Statistical Analysis for Genetic Epidemiology (S.A.G.E.) Version 5.1.1 User Reference Manual

Department of Epidemiology and Biostatistics
Case Western Reserve University
Cleveland, Ohio

December 12, 2005

Limited Warranty

Case Western Reserve University warrants that the media by which the Software is distributed will be free from defects for a period of sixty (60) days from the date of delivery of the Software to you. Your sole remedy in the event of a breach of this warranty will be that Case Western Reserve University will, at its option, replace any defective software returned to Case Western Reserve University within the warranty period or refund the money you paid for the software. Case Western Reserve University does not warrant that the software will meet your requirements or that operation of the software will be uninterrupted or that the software will be error-free.

THE ABOVE WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESSED OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM COUNTRY TO COUNTRY AND STATE TO STATE.

Disclaimer of Damages

REGARDLESS OF WHETHER ANY REMEDY SET FOR THE HEREIN FAILS OF ITS ESSENTIAL PURPOSE, IN NO EVENT WILL CASE WESTERN RESERVE UNIVERSITY BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT OR SIMILAR DAMAGES, INCLUDING ANY LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF CASE WESTERN RESERVE UNIVERSITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

IN NO CASE SHALL THE LIABILITY OF CASE WESTERN RESERVE UNIVERSITY EXCEED THE PURCHASE PRICE FOR THE SOFTWARE. NO RESPONSIBILITY IS ASSUMED BY THE AUTHORS OR ANY OTHERS WHO CONTRIBUTED TO **S.A.G.E.** IF ERRORS OR PROBLEMS ARE FOUND, PLEASE CONTACT THE ADDRESS BELOW.

S.A.G.E. v5.1.1

S.A.G.E. Contact Information:

S.A.G.E.

Division of Molecular and Genetic Epidemiology

Department of Epidemiology and Biostatistics

Wolstein Research Building

2103 Cornell Rd

Case Western Reserve University

Cleveland, Ohio 44106-7281

Email: sage@darwin.cwru.edu

http://darwin.cwru.edu

Tech Support: sage@darwin.cwru.edu

S.A.G.E. is freely available from our web site: **http://darwin.epbi.cwru.edu/sage/**

NOTICE

The recommended way of referencing the current release of the S.A.G.E. programs is as follows:

S.A.G.E. 5.x [2005]. Statistical Analysis for Genetic Epidemiology http://darwin.cwru.edu/sage/. All users of S.A.G.E. Software are under obligation to:

1. Make sure that every publication that presents results from using S.A.G.E. carries an appropriate acknowledgment such as: "(Some of) The results of this paper were obtained by using the program package S.A.G.E., which is supported by a U.S. Public Health Service Resource Grant (RR03655) from the National Center for Research Resources" (it is important that the grant number appears under "acknowledgments").
2. Send bibliographic information about every paper in which S.A.G.E. is used (author(s), title, journal, volume and page numbers; a reprint will do provided it has the necessary information on it) to:

R.C. Elston

Division of Molecular and Genetic Epidemiology

Wolstein Research Building

2103 Cornell Rd

Case Western Reserve University

Cleveland, Ohio 44106-7281

USA

Contents

Chapter 1

Introduction

Statistical Analysis for Genetic Epidemiology (S.A.G.E.) is a collection of compiled C++ programs that perform a wide variety of genetic analyses. The range of functionality includes tools for

- extracting summary statistics describing the data and evaluating general data quality,
- estimating allele frequencies,
- estimating heritability and familial correlations,
- inferring mixture models for genetic transmission, and penetrance functions, including variable age of onset,
- estimating identity-by-descent (IBD) allele sharing probabilities between relative pairs,
- performing model-based linkage analysis,
- performing model-free linkage analysis,
- performing transmission/disequilibrium (TDT) analysis, and
- analyzing trait/allele associations.

S.A.G.E. runs on a variety of platforms, including Linux, Windows, Solaris, and Tru64. The programs may be run from a command line from a cross-platform graphical user interface (GUI) that is included as part of the complete package. The software is extremely flexible with respect to the structure of input pedigree data files and, unless otherwise stated, the dependent phenotypes and traits may be discrete (including dichotomous data) or continuous.

Please check our web page for the most up-to-date information on the S.A.G.E. 5.x programs at the following URL:

http://darwin.case.edu/

1.1 Program Descriptions

1.1.1 Summary Statistics

PEDINFO

PED*igree* ***INFO****rmation and statistics*: Provides many useful descriptive statistics on pedigree data including means, variances and histograms of family, sibship and pedigree sizes, and counts of each type of relative pair.

1.1.2 Data Quality

MARKERINFO

MARKER INFOrmation: Detects Mendelian inconsistencies of markers in pedigree data.

RELTEST

REL*ationship* ***TEST****ing*: Indicates pairs of relatives to be reclassified according to their true relationship using multi-point genome scan data. The method is based on a Markov process model of identity-by-descent (IBD) allele-sharing along chromosomes. This program currently analyzes four different types of putative pairs: full sib pairs, half sib pairs, parent offspring pairs and unrelated marital pairs. A summary file is produced that contains the pairs to be reclassified together with their Mean Allele-Sharing Statistic, Parent Offspring Statistic and, for each individual, the percentage of marker data that is missing.

1.1.3 Allele Frequency Estimation

FREQ

Allele ***FREQ****uency estimator*: Estimates allele frequencies from marker data on related individuals with known pedigree structure and generates marker locus description files, needed by GENIBD, MLOD, and other S.A.G.E. programs. Future versions will also have the ability to estimate genotype and haplotype frequencies in the presence of allelic disequilibrium.

1.1.4 Familial Aggregation

ASSOC

Marker-Trait ***ASSOC****iations in Pedigree Data*: Simultaneously analyzes from pedigree data the association between a trait and covariates, which can include marker phenotypes that have been transformed into quantitative covariates, and residual familial correlations/heritability.

FCOR

***F**amily **COR**relations*: Calculates multivariate familial correlations with their asymptotic standard errors. Calculates familial correlations for all pair types available in the pedigrees without assuming multivariate normality of the traits across family members.

1.1.5 Commingling Analysis

SEGREG

SEGREG*ation models*: This program can be used to fit mixtures of two or three normal distributions, simultaneously applying a power transformation to the data and also allowing for both ascertainment and residual familial correlations.

1.1.6 Segregation Analysis

SEGREG

SEGREG*ation models*: Fits and tests Mendelian segregation models in the presence of residual familial correlations. The trait analyzed can be continuous, binary, or a binary disease trait with variable age of onset. This program can also be used for commingling analysis, to predict the major genotype of any pedigree member, and to prepare penetrance files for model-based linkage analysis.

AGEON

***AGE** of **ON**set*: Produces maximum likelihood estimates of the parameters of a mixed power-normal distribution for a binary trait with variable age of onset. The mean, variance and susceptibility parameters can be specified as dependent on covariates.

1.1.7 IBD Allele Sharing Analysis

GENIBD

***GEN**erate **IBD** sharing probabilities*: Generates both single- and multi-point identity-by-descent (IBD) distributions using a variety of algorithms tuned for different types of relative pairs in pedigrees. Exact methods can be used for small pedigrees with loops, and simulation methods are available for large extended pedigrees with loops. In the case of small pedigrees IBD sharing can also be interpolated between markers.

1.1.8 Model-Based Linkage Analysis

LODLINK

*Single-point model-based **LOD** score **LINK**age analysis*: LOD scores and recombination fractions are obtained between a marker and trait that follows any Mendelian model allowed by SEGREG (which can be used to generate the appropriate penetrance files). Test of linkage heterogeneity, and of linkage in the presence of linkage heterogeneity, are included.

MLOD

M*ulti-point model-based* **LOD** *score analysis*: Performs multi-point model-based LOD-score linkage analysis on small pedigrees. Analysis is greatly optimized for examining multiple one-locus trait models and will, in future versions, allow for meiosis specific (e.g., age and sex specific) recombination fractions.

1.1.9 Model-Free Linkage Analysis

LODPAL

***LOD** score **P**air **A**na**L**ysis*: Performs analysis based on the LOD score formulation for affected-sib-pairs (ASP). The current implementation is of the general conditional logistic model, including the one-parameter model that allows for the inclusion of all affected-relative-pairs, covariates and epistatic interactions.

SIBPAL

***SIB**ling **P**air **A**na**L**ysis*: Performs mean tests, proportion tests and linear regression-based modeling of squared sib-pair differences and mean-corrected sums of a trait as a function of marker allele identity-by-descent sharing. Available analyses can use either single- or multi-point IBD information, and models allow for both binary and continuous traits due to multiple genetic loci, including epistatic interactions and covariate effects.

1.1.10 Transmission Disequilibrium

TDTEX

***T**ransmission **D**isequilibrium **T**est (**EX**act)*: This program implements several asymptotic and exact versions of the transmission disequilibrium test (TDT) for testing linkage between marker and disease loci in the presence of allelic association. The exact tests are useful in cases where little data are available or there are many alleles at the marker locus. Different types of tests are available, including an exact test and a Markov chain Monte Carlo randomization test, as well as several exact marginal homogeneity tests.

1.1.11 Allelic Association

ASSOC

*Marker-Trait **ASSOC**iations in Pedigree Data*: Analyzes the association between a continuous trait and covariates, which can include marker phenotypes that have been transformed into quantitative covariates, from pedigree data in the presence of familial correlations. Together with the *Transmitted Allele Indicator* (available as a user-defined function), performs a pedigree transmission disequilibrium test (TDT).

