2001.

[15] D. I. Boomsma and C. V. Dolan. A comparison of power to detect a qtl in si b-pair data using multivariate phenotypes, mean phenotypes, and factor scores. *Behavior Genetics*, 28:329–340, 1998.

[16] M. de Andrade, C. Amos, and T. J. Thiel. Methods to estimate genetic components of variance for quantitative traits in family studies. *Genetic Epidemiology*, 17:64–76, 1999.

[17] M. de Andrade, B. Fridley, E. Boerwinkle, and S. T. Turner. Diagnostic tools in linkage analysis for quantitative traits. *Genetic Epidemiology*, 39:1–38, 2003.

[18] M. de Andrade, R. Gueguen, S. Visvikis, C. Sass, G. Siest, and C. Amos. Extension of variance components approach to incorporate temporal trends and longitudinal pedigree data analysis. *Genetic Epidemiology*, 22:221–232, 2000.

[19] M. de Andrade, J. Krushkal, L. Yu, and C. Amos. Act - a computer package for analysis of complex traits. *Am J Human Genetics*, 63:A287, 1998.

[20] M. de Andrade, C. Olswold, J.P. Slusser, L.A.Tordsen, E.J. Atkinson, K.G. Rabe, and S.L.Slager. Identification of genes involved in alcohol consumption and cigarette smoking. *BMC Genetics*, 6:S112, 2005.

[21] M. de Andrade, T. J. Thiel, L. Yu, and C. I. Amos. Assessing linkage in crhomosome 5 using components of variance approach: univariate versus multivariate. *Genetic Epidemiology*, 14:773–778, 1997.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *jrssb*, 39:1–38, 1977.

[23] P. J. Diggle, K. Y. Liang, and S. L. Zeger. *Analysis of longitudinal data*. Clarendon Press, Oxford, 1995.

[24] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, New York, 1993.

[25] R. C. Elston, S. Buxbaum, K. B. Jacobs, and J. M. Olson. Haseman and elston revisited. *Genetic Epidemiology*, 19:1–17, 2000.

[26] R. A. Fisher. The correlation between relatives on the supposition of mendelian inheritance. *Transactions of the Royal Society - Edinburgh*, 52:399–433, 1918.

[27] J. K. Haseman and R. C. Elston. The investigation of linkage between a quantitative trait and a marker locus. *Behavior Genetics*, 2:3–19, 1972.

[28] J. L. Hopper and J. D. Mathews. Extensions to multivariate normal models for pedigree analysis. *Annals of Human Genetics*, 46:373–383, 1982.

[29] P. J. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Fifth Berkeley Symp on Math Stat Probab*, volume 1, pages 221–233, 1967.

[30] S. J. Iturria and J. Blangero. An em algorithm for obtaining maximum likelihood estimates in the multi-phenotype variance components linkage model. *Annals of Human Genetics*, 64:349–369, 2000.

[31] R. E. Jennrich and M. D. Schluchter. Unbalanced repeated-measures models with structured covariance matrices. *Biometrics*, 42:802–802, 1986.

[32] N. M. Laird. Computation of variance components using the em algorithm. *J Stat Comp Sim*, 14:295–303, 1982.

[33] E. L. Lehman. *Nonparametrics: Statistical Methods Based On Ranks*. Holden-Day, San Fransisco, CA, 1975.

[34] A. L. Louis. Finding the observed information matrix when using the em algorithm. *jrssb*, 44:226–233, 1982.

[35] X.-L. Meng and D. B. Robin. Using em to obtain asymptotic variance-covariance matrices: The sem algorithm. *jasa*, 86:899–909, 1991.

[36] M. C. Neale and L. R. Cardon. *Methodology for genetic studies of twins and families*. Kluwer Academic Publishers, Dordrecht, 1993.

[37] M. A. Province and D. C. Rao. A new model for the resolution of cultural and biological inheritance in the presence of temporaltrends: application to systolic blood pressure. *Genetic Epidemiology*, 2:363–374, 1985.

[38] M. A. Province and D. C. Rao. Familial aggregation in the presence of temporal trends. *Statistics in Medicine*, 7:185–198, 1988.

[39] D. C. Rao, N. E. Morton, and S. Yee. Analysis of family resemblance. ii. a linear model for familial correlation. *American Journal of Human Genetics*, 26:331–359, 1974.

[40] S. Schmitz, S. S. Cherny, and D. W. Fulker. Increase in power through multivariate analyses. *Behavior Genetics*, 28:357–363, 1998.

[41] S. R. Searle. *Matrix Algebra Useful for Statistics*. Wiley, New York, 1982.

[42] S. R. Searle, G. Gasella, and C. E. McCulloch. *Variance Components*. Wiley & Sons, New York, 1992.

[43] S. G. Self and K. Y. Liang. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard condtitions. *jasa*, 82:605–610, 1987.

[44] P. C. Sham and S. Purcell. Equivalence between haseman-elston and variance-components linkage analyses for sib pairs. *American Journal of Human Genetics*, 68:1527–1532, 2001.

[45] E. Sobel and K. Lange. Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics. *Am J Human Genetic*, 58:1323–1337, 1996.

[46] A. J. M. Sorant and R. C. Elson. *A subroutine package for function maximization (a user's guide to MAXFUN version 6.0): S.A.G.E. documentation*, 1994.

[47] E. A. Thompson and R. G. Shaw. Pedigree analysis for quantitative traits: Variance components without matrix inversion. *Biometrics*, 46:399–413, 1990.

[48] S. T. Turner, T. R. Rebbeck, and C. F. Sing. Sodium-lithium countertransport and probability of hypertension in caucasians 47 to 89 years old. *J Hypertension*, 20(6):841–850, 1992.

[49] S. T. Turner and C. F. Sing. Erythorocyte sodium transport and the probability of having hypertension. *J Hypertension*, 14(7):829–837, 1996.

[50] S. T. Turner, W. H. Weidman, V. V. Michels, T. J. reed, C. L. Ormson, T. Fuller, and C. F. Sing. Distribution of sodium-lithium countertransport and blood pressure in caucasians five to eighty-nine years of age. *Hypertension*, 13:378–391, 1989.

[51] J. T. Williams, P. Van Eerdewegh, L. Almasy, and J. Blangero. Joint multipoint linkage analysis of multivariate qualitative and quantitative traits. i. likelihood formulation and simulation results. *Am J of Human Genetics*, 65:1134–1147, 1999.