1.1.12 Haplotype Analysis

DECIPHER

Obtains maximum likelihood estimates of population haplotype frequencies for autosomal or X-linked markers, and determines all possible diplotypes and the most likely diplotypes for each individual. Estimates haplotype frequencies for different populations as specified by the user. Performs likelihood ratio tests and permutation tests to compare haplotype frequency distributions for dichotomous phenotypes.

1.2 Program Limitations

All programs will cease to function when the user's current license expires.

All programs currently make the following assumptions in all of their analysis methods:

- each genetic marker has a known genotype-phenotype relation (which may be either deterministic or probabilistic),
- the founders of each constituent pedigree[1] are not inbred and are unrelated to one another, and in particular, the pedigrees do not comprise loops,
- the members of each constituent pedigree are unrelated to the members of any other constituent pedigree, and
- there are no selection, migration or mutation effects[2].

[1] The user defines pedigrees by giving each pedigree a unique identification number. A subset of pedigree members for whom there is no information on how they are related to other members of that pedigree is called a *constituent pedigree* and is treated as an independent pedigree in all analyses. With the exception of the program PEDINFO, the term *pedigree* will always refer to a *constituent pedigree* as defined here.

[2] Several programs allow for non-Hardy-Weinberg equilibrium proportions and the SEGREG program allows for general transmission models.

1.3 Conventions Used in this Manual

This document uses the following typographical conventions to help clarify the correct specification of S.A.G.E. program commands and options:

1. All references to `parameters` and `attributes` are printed using a `non-proportional` font.
2. All references to *named constant* values (e.g., **true** and **false**) outside of a syntax table (see 2.2.4) are printed using a **bold** font.
3. Examples of parameter files are printed using a `non-proportional` font. Specific values within a given example are printed using a **`bold non-proportional`** font.
4. Examples of program outputs are printed using a `non-proportional` font.
5. Technical terms that have not been previously introduced in the manual are printed using an *italics* font. The term's definition will be explicitly given if its meaning is not evident from the context.
6. Text that needs to be otherwise EMPHASIZED is printed using an UPPER CASE font.

Chapter 2

Program Input and Output

Each S.A.G.E. program requires several input files in order to run. No program requires all of the possible input files. Refer to the individual program documentation for specific information on which files are required. The file types currently used for program input are:

Section	File Type	Description
2.2	Parameter file	Specifies the parameters and options with which to perform a particular analysis.
2.3.2	Pedigree data file	Contains delimited records for each individual, including fields for identifiers, sex, parents, trait and marker data.
2.5	Marker locus description file	Lists the alleles, allele frequencies and phenotype to genotype mapping for each marker locus.
2.5	Trait locus description file	Lists the genetic model for each of the traits to be analyzed for linkage using a specific genetic model
2.6	Genome description file	Contains a description of the linked marker regions, including distances between markers. This file is not required for single-point[a] analysis.
2.7	IBD sharing file	Stores identity-by-descent (IBD) distributions between pairs of related individuals at one or more marker loci.
	LODPAL pairs file	Stores the pre-constructed pair-specific covariate and/or weight values to be used in the analysis

[a] Single-point in the sense that information is used from only one observed marker locus at a time. When performing linkage analysis this is often called "two-point" analysis.

Each program also produces one or more output files that contain results and diagnostic information. Refer to the individual program documentation for specific information on which files are produced and details of what they contain.

The file types currently used for program output are:

Section	File Type	Description
2.8	Information output file	Contains informational diagnostic messages, warnings and program errors. Each program generates one information output file. No analysis results are stored in this file. Information files are automatically named with a "INF" extension (for example, "segreg_analysis.inf").
2.9	Analysis output file	Each program may generate one or more analysis output files. These files contain the results of each analysis or may summarize the results of many analyses.

2.1 Running S.A.G.E. Programs

S.A.G.E. programs may be executed from the provided Graphical User Interface (GUI) or, alternatively, by means of a command line directive that specifies the name of a selected program followed by a list of *arguments* to the program. The order of the arguments is important. If the user enters an incorrect or incomplete command line, then the software will display a usage statement that indicates the correct syntax for the desired program[1]. For example, if the PEDINFO program name were entered without any arguments, the result would be as follows:

>**pedinfo**

```
S.A.G.E. v4.x -- PEDINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: pedinfo <parameters> <pedigree>
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
```

As indicated in this program usage statement, two input files need to be listed on the command line.
A typical run of PEDINFO may look like the following, if run from the "`example/output/pedinfo`" directory:

```
>cd example/output/pedinfo
>pedinfo ../input/parameters ../input/pedigree
S.A.G.E. v4.x -- PEDINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading parameter file.....................done.
Reading pedigree file.....................
from ../../input/pedigree1..done.
```

[1]The GUI is designed to generate syntactically correct lists of program arguments based on the user selections in the various screens and dialogs. The argument lists are in turn forwarded to the desired S.A.G.E. program(s) for processing. Although the GUI does perform validation of user selections before submitting them for processing, it is nevertheless possible that the user can generate an invalid S.A.G.E. command through the GUI. The user is therefore advised to check the S.A.G.E. outputs to the console as well as the information file (*.INF) when the program does not seem to work as expected.

```
Sorting pedigrees.........................done.
Generating statistics.....................done.
Analysis complete!
```

2.2 The Parameter File

User options for analysis are specified to S.A.G.E. programs as a list of instructions within a *parameter file*[2]. When a particular S.A.G.E. program is executed it evaluates the contents of the specified parameter file to determine

1. how to interpret the contents of the given pedigree data file,
2. how many different analyses have been requested and
3. which options have been specified for each analysis.

A parameter file is simply a text file containing a list of S.A.G.E. program instructions written according to a specific syntax (see below). A single parameter file may be used to specify options for one or more S.A.G.E. programs in any combination. In other words, one parameter file could specify analysis options for several different S.A.G.E. programs, or different options for repeated calls to the same program, or both. And, of course, the user always has the option of creating a set of different parameter files if that makes it easier to manage a given project [3]. Since the parameter file also contains user-supplied specifications on how to interpret the pedigree data file, the ability to specify an arbitrary set of S.A.G.E. analyses within a single parameter file makes the software very flexible.

2.2.1 Creating a Parameter File

One of the primary functions of the GUI is to create parameter files for S.A.G.E. programs. The GUI is designed to translate the user's selection on the various screens and dialogs into syntactically correct lists of program arguments, which are automatically passed into the appropriate S.A.G.E. program. This feature is intended to reduce the complexity associated with learning the syntax of S.A.G.E. parameter files, and is expected to be particularly beneficial to novice users of the software. Experienced users who prefer to create and edit their parameter files directly will continue to have the option of doing so.

A parameter file may be created and modified using a standard text editor on the local S.A.G.E. platform (i.e., the system on which S.A.G.E. has been installed), or the file may be produced on a different system and copied to the local S.A.G.E. platform. The user will normally want to copy the parameter file into the same directory that contains the pedigree data file for a given project, although this is not required.

[2]The reader is cautioned that the word *parameter* will have two meanings in this document. In one context it will refer to the set of defined S.A.G.E. *keywords*, but in a statistical context it will refer to some distribution characteristic (e.g., the mean (μ) or variance (σ^2) of a normal distribution). One goal of the typographical conventions (see 1.3) is to make the context of this word as clear as possible.

[3]S.A.G.E. programs accept only one parameter file at a time (the one named as a program argument), regardless of the number available.

In computing environments that include both Unix workstations and Windows PCs, many individuals find the text editors available in Windows to be more user-friendly and convenient, and therefore would prefer to edit their parameter files with either Notepad or WordPad. Users who take this approach must remember to remove the spurious *carriage return* character (^M) which appears at the end of each line of the text file after it has been copied to the Unix target directory[4].

[4]Utility programs available under Unix, such as *dos2unix* make this task fairly easy.

2.2.2 Parameter File Syntax and Structure

A parameter file consists of a list of S.A.G.E. program instructions known as *statements*. When a particular parameter file is passed to a S.A.G.E. program (as a command line argument), the specified program reads each line in the parameter file, from top to bottom, and configures itself to perform the analyses indicated by the listed statements.

All S.A.G.E. statements are formed according to the following format:

`parameter`[= **value**] [, `attribute` [= **value**]]*[{[*statement*]*}]

in which the square brackets ([ ]) indicate groupings of optional terms and are not to be entered by the user. The asterisk (*) indicates that the preceding group or item may be repeated zero or more times. Note that the brackets ([ ]) and asterisk (*) are artifacts of the above format definition, and are not to be entered by the user.

In words, the above format definition says:

> "A *statement* is a `parameter` followed by an optional equal-sign-and-value pair[5], followed by zero or more optional comma-and-`attribute` pairs (in which each `attribute` may be followed by an optional equal-sign-and-value pair). This totality, in turn, is optionally followed by a brace-enclosed list of zero or more *statements*."

The terms `parameter` and `attribute` represent S.A.G.E. *keywords* specified throughout this document, and the braces ({ }) are used to enclose an optional *block* of zero or more subsequent statements[6]. Further, the < and > symbols may be used instead if there are no { and } symbols on the user's keyboard.

The recursive manner by which a statement is defined in terms of itself is no accident and is, in fact, a common way to specify a formal language structure.

[5] `Parameters` and `attributes` often do not require an explicit value to be assigned, allowing the user to run the selected S.A.G.E. program with its default values.

[6] When a brace-enclosed block is nested within another, enclosing block, the nested blocks are referred to as *sub-blocks*.

An illustration of the overall structure of a parameter file is as follows:

```
parameter = value, attribute = value
parameter = value, attribute = value

parameter, attribute = value, attribute = value, attribute = value
parameter = value, attribute = value

parameter = value {
   parameter = value
   parameter, attribute = value
   parameter {
      parameter = value
      parameter = value
      parameter = value
   }
   parameter {
      parameter = value
      parameter = value
      parameter = value
      parameter = value, attribute = value
      parameter, attribute = value, attribute = value, attribute = value

      parameter = value, attribute = value
   }
   parameter = value, attribute = value
   parameter, attribute = value, attribute = value, attribute = value
   parameter = value, attribute = value
}
```

The S.A.G.E. statement grammar described above is complex, which can make the software difficult to learn. As noted previously the S.A.G.E. GUI is designed to eliminate the burden of learning the parameter file syntax; however, there may be times when an experienced user would prefer to manipulate the parameter files directly. The following section provides clarifying details and examples of parameter file syntax.

2.2.2.1 Syntax Details

1. The specific `parameters` and `attributes` listed within this document are S.A.G.E. *reserved words*, meaning that they must be spelled exactly as shown in their corresponding syntax table (see 2.2.4).

2. The names of phenotypes, traits and covariates found in the pedigree data file may be the same as the names of `parameters` and `attributes`, although this practice is likely to cause confusion and is therefore not recommended.

3. White space, including blanks, tabs and newline characters, are required only to differentiate between successive `parameters`, `attributes` and values, and are otherwise ignored[7]

[7] White space that occurs as part of a QUOTED character string (e.g., “Body Mass Index”) is not ignored.

by S.A.G.E. programs. They may usually be inserted or omitted from statements at the user's discretion.[8]

4. Blank lines between successive statements are ignored and may be inserted as necessary to make the parameter file easier to read.

5. Due to the recursive nature of their definition, statements can be *nested* when listed in the parameter file. That is, a particular statement may be specified as containing another *sub-statement* which in turn contains one or more *sub-sub-statements*, etc. This manual refers to such sub-statements as *sub-blocks*.

6. A single statement may fit on a single line or may continue over several lines. Further, the placement of braces (for enclosed blocks and sub-blocks) is left entirely to the discretion of the user. The following example shows two ways to specify the same `segreg` statement:

```
segreg,out="my_analysis.out"{trait=BMI type_mean{option=three}}
```

```
segreg, out = "my_analysis.out"
{
   trait = BMI
   type_mean
   {
      option = three
   }
}
```

2.2.2.2 Adding Comments to the Parameter File

The insertion of a pound sign (#) at any point of a line in a parameter file causes S.A.G.E. to ignore the remainder of that line. Thus, the user can *comment* on the contents of a parameter file that may need to be reviewed at some future time. The following example shows how the above-listed `segreg` block might be commented:

```
# Perform analysis on Body Mass Index
segreg, out = "my_analysis.out"
{
   trait = BMI
   type_mean
   {
      option = three # Run the 3-mean model
   }
}
```

[8] Many users find that judicious use of blank spaces can make a parameter file easier to read, and therefore less prone to error.

Here is a more elaborate approach to commenting a S.A.G.E. parameter file:

```
# ------------------------------------------------------------------------------
# Begin File: Example_01.par
# ------------------------------------------------------------------------------
# ****************************************************************************** #
# ***************          S.A.G.E. Parameter File            *************** #
# ***************              C. L. Dodgeson                 *************** #
# ***************                 9 Sep 03                    *************** #
# ****************************************************************************** #

# ****************************************************************************** #
# ***************               PEDIGREE BLOCK                *************** #
# ****************************************************************************** #
pedigree, character {
   # ---------------------------------------------------------------------------
   # General specifications
   # ---------------------------------------------------------------------------
   delimiters               = ","
   delimiter_mode           = single
   individual_missing_value = " "
   sex_code, male           = M
   sex_code, female         = F
   sex_code, missing        = " "
   verbose                  = 100
   require_record           = false
   whitespace               = "  \t"
   # ---------------------------------------------------------------------------
   # Field specifications
   # ---------------------------------------------------------------------------
   pedigree_id   = PID       , name = PedID
   individual_id = ID        , name = IndID
   parent_id     = P1        , name = Mom
   parent_id     = P2        , name = Dad
   sex_field     = SEX       , name = Sex
   phenotype     = DISEASE   , name = "Aff Stat,
      binary,
      affected   = 1,
      unaffected = 0
   phenotype     = HEMATOCRIT, name = Hematrocrit
}
# ****************************************************************************** #
# ***************                PEDINFO BLOCK                *************** #
# ****************************************************************************** #
pedinfo, out = “assgn 01 pedinfo altered out.txt” {
   phenotype     = "Aff Stat"
   phenotype     = Hematrocrit
   each_pedigree = false
}
# ------------------------------------------------------------------------------
# End File: Example_01.par
# ------------------------------------------------------------------------------
```

2.2.3 `Parameter` and `Attribute` Values

2.2.3.1 Character Strings

When a particular parameter or attribute takes a *character string*[9] for its value, the user should enter the desired *alphanumeric* character sequence[10] after the equal sign (=). Enclosing *double quotes*[11] are only required in the following cases:

- the string contains blank spaces (e.g., “Alice in Wonderland”),
- the string contains *non-alphanumeric* characters[12] (e.g., “Alice-in-Wonderland”)
- the string contains no characters at all i.e., it has length of zero (e.g., “”). A zero-length string is sometimes referred to as a *null* string.

S.A.G.E. is *case insensitive* with respect to the names of traits, phenotypes and covariates, and therefore, the following statements are equivalent:

```
trait = HT, type = continuous
trait = hT, type = continuous
trait = Ht, type = continuous
trait = ht, type = continuous
```

In all other cases, S.A.G.E. is *case sensitive*.

2.2.3.2 Numeric Values

When a particular parameter or attribute takes a numeric quantity for its value, the user is required to enter a constant according to the normal conventions of decimal notation. Specific constraints on the value are as follows:

[9] A character string is simply a contiguous sequence of zero or more letters, digits, or other typographic symbols, including spaces.

[10] An alphanumeric string may contain only letters (upper or lower case) and decimal digits.

[11] We distinguish between two kinds of quotation marks: double quotes (“ ”) and single quotes (’ ’). Unless otherwise stated, the double quotes should be used whenever the syntax rules call for a quoted string.

[12] Typographic symbols OTHER than letters or digits: ~!@#$%^&*()+‘-={}|[]\:”;’<>?,./

Numeric Value Constraint Notation	
Notational Form	**Meaning**
(-∞, ∞)	Any real number
(-∞, ∞) - {*a*}	Any real number except for *a*.
(*a*, ∞)	Any real number greater than *a*.
[*a*, ∞)	Any real number greater than or equal to *a*.
(-∞, *b*)	Any real number less than *b*.
(-∞, *b*]	Any real number less or equal to *b*.
(*a*, *b*)	Any real number greater than *a* and less than *b*.
[*a*, *b*)	Any real number greater than or equal to *a* and less than *b*.
(*a*, *b*]	Any real number greater than *a* and less than or equal to *b*.
[*a*, *b*]	Any real number greater than or equal to *a* and less than or equal to *b*.
{x_1, x_2, x_3, ..., x_n,}	Any one of a discrete list of given items.
{*i*, *i*+1, *i*+2, ..., *n*}	Any integer from *i* to *n*, inclusive.
{*i*, *i*+1, *i*+2, ...}	Any integer greater than or equal to *i*.
{..., *i*-2, *i*-1, *i*}	Any integer (positive or negative) less than or equal to *i*.

In addition to decimal quantities, S.A.G.E. also accepts the following *named constants*:

- **pi**, designating the transcendental number π = 3.141592654...
- **e**, designating the base of the natural logarithms = 2.718281828459...

The following example shows some ways in which numeric values may appear within S.A.G.E. statements:

```
segreg {
   composite_trait {
      covariate = BMI, val = 27.69
   }
   transmission {
      option = homog_general
      tau = A*, val = 0.5
   }
}
```

2.2.4 Reading and Interpreting the Syntax Tables

As mentioned previously, S.A.G.E. requires the user to specify options to its various programs by entering a list of statements into a parameter file, in a manner similar to common programming languages such as C++ and Python. When S.A.G.E. is executed, it reads the parameter file to determine

1. how to interpret the given pedigree data file(s),

2. which analyses (programs) are to be run, and
3. how to configure itself for the requested analyses.

For every `parameter` defined within the S.A.G.E. family of programs, this user document provides the following information in tabular form:

- `parameter` designation
- list of `attributes` associated with a given `parameter`
- a brief explanation
- range of valid or possible values that the `parameter` or `attribute` can take (see 2.2.3.2)
- the default value
- whether or not a value is required
- a list of applicable notes when more detailed explanation is required; these notes will always be found immediately at the end of the table.

To understand how to interpret the syntax tables used in this document, consider the following example:

parameter [, attribute]	**Explanation**

`pedigree_id`
`individual_id`
`parent_id`
`sex_field`[a]
	Declare respective pedigree field names for pedigree ID[b], individual ID, parental ID and sex designator.

	Value Range: Character string representing the valid name of a field[c] in the pedigree data file.
	Default Value: None[d]
	Required: Yes[e]
	Applicable Notes: 1, 2[f]
`sex_code`	Declare codes used to specify sex of individuals in the pedigree.
	Value Range: N/A[g]
	Default Value: None
	Required: No
	Applicable Notes: None
`, male`[h]	Specifies male sex code
	Value Range: Character string. Typical values are: 1, 0, M, m
	Default Value: M
	Required: No
	Applicable Notes: None
`, female`	Specifies female sex code.
	Value Range: Character string. Typical values are: 0, 1, F, f
	Default Value: F
	Required: No
	Applicable Notes: None

[a] The occurrence of multiple *parameter* names in a single cell means that the explanatory information at the right is applicable to all of them, and attributes listed within the cell are also applicable to all of them.

[b] An acronym for *identifier*.

[c] For users who are accustomed to spreadsheets, the database term *field* is analogous to *column*, and the term *record* is analogous to *row*.

[d] **None** means the default value is unspecified, inapplicable or both.

[e] If **Yes**, then the listed parameter or attribute is required, and the user must explicitly enter the listed parameter or attribute, optionally followed by an assignment expression (eg., `sex_code, male = "M"`), into the parameter file. If **No**, then the listed parameter or attribute is *not* required, and the specified default value will be used in the analysis. Note: When relying on default values for a given analysis, the user should take care to ensure that they are appropriate for the intended model.

[f] The applicable notes will be found immediately below the table.

[g] N/A means *not applicable*, i.e., that the `parameter` or `attribute` in question is *self-defining* and does not take any values.

[h] `Attributes` are indented with respect to their associated `parameters`, but appear in the same cell. Relevant explanatory information appears to their immediate right.

2.3 The Pedigree Data File

For family data to be accurately analyzed they must be described and represented precisely. The following are the definitions of various non-obvious family structures and relationships that are used throughout this manual[13].

Term	Definition
pedigree[a]	A set of individuals identified as belonging to the same pedigree, i.e., having the same pedigree ID[b]. These individuals may or may not be related in any way, but those who are NOT members of the same pedigree should NOT be related.
individual	A member of a pedigree.
founder[c]	An individual with at least one descendant who has neither parent in the pedigree. Founders are assumed to be unrelated by ancestry to any other founder.
non-founder	An individual descended from at least one founder.
mate relationship	Two individuals in a pedigree who have one or more offspring with each other are related by a mate relationship. Each individual may be a member of several mate relationships.
nuclear family	A set of two individuals who have a mate relationship and their natural children.
constituent pedigree[d]	A *complete* set of individuals in the same pedigree who are related by marriage, ancestry or descent and for whom there is enough information to indicate that they are so related. By complete is meant that all individuals in the pedigree who are so related must be included in the constituent pedigree.
singletons[e]	The set of individuals who have no relation to any other member of the pedigree they belong to.
marriage ring	A cycle of mate relationships in the undirected graph of individuals in a constituent pedigree.
non-marriage loop	A cycle containing at least one offspring and one mate relationship in the undirected graph of individuals in a constituent pedigree. (This includes consanguineous loops and non-consanguineous loops that involve both mate and offspring relationships).

[a] Other software packages refer to our definition of pedigrees as kindreds.

[b] "ID" is an acronym for *identifier*, and is used frequently throughout this document.

[c] Founders do not include singleton individuals.

[d] A constituent pedigree is what is typically referred to as a pedigree in the literature. The distinction is made because of the prevalence of incomplete and fragmented datasets.

[e] Singletons are sometimes not differentiated from founders in the literature.

2.3.1 Pedigree Data File Specification

A pedigree data file is a text file composed of one or more records, each of which contains information about a single individual in a pedigree. Each record must end with a carriage return or linefeed

[13] Some of these definitions are fairly technical but under most circumstances the conventional definitions will suffice.

character[14] and contains the following fields:

Pedigree Data Field Requirements			
Field	Value Type	Description	Required
Pedigree ID	character string or numeric	Uniquely identifies a particular pedigree within the file.	yes
Individual ID	character string or numeric	Uniquely identifies a particular individual within a pedigree[a].	yes
Parent ID	character string or numeric	Identifier of the individual's parent (either father or mother).	yes
Parent ID	character string or numeric	Identifier of the individual's other parent (either mother or father, depending on which was specified previously[b]).	yes
Sex	character string or numeric	Individual's sex[c].	yes
Continuous Traits, Phenotypes and Covariates	numeric	Observational trait and phenotype data with respect to the individual.	no
Discrete Traits, Phenotypes and Covariates	character string or numeric	Observational trait and phenotype data with respect to the individual.	no
Genotype Data	character string or numeric	Genotypic data with respect to the individual.	no[d]
Other Fields	character string or numeric	Other data related to the individual.	no

[a] Implicit in this is the possibility that the same Individual ID may appear more than once in a given pedigree data file, referring to a different individual at each occurrence.

[b] At the user's option, the pedigree data file may list the father's ID first, followed by the mother's ID.

[c] Incorrect use of the word *gender* is studiously avoided here. As the poet says, "Nouns have gender, whereas people have sex ... and enjoy it!"

[d] Genotypic data are, of course, required for programs that perform linkage analysis, allelic association analysis, etc.

There are two distinct types of record formats for pedigree data files: *character delimited* and *column delimited*. The two formats differ only in the method by which the pedigree fields are distinguished from one another within a given record, and are detailed in the following sections.

2.3.1.1 Character Delimited records

In a *character delimited* record the individual fields are separated by one or more characters, known as *delimiters*, which are usually not present in any of the data elements themselves. Commonly used delimiters are the comma, the tab, and the space, but any non-alphanumeric character may be used. If your data are separated by a fixed known delimiter, then S.A.G.E. will read your pedigree file as character delimited records, and you will need to specify which delimiter is used along with some additional *metadata*[15] that specify the exact order, names and types of the fields in your pedigree

[14] Any combination of carriage return and line feed characters is sufficient to terminate a record. This allows data files from most popular operating systems to be used without translation.

[15] Database terminology that means "information about the data", i.e., field names, data types, value ranges, etc.

records.

Files that are formatted for LINKAGE, GENEHUNTER, PAP, GAS or similar computer programs may all be read as character delimited records with little or no modification[16]. Programs that readily generate data in a character delimited form are spreadsheet programs like Microsoft Excel, most pedigree drawing programs, and most database programs.

Character delimited pedigree data file records can also be read as *column delimited* records provided that:

1. each record is exactly one line long,
2. there is at least one delimiter character between fields,
3. all fields contain at least one non-delimiter character, i.e., no field is empty, and
4. the corresponding fields in different records are the same length.

Pedigrees stored in character-delimited format contain exactly one line for each individual with specific delimiter characters separating data fields.

THE FORMAT OF THE CHARACTER DELIMITED PEDIGREE DATA FILE IS DEFINED BY A CHARACTER-DELIMITED LIST OF DISTINCT NAMES THAT IDENTIFY EACH FIELD. THIS LIST OF NAMES MAY BE SPECIFIED AS THE FIRST LINE, OR HEADER, OF A CHARACTER DELIMITED PEDIGREE DATA FILE; OR ALTERNATELY, IT MAY BE GIVEN AS A SET OF PARAMETERS IN A PEDIGREE BLOCK WITHIN THE PARAMETER FILE.

The content of each field in this list has no default semantic meaning and the field it identifies may be used for any purpose once read in.

Associating a field with a meaning, such as a pedigree ID, individual ID, marker phenotype, trait, etc. is accomplished by specifying additional parameters to map the field names to data field types. It is not necessary to specify all fields, and the parameter file may even specify fields that do not appear in the file so long as all required fields are in the pedigree data file. Whitespace is stripped from the beginning and end of the content of each field.

Here is an example of a comma delimited pedigree data file that includes the name of each field in a header line, one of the many possible character delimited formats:

```
pedid, indid, mom, dad, sex,     trait1, trait2, marker1
    1,     1,    ,    ,   M,   Affected,   10.3,     A/A
    1,     2,    ,    ,   F, Unaffected,    1.3,     a/a
    1,     3,   1,   2,   M,   Affected,    7.9,     A/a
```

Several options are provided to let the user modify the way a character delimited pedigree is processed by S.A.G.E. programs. The sets of characters that represent whitespace and delimiter characters may be redefined. There is an option that alters the way multiple consecutive delimiter characters are interpreted, by treating them as a single delimiter. This is extremely useful when reading multiple space delimited, or other fixed column formats, that do not include empty fields. Empty

[16] If necessary, column-delimited input files, such as those required for PAP can be imported into a spreadsheet program (Microsoft Excel, for example) and then exported in a character delimited format.

fields are a problem in this mode because it is not possible to detect them. For example, suppose each line in the following fixed column, space delimited, file is parsed into 6 fields using the delimiters and delimiter_mode options to read multiple blanks as a single field delimiter and skip leading and trailing blanks. The following delimited pedigree file is correctly specified:

```
PEDID INDID MOM DAD SEX TRAIT1
    1     1   0   0   M      0
    1     2   0   0   F      0
    1     3   1   2   M      2
```

If 0, the missing value code for parents and traits in this example, were replaced with a space character as indicated below, the resulting fixed column records would be parsed inconsistently. The two parents would have the SEX field as their MOM field, as well as other errors due to missing values not being detected.

```
PEDID INDID MOM DAD SEX TRAIT1
    1     1           M
    1     2           F
    1     3   1   2   M      2
```

2.3.1.2 Column delimited records

In a *column delimited* record the individual fields are distinguished from one another by their respective locations within the record. Each character in the record occupies a single location, or *column* and, starting with the leftmost column, the locations are identified as column 1, column 2, column 3 and so on. In the following example, the word "Queen" is located *at* column five[17], and the word "tarts" is located at column 36:

```
123456789_123456789_123456789_123456789_123456789_123456789_123456789_
The Queen of Hearts, she made some tarts, all on a summer day.
```

Once the appropriate pedigree record format has been determined, a single pedigree statement in the parameter file is used to specify the structure and content of the file; it has the following syntax:

```
pedigree[,column]
{
   [statement]*
}
```

Pedigree records are assumed to be character delimited by default. You can configure S.A.G.E. to accept column delimited records by including the `column` attribute, as in the following example:

[17] Meaning that the first letter of the word is *in* column five, and extending to the right for as many additional columns as there are remaining letters.

```
pedigree, column
{
   format = “4A5, 1X, A1, ...”
   individual_missing_value=0
   sex_code, male=1, female=2, unknown=?
   pedigree_id
   individual_id
   mother_id
   father_id
   sex_field
   ...
}
```

Each record in a pedigree data file stored in column delimited form comprises one or more sequential lines that contain fields that are at a fixed offset from the beginning of the line. No separation is required between fields because each field is of fixed length and each record is a fixed number of columns long. Parameters must be specified to define the order of the fields, the locations and widths of each field, as well as information on how each field is encoded. A technique borrowed from the FORTRAN programming language, called a FORTRAN format statement, is used to specify the locations, widths and other information on how the fields are encoded. A tutorial on how to create FORTRAN format statements can be found on the S.A.G.E. web site at http://darwin.cwru.edu/sagegui/help/fortran.html, or in most FORTRAN reference texts.

2.3.1.3 Pedigree Data Quality

Users are always well-advised to ensure that their pedigree data files are as error-free as possible[18], with particular attention paid to the correctness of family relationships within individual pedigrees. Nevertheless, S.A.G.E. programs are able to run in the presence of less-than-perfect data. With the exception of the five required fields mentioned previously, missing data will not prevent S.A.G.E. analyses from running to completion.

Note: *If the pedigree block of the parameter file lists a variable that does not appear (or is spelled differently) within the corresponding pedigree data file, S.A.G.E. will issue an appropriate error message and halt immediately.*

2.3.2 `Pedigree` Block Syntax

The following table shows the S.A.G.E. syntax for specifying the structure and content of a pedigree data file. Unless otherwise noted, all parameters and their corresponding attributes must be specified within a `pedigree` block of a parameter file.

The following table shows the syntax for a `pedigree` block statement:

[18] A well-known software apothegm is *“garbage in, garbage out”*, also expressed as the acronym *GIGO*.

parameter [, attribute]	**Explanation**	
`pedigree`	Starts a pedigree specification block.	
	Value Range	{column, character}
	Default Value	character
	Required	No
	Applicable Notes	1
`, file`	Specifies the name of a pedigree data file.	
	Value Range	Character string representing a valid file name.
	Default Value	None
	Required	No
	Applicable Notes	2, 3
`, delimited`	Specifies that the fields in the pedigree data file are character delimited, as opposed to the column-oriented format.	
	Value Range	delimited column
	Default Value	delimited
	Required	No
	Applicable Notes	2, 4

Notes

1. In this case, a comma is used to indicate that a value is being given, as in the following examples:

```
pedigree, character #This is the default setting
{
   #Pedigree specifications follow
   ...
}
pedigree, column
{
   #Pedigree specifications follow
   format = “4A5, 1X, A1, ...”
   ...
}
```

2. S.A.G.E. programs are capable of processing multiple pedigree data files simultaneously. This feature is especially useful for analyzing marker data that span the entire genome, in which case each chromosome is normally allocated to its own pedigree data file. To analyze data across multiple pedigree files, create a separate `pedigree` block for each pedigree file, and use the `file` attribute to name a particular file, as in the following example:

```
pedigree, delimited, file = “Chr1.ped”{
   delimiters               = "\t" # The ’\t’ indicates the tab key
   delimiter_mode           = multiple
   individual_missing_value = 0
   ...
   pedigree_id   = PID
   individual_id = ID
   parent_id     = P1
   parent_id     = P2
   sex_field     = sex
   ...
   allele = D1S2195a,         name = D1S2195
   allele = D1S2195b,         name = D1S2195
   allele = D1S1426a,         name = D1S1426
   allele = D1S1426b,         name = D1S1426
   ...
}

pedigree, delimited, file = “Chr2.ped”{
   delimiters               = "\t"
   delimiter_mode           = multiple
   individual_missing_value = 0
   ...
   pedigree_id   = PID
   individual_id = ID
   parent_id     = P1
   parent_id     = P2
   sex_field     = sex
   ...
   allele = D2S2195a,         name = D2S2195
   allele = D2S2195b,         name = D2S2195
   allele = D2S1426a,         name = D2S1426
   allele = D2S1426b,         name = D2S1426
   ...
}

pedigree, delimited, file = “Chr3.ped”{
   delimiters               = "\t"
   delimiter_mode           = multiple
   individual_missing_value = 0
   ...
   pedigree_id   = PID
   individual_id = ID
   parent_id     = P1
   parent_id     = P2
   sex_field     = sex
   ...
   allele = D3S2195a,         name = D3S2195
   allele = D3S2195b,         name = D3S2195
   allele = D3S1426a,         name = D3S1426
   allele = D3S1426b,         name = D3S1426
   ...
}
```

3. Even if the user specifies a filename at the start of each pedigree block, as shown in the above example, it is still necessary to supply the name of a pedigree filename on the program command line when running the program.

4. When using the `column` delimited format, the data fields will automatically be converted into and stored as the most appropriate data type for the given data. Thus, the fields for `pedigree_id`, `individual_id`, `parent_id`, and `sex_field` will all be stored as string types, whereas trait, phenotype and covariate values will be stored as numeric types (integers or reals) if at all possible. Some types of categorical data may only be storable as strings (for example, "High", "Medium" and "Low"). Also the user can force a numeric quantity to be stored as a character string by using the `string` option when specifying the field's name in the pedigree block.

Finally, the pedigree data file name specified at the S.A.G.E. program *command line* will automatically be assigned to the first pedigree block that does not specify the `file` attribute. Suppose the statements listed in the above example were contained in a parameter file named *hypertension_study.par*. Then the file attribute for the *first* pedigree block would be optional if the command line were:

```
>freq hypertension_study.par Chr1.ped
```

2.3.2.1 General Pedigree Formatting Options

parameter [, attribute]	**Explanation**
`format`	Designates either 1. a FORTRAN-style format statement used to specify the record layout of a column delimited pedigree file, or 2. a delimited, sequential listing of each field of a character delimited pedigree data file. Value Range: Quoted character string. Default Value: None Required: Not required for character delimited pedigree data files with a header record as the first entry. Required otherwise. Applicable Notes: 1, 2
`delimiters`	Specifies delimiter characters Value Range: Quoted character string. Default Value: `" ,\t"` Required: No Applicable Notes: 3
`whitespace`	Specifies characters that must be treated as whitespace. Value Range: Quoted character string. Default Value: " " (blank space) Required: No Applicable Notes: 3
`delimiter_mode`	Specifies delimiter interpretation mode. If set to `multiple`, then a set of successive delimiters in the pedigree data file will be treated as a single delimiter. Value Range: {single, multiple} Default Value: single Required: No Applicable Notes: 4
`verbose`	Specifies the number of individual records from the pedigree file to be printed to the program information output information file for visual verification of correctness. Value Range: {0, 1, 2, 3, ...} Default Value: 10, meaning that the first ten pedigree records will be printed to the information output file. Required: No Applicable Notes: None

require_record	Specifies whether or not to omit automatically generated "dummy" parent records in the pedigree data file for individuals with missing parent data. A value of **false** means that the parent records will be added to the analysis as needed.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	5

Notes

1. The `format` parameter is used to list the name of each field in the character delimited pedigree data file. Its value should be a delimited list of field names in the same order as those to be read from the file. The delimiter characters that separate each field name in this list are the same as those given in the `delimiters` parameter. If this parameter is not given, or is empty, then the first line of the character delimited pedigree file will be used to specify the `format` parameter.

2. In column delimited pedigree records, the fields are read into the program according to the order presented in the `format` parameter of the `pedigree` block.

3. The `delimiters` parameter specifies the characters that separate fields in each record. As a result, the delimiter characters should not be present in any fields. The default is that any comma (,) or tab (\t) character is interpreted as a delimiter character. Similarly, the `whitespace` parameter specifies characters that will be ignored when they appear at the beginning or end of fields.

4. The `delimiter_mode` parameter is used to alter how records are parsed. When the value of `delimiter_mode` is set to **single** each delimiter character found will terminate the current field. When the value of `delimiter_mode` is set to **multiple**, consecutive delimiters are treated as a single delimiter and delimiters that occur at the beginning and end of the record are ignored. Typically, tab and comma delimited files should be set to the value **single**, while space delimited files should be set to the value **multiple**.

5. By default, each individual in a pedigree must have one record in the pedigree data file. However, data on sibships without parent data are not uncommon. Distinguishing parent IDs must still be assigned to all individuals, but empty records for the dummy parents can be omitted if the `require_record` parameter is set to **false**.

2.3.2.2 Parameters for Individual and Family Identification Fields

parameter [, attribute]	**Explanation**

`pedigree_id`
`individual_id`
`parent_id`
`sex_field`
	Specifies pedigree field names for pedigree ID, individual ID, parental ID and Sex.

	Value Range: Character string representing the valid name of a field in the pedigree data file.
	Default Value: None
	Required: Yes
	Applicable Notes: 1, 2
`individual_missing_value`	Specifies codes for missing individuals (typically founders).
	Value Range: Character string. Typical values are: 0, "" (zero-length string), " " (blank space), 999, -1
	Default Value: "" (zero-length string)
	Required: No
	Applicable Notes: 3
`sex_code`	Declare codes used to specify sex of individuals in the pedigree.
	Value Range: N/A
	Default Value: N/A
	Required: No
	Applicable Notes: 4
`, male`	Specifies male sex code
	Value Range: Character string. Typical values are: 1, 0, M, m
	Default Value: M
	Required: No
	Applicable Notes: None
`, female`	Specifies female sex code.
	Value Range: Character string. Typical values are: 0, 1, F, f
	Default Value: F
	Required: No
	Applicable Notes: None

, missing	Specifies missing value code specifically for the sex field. May be different from the missing value code used for phenotype, trait and covariate fields.	
	Value Range	Character string. Typical values are: 0, “” (zero-length string), “ “ (blank space), 999, -1
	Default Value	“” (zero-length string)
	Required	No
	Applicable Notes	None
, trait	Declares sex code to be a binary trait, and subject to statistical analysis as such. Automatically creates a trait, sex_code, with values as follows: • male = 0 • female = 1	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	5

Notes

1. For character delimited pedigree records, the values assigned to the pedigree_id, individual_id, parent_id, and sex_field parameters are used to identify the order and location of the pedigree ID, individual ID, parent IDs and sex fields for each individual. Each of these values should match an element in the format parameter or the column header provided in the pedigree data file. The number of, and order in which, these fields are specified is arbitrary, and not all of these fields need appear in the pedigree data file, subject to several constraints. These constraints are that each record in the pedigree data file must include a pedigree ID and an individual ID, no more than one sex field and exactly two parent ID fields.

2. For column delimited pedigree records, the optional pedigree_id, individual_id, parent_id, and sex_field parameters should not be assigned any value. They are used to define the order and location of the pedigree ID, individual ID, parent IDs and the sex field for each individual. If any of these parameters are specified, then no default order is assumed and a parameter must be included for each field that is to be read in. If none are specified then it is assumed that the first five fields specified in the format statement correspond to the these fields in the following order:

   - pedigree ID
   - individual ID
   - Sex of individual

- First parent ID
- Second parent ID

3. This is the code that is used in the parent ID fields of founders.

4. Subject to the constraints described in note #2 (above), Sex codes may be specified separately or in the same parameter. e.g.:

```
# This is a valid sex_code parameter:
sex_code,male=M
sex_code,female=F
# ... and so also is this:
sex_code,male="M",female="F"
```

5. By including the attribute `trait`, the sex_code can be used as a quantitative (0,1) variable.

2.3.2.3 Parameters for Phenotype, Trait & Covariate Fields

parameter [, attribute]	**Explanation**	

`phenotype`
`trait`
`covariate`
	Specifies pedigree names for phenotype, trait and covariate fields.	

	Value Range	Character string representing the valid name of a field in the pedigree data file.
	Default Value	None
	Required	No
	Applicable Notes	1, 2
`, missing`	Specifies missing value code.	
	Value Range	Character string. Typical values are: 0, "" (zero-length string), " " (blank space), 999, -1
	Default Value	"" (zero-length string)
	Required	No
	Applicable Notes	None
`, binary`	Indicator for binary phenotype.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	None
`, affected`	Specifies code for affected status of binary phenotypes.	
	Value Range	Character string. Typical values are: A, 1, AFFECTED, Affected, yes, true, pos
	Default Value	1
	Required	No
	Applicable Notes	None
`, unaffected`	Specifies code for unaffected status of binary phenotypes.	
	Value Range	Character string. Typical values are: U, 0, UNAFFECTED, Unaffected, no, false, neg
	Default Value	0
	Required	No
	Applicable Notes	3

`, name`	Specifies a field name different from the name given in the pedigree header line. Value Range: Character string. Default Value: None Required: No Applicable Notes: 3
`string`	Designates a pedigree field that the user wishes to manipulate and/or report along with other data from an individual's record (an assay bar code, for example). Used for fields that do not fall into one of the previously mentioned types: • phenotype • trait • covariate • marker • allele • trait-marker Value Range: Character string. Default Value: None Required: No Applicable Notes: None

Notes

1. The `phenotype`, `trait` and `covariate` parameters all perform the same basic function; the values assigned to them identify fields in the pedigree data file that contain continuous or discrete phenotypic information The following guidelines may clarify when each different parameter should be used:

   (a) `phenotype` fields are a generic designation and convey no suggestion of how the field is to be used.

   (b) `trait` fields are typically selected by many analyses to be used as major variates.

   Thus these `parameters` simply provide hints to S.A.G.E. on how to make reasonable use of phenotypic information. Refer to the program specific documentation for information on the specific behaviors of these parameters and how to override them.

   Each trait field in a record may contain any character string that represents

   - a missing value code,
   - the affected or unaffected phenotype code (for binary phenotypes), or
   - a numeric phenotype (for quantitative phenotypes).

2. The `phenotype`, `trait` and `covariate` parameters should be included for each field in the pedigree data file that contains quantitative or categorical phenotypic information. The value of each such parameter should be set to the name by which it will be referred to in the rest of the parameter file and in the program output. Like other parameters that specify fields in a column delimited pedigree file, the order of the parameters is important and determines how the fields specified in the format statement are interpreted. **Remember: any field specified as a `phenotype` will *automatically* be analyzed by S.A.G.E. programs**, whereas fields specified as either trait or covariate will be analyzed optionally, depending on whether or not they have been listed within the relevant analysis block. The choice between (dependent) `trait` and `covariate` is largely a device to help the user remember how the

3. A `name` attribute may optionally be attached to `phenotype`, `trait` and `covariate` parameters, effectively creating an internal *alias* for the field. This feature is useful when the original names listed in the pedigree data file are obscure or unclear (usually due to abbreviation), and the user would like to create analyses, models and reports with more informative names.

   For example, if the pedigree data files contains four fields named Trait1, Trait2, Covariate1, and Affection, then the user may specify alternate names as in the following example:

```
phenotype = Trait1,     name = "Generic phenotype", missing = "X"
trait     = Trait2,     missing = "-99"
covariate = Covariate1, name = "Covariate #1"
trait     = Affection,  binary, affected = 1, unaffected = 0, missing = "?"
```

   As a result, the field originally designated as “Trait1” could be referenced as “Generic phenotype” within S.A.G.E. analyses, and similarly, the field originally designated as “Covariate1” could be referenced as “Covariate #1” within subsequent analyses.

2.3.2.4 Parameters for Genotype Data Fields

parameter [, attribute]	**Explanation**
allele marker	Specifies pedigree field name for a particular allele or marker. Value Range: Character string Default Value: None Required: No Applicable Notes: 1, 2, 3, 4, 5, 6
, x_linked	Indicator for X-linked marker. Value Range: N/A Default Value: None Required: No Applicable Notes: None
, y_linked	Indicator for Y-linked marker. Value Range: N/A Default Value: None Required: No Applicable Notes: None
, missing	Specifies missing value code Value Range: Character string. Typical values are: 0, "" (zero-length string), " " (blank space), 999, -1 Default Value: "" (zero-length string) Required: No Applicable Notes: None
, name	Specifies a marker or allele name different from the name given in the pedigree header line. Value Range: Character string. Default Value: None Required: No Applicable Notes: 6
, delimiter	Character used to delimit alleles of codominant markers in a pedigree data file. This is only necessary if markers are read in as a single field and are codominant. Value Range: Quoted character string. Default Value: "/" Required: No Applicable Notes: None

, minimum_allele_freq , minimum	Specifies minimum allele frequency for the marker. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 7, 8
, maximum_allele_freq , maximum	Specifies maximum allele frequency for the marker. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 7, 8
, equal , equal_allele_freq	Sets all allele frequencies to be equal. Value Range: N/A Default Value: None Required: No Applicable Notes: 7, 9
, complement , compl_allele_freq	Sets allele frequencies proportional to complementary values. Value Range: N/A Default Value: None Required: No Applicable Notes: 7, 10
trait_marker	Designates a trait for model-based linkage analysis (e.g., for MLOD or LODLINK) Value Range: Character string Default Value: None Required: No Applicable Notes: 11

Notes

1. The value is set to the name of the allele marker in the trait locus description file. The order of the parameters is important and determines how the fields specified in the format statement are interpreted.

2. Each allele field may be any character string that represents:

   - a missing value code, or
   - a single allele name.

   Each marker field may be any character string that represents:

   - a missing value code,
   - an allele name, the allele delimiter character and another allele name, or
   - a marker phenotype name.

3. A single `marker` parameter or two `allele` parameters should be included for each marker locus in the pedigree data file. Each marker locus field that is to be used should have a corresponding entry in the marker locus description file that defines its alleles, genotypes and

phenotype to genotype mapping. THOSE NOT FOUND IN THE MARKER LOCUS DESCRIPTION FILE WILL NOT BE ANALYZED BY ANY APPLICATION THAT REQUIRES THE MARKER LOCUS DESCRIPTION FILE.

4. E.g., to specify three markers to be read, named D42S1, D42S3, D42S4; marker D42S1 is given by two allele fields, and the others are marker fields:

```
marker=D42S3             # Order is irrelevant
allele=D42S1a,name=D42S1 # First allele for marker D42S1
marker=D42S4
allele=D42S1b,name=D42S1 # Second allele for marker D42S1
```

5. E.g., to specify reading three markers named "D42S1", "D42S2", "D42S3", a trait named "Trait1", another marker "D42S4", a trait-marker called "MOD", and a binary covariate named "Cov" in sequential fields in each record of a column delimited pedigree file:

```
# Order is important
allele = D42S1 # First allele of D42S1
allele = D42S1 # Second allele of D42S1
marker = D42S2
marker = D42S3
trait = Trait1
marker = D42S4
trait_marker = MOD
covariate = Cov, binary, affected=1, unaffected=2, missing=3
```

6. A `name` attribute may optionally be specified for `allele` and `marker` parameters. It should be specified when the name of the field in the pedigree data file is not the same as the name that appears in the marker locus description file. If a `name` attribute is not specified, the marker name is assumed to be the field name. The order in which these fields are specified is arbitrary and not all the fields need appear in the pedigree data file.

   A `trait_marker` parameter should be included for each trait in the pedigree data file that is to undergo a model-based linkage analysis. The value is set to the name of the trait-marker in the trait locus description file. The order in which these fields are specified is arbitrary and not all fields need appear in the pedigree data file.

7. For each locus, the information can be modified by adding the proper attributes to `marker`/`allele` parameter within the `pedigree` block.

8. For frequency adjustment, add attributes to the `marker/allele` statement within the `pedigree` block. For example:

```
pedigree {
   marker = D1S111, minimum_allele_freq = p
}
```

   This will replace with p all frequencies less than p, and then the frequencies will be normalized to sum to 1. The `maximum_allele_freq` parameter works in an analogous manner.

9. This will set all allele frequencies equal to 1/(number of alleles).

10. This will complement all allele frequencies and then normalize them to sum to 1.

11. A `trait_marker` parameter should be included for each trait in the pedigree data file that is to undergo a model-based linkage analysis. Thus the trait becomes like a marker and has requirements similar to those of a marker parameter, and hence is called a trait-marker. Instead of mixing markers and trait-markers in the same locus description file, each trait-marker should have an entry in the trait locus description file. THOSE NOT FOUND IN THE TRAIT LOCUS DESCRIPTION FILE WILL NOT BE ANALYZED.

2.3.3 The `marker` Sub-Block

The following table show the correct syntax for the `marker` sub-block:

parameter [, attribute]	**Explanation**
`allele_frequency`	Specifies allele frequency adjustment. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, equal`	Sets all allele frequencies to be equal. Value Range: N/A Default Value: None Required: No Applicable Notes: 1, 2
`, complement`	Sets all allele frequencies proportional to complementary values Value Range: N/A Default Value: None Required: No Applicable Notes: 1, 2
`, minimum`	Ensures that allele frequencies are not set below a minimum value. Note that after normalization (to sum to 1) some allele frequencies may be smaller than the set minimum or larger than the set maximum Value Range: [0,1] Default Value: None Required: No Applicable Notes: 1, 2
`, maximum`	Ensures that allele frequencies are not set above a maximum value. Value Range: [0,1] Default Value: None Required: No Applicable Notes: 1, 2

`allele_missing`	Specifies missing value code	
	Value Range	Character string. Typical values are: 0, "" (zero-length string), " " (blank space), 999, -1
	Default Value	"" (zero-length string)
	Required	No
	Applicable Notes	4, 5, 7
`allele_delimiter`	Character used to delimit alleles of codominant markers in a pedigree data file. This is only relevant if markers are read in as a single field and are codominant.	
	Value Range	Quoted character string.
	Default Value	"/"
	Required	No
	Applicable Notes	2, 3, 6

Notes

1. See notes 7 through 10 of Section 2.3.2.4
2. `equal` has higher precedence than `complement`, `complement` has higher `precedence` than `minimum` and `maximum`, and `minimum` and `maximum` have the same precedence.
3. For setting `delimiter` other than the default value of /, add the parameter `allele_delimiter` within the marker block.
4. The current global `allele_delimiter` statement in the parameter file will be ignored when the value for `allele_delimiter` is specified within the marker block.
5. For missing values other than the default , add the parameter `allele_missing` within the `marker` block. For example:

```
marker {
   allele_frequency, minimum = p
   allele_delimiter = ":"
   allele_missing = "."
}
```

6. The current `missing` attribute in the locus description file is ignored when the value for `allele_missing` in the `marker` block is specified.
7. For setting the delimiter to a value other than the default value of slash (/) add the `allele_delimiter` attribute to the `marker`/`allele` parameter within the `pedigree` block. For example:

```
pedigree {
   marker = D1S111, allele_delimiter = ":"
}
```

8. To specify a missing value other than the default, add the parameter `allele_missing` within the `allele` block. For example:

```
pedigree {
   marker = D1S111, allele_missing = “.”
}
```

2.3.4 Character Delimited Pedigree Data File Examples

Here is a typical pedigree data file in comma delimited format:

```
PID, ID,P1,P2, SEX, JUNK, D42S1,D42S2,D42S3,D42S4,D42S5,D42S6,TRAIT 1
1018, 1, 0, 0,   m,  0 0,   0 0,  0 0,  0 0,  0 0,  0 0,  0 0,   XXXX
1018, 2, 0, 0,   x,  0 0,   0 0,  0 0,  0 0,  0 0,  0 0,  0 0,   XXXX
1018, 3, 1, 2,   f,  5 3,   1 3,  6 7,  8 1,  2 2,  7 2,  7 1,   23.1
1018, 4, 1, 2,   f,  5 3,   1 3,  8 7,  8 1,  2 2,  7 2,  7 2,   44.1
1018, 5, 1, 2,   f,  5 3,   1 3,  8 7,  8 2,  2 2,  7 2,  7 1,   XXXX
1018, 6, 1, 2,   m,  5 3,   1 3,  8 7,  8 1,  3 2,  7 2,  4 1,    9.3
```

Suppose each record in the above pedigree data file is one line long and you want to use the following fields:

Field	Field Name
Pedigree ID	PID
Individual ID	ID
Sex field	SEX
First Parent ID	P1
Second Parent ID	P2
Trait	TRAIT 1
Marker D42S1	D42S1
Marker D42S2	D42S2
Marker D42S3	D42S3
Marker D42S4	D42S4
Marker D42S6	D42S6

then the following pedigree parameter can be used to read the pedigree data file:

```
pedigree # Example character delimited parameter statement
{
   # The following format string could be used if the pedigree file did not
   # already include a header line. Do NOT include both!
   # format="PID,ID,P1,P2,SEX,JUNK,D42S1,D42S2,D42S3,D42S4,D42S5,D42S6,TRAIT 1"

   pedigree_id=PID # Pedigree Field Specification
   individual_id=ID
   sex_field=SEX
   parent_id=P1
   parent_id=P2
   trait="TRAIT 1",name="Trait",missing="XXXX" # order is irrelevant
   marker=D42S4
   marker=D42S6
   marker=D42S1
   marker=D42S2
```

```
   marker=D42S3

   # Pedigree encoding information:
   individual_missing_value="0"
   sex_code, male="m",female="f",missing="x"
}
allele_delimiter=" " # Set the allele delimiter.
```

2.3.5 Column Delimited Pedigree File Examples

Here is a typical pedigree data file in column delimited format:

```
1018 1 m
1018 2 x
1018 3 1 2 f 2 5 3 1 3 6 7 8 1 2 2 7 2 7 1 23.1
1018 4 1 2 f 5 5 3 1 3 8 7 8 1 2 2 7 2 7 2 44.1
1018 5 1 2 f 2 5 3 1 3 8 7 8 2 2 2 7 2 7 1
1018 6 1 2 m 5 5 3 1 3 8 7 8 1 3 2 7 2 4 1  9.3
....|....|....|....|....|....|....|....|....|....|....|....|....|....|
```

Suppose each record is one line long with the following fields:

Field	Columns
Pedigree ID	1-4
Individual ID	6-7
Sex field	18
First Parent ID	9-10
Second Parent ID	12-13
Trait	65-68
Marker D42S1	29-33
Marker D42S2	35-39
Marker D42S3	41-45
Marker D42S4	47-51
Marker D42S5	53-57
Marker D42S6	59-63

Then the following pedigree parameter will read the pedigree data file:

```
# Set the marker delimiter to spaces
allele_delimiter=" "
# Example column delimited parameter
pedigree,column {
   # Pedigree Field Specification
   # FORTRAN Format Statement
   format="A4,1X,A2,T18,A1,T9,A2,1X,A2,T28,6(1x,A5),1X,A3"

   # Fields are listed in the order they are to be read
   marker="D42S1"
   marker="D42S2"
   marker="D42S3"
   marker="D42S4"
   marker="D42S5"
   marker="D42S6"
   trait="Trait 1"
```

```
    # Pedigree encoding information
    individual_missing_value=" "
    sex_code, male="m",female="f",missing="x"
}
```

2.4 User-Defined Functions

User-defined `function` parameters specify the creation of new traits, phenotypes, or covariates as a function of existing pedigree variables. Like other configuration parameters, function parameters may appear anywhere in the parameter file, but they are processed immediately after the pedigree data are read, IN THE ORDER IN WHICH THEY APPEAR. Thus, variables created by previous functions can be used in the specification of subsequent functions. Once created, a function variable may be used just like a trait, phenotype, or covariate read from a pedigree data file in all S.A.G.E programs except GENIBD, which does not currently support the use of function blocks.

2.4.1 The `function` Parameter

The following syntax table specifies the permissible parameter settings for the `function` parameter.

`parameter [, attribute]`	**Explanation**	
`function`	Starts a function block.	
	Value Range	N/A
	Default Value	None
	Required	N/A
	Applicable Notes	None

The following lists all parameters that may occur in a `function` block.

parameter [, attribute]	**Explanation**
`constant`	Names a constant. May appear multiple times if there are multiple constants to be specified. Value Range: Quoted character string. Default Value: None Required: No Applicable Notes: None
`, expression`	Specifies the expression used to calculate a value for the constant. Value Range: Quoted character string. Default Value: None Required: No Applicable Notes: See 2.4.2

`phenotype`
`trait`
`covariate`
	Names function variable. Only one function variable per function statement is allowed
	Value Range: Character string.
	Default Value: None
	Required: No
	Applicable Notes: 1, 2

`, expression`	Specifies the algebraic expression used to calculate a value for the function variable. Value Range: Character string. Default Value: None Required: Yes Applicable Notes: 3, 4, 5
`, missing`	Specifies missing value code. Value Range: Character string. Typical values are: 0, “” (zero-length string), “ ” (blank space), 999, -1 Default Value: “” (zero-length string) Required: No Applicable Notes: None
`, binary`	Indicates a binary phenotype. Value Range: N/A Default Value: None Required: N/A Applicable Notes: None

`, affected`	Specifies code for affected status of binary phenotypes. Value Range: Character string Default Value: 1 Required: No Applicable Notes: None
`, unaffected`	Specifies code for unaffected status of binary phenotypes. Value Range: Character string. Default Value: 0 Required: No Applicable Notes: None
`time_limit`	Specifies a time limit, in seconds, for evaluating constants and expressions. Value Range: {0, 1, 2, 3, ...} Default Value: 30 Required: No Applicable Notes: 6

Notes

1. The three possible function variable types are `phenotype`, `trait`, and `covariate`. `Phenotype` and `covariate` fields are a generic designation and convey no suggestion of how the field is to be used. `Trait` fields are typically selected by many analyses to be used as major variates. Thus these parameters simply provide hints to make reasonable use of phenotypic information. Refer to the program-specific documentation for specific information on the behaviors of these parameters and how they may be overridden.

2. The value may be a character string representing the name of a new phenotype, trait or covariate; however, THE FIRST CHARACTER MAY NOT BE A DIGIT. The name may not be that of an existing pedigree variable. Note that S.A.G.E. is CASE INSENSITIVE with respect to the names of traits, phenotypes and covariates. Thus the name “mean_bmi” is considered identical to “mean_BMI”.

3. A missing value for any of the variables in a function expression will result in a missing value for that function variable.

4. The value of `expression` should be an algebraic expression referring to one or more existing variables (traits, phenotypes, covariates or markers, either read from the pedigree data file or previously created as function variables) as well as allowable operators, elementary functions and constants. The variable name used in an expression may be specified by any character string, BUT THE FIRST CHARACTER OF THE STRING MAY NOT BE A DIGIT. Expressions should always be enclosed in double quotes (“”), and MUST BE ALL ON ONE LINE.

   Examples

   - Derive a new trait from an existing trait

     ```
     function {
        # Create trait x from traits HDL and LDL
        trait = x, expression="log(HDL) - log(LDL)"
     }
     ```

- Derive a new trait from an existing trait

```
function {
  # Create trait x from traits HDL and LDL
  trait = x, expression="log(HDL / LDL)"
}
```

  The above two functions (1 and 2) are equivalent. Note also that if LDL is 0, this trait is undefined (and hence a missing value is assigned to it).

- A more complex example

```
function {
   # Creates, from variables h1 and h2, the covariate “average” whose
   # value is 1 if (h1 + h2) / 2 is greater than .275, and 0 otherwise.
   # If the program cannot evaluate an expression in 2 seconds or less,
   # it will abort, giving a fatal error message.
   time_limit=2
   constant=gamma, expression = .275
   covariate = “average”, expression = “(h1 + h2)/2 > gamma”
}
```

5. Variable names are not case sensitive, but elementary function and constant names are.

6. The `time_limit` parameter is provided to avoid situations where the calculation of values takes an inordinate amount of time. In most cases it need not be changed.

2.4.2 Expression Elements

This section describes the constants, operators, and functions that may be used in function block expressions.

2.4.2.1 Constants

The following constants may be used[19]:

Constant Type	Example
Any rational number	-3.0, 5, 1.23, ...
e	2.71828...
pi	3.14159...

[19]The two names *e* and *pi* are reserved and may not be used as the names of phenotypes, traits or covariates.

2.4.2.2 Operators and Expressions

Operator	Meaning	Example
`-`	Unary Negation	`expression = "-BMI"` `expression = "-X + 10"`
`**`	Exponentiation	`expression = "DBH**2" (power of two)` `expression = "X**0.5" (square root)`
`%`	Modulus (remainder after integer division)	`expression = "Age % 10"`
`/`	Division	`expression = "Age / 10"`
`*`	Multiplication	`expression = "Weight * 1.19"` `expression = "Weight * (Affected == 1)"`[a]
`-`	Subtraction	`expression = "Height - AvgHeight "`
`+`	Addition	`expression = "Var_X + Var_Y"`
`!=, <>`	Not equal to	`expression = "Affected <> 0"` `expression = "Sex != M"`
`==`	Equal to (***two*** equal signs!)	`expression = "Affected == 0"` `expression = "Sex == M"`
`>=`	Greater than or equal to	`expression = "Age >= 65"` `expression = "Age >= (65 - AgeOnset)"`
`>`	Greater than	`expression = "Age > 65"` `expression = "Age > (65 - AgeOnset)"`
`<=`	Less than or equal to	`expression = "Age <= 65"` `expression = "Age <= (65 - AgeOnset)"`
`<`	Less than	`expression = "Age < 65"` `expression = "Age < (65 - AgeOnset)"`
`not`	Logical negation	`expression = "not(Affected)"` `expression = "not(Sex == M)"`
`and`	Logical AND (intersection)	`expression = "(Affected and Sex == M)"` `expression = "(Age > 65 and not (Affected == 1))"`
`or`	Logical OR (union)	`expression = "(Affected or Sex == M)"`
`()`	Parentheses (logical & arithmetic grouping)[b]	`expression = "(x + (y * z))"`

[a] Parentheses may be used to group terms in the normal manner. In this example the use of the comparison operator, "==", creates a logical expression whose evaluation results in either 0 or 1.

[b] Strictly speaking, the parentheses are not operators so much as punctuation, a subtle difference that does not prevent their being listed in this table.

Operators are evaluated in order of operator precedence from highest to lowest in the following list. Except when there are parentheses (see below), all operators of an equal precedence are evaluated before operators of lower precedence (from left to right, except for comparison operators which are evaluated from right to left). Operator precedence from highest to lowest is as follows (operators on the same line have equal precedence):

Operator	Precedence Level
`- (Unary Negation)`	(Highest) 8
`**`	7
`*, /, %`	6
`+, -`	5
`<, <=, >, >=, ==, <>, !=`	4
`not`	3
`and`	2
`or`	(Lowest) 1

Operator precedence may be overridden by parentheses. Expressions in parentheses are evaluated first (BRACKETS OR BRACES MAY NOT BE USED). Multiple parentheses are permissible; the computation starts within the innermost parentheses and works outwards.

The logical expression operators *or*, *and* & *not* may be used only in expressions that evaluate to either zero (false) or one (true). Otherwise the results are not defined.

2.4.2.3 Elementary Functions

The following elementary functions may be used:

Function Syntax	**Mathematical Equivalent**	**Meaning**
exp(x)	e^x	e to the power of x
log(x)	$\ln x$	natural log of x
log10(x)	$\log_{10} x$	log to the base 10 of x
pow(x,y)	x^y	x raised to the power of y
sqrt(x)	$\sqrt{x}$, or $x^{\frac{1}{2}}$	positive square root of x
fabs(s)	$\|x\|$	absolute value of x
ceil(x)		smallest integer >= x, for any x
floor(x)		largest integer <= x, for any x
min(x_1,x_2, ..., x_n)		$x_i$such that x_i<= x_j, for j = 1, 2, ..., n
max(x_1,x_2, ..., x_n)		$x_i$such that x_i<= x_j, for j = 1, 2, ..., n

2.4.2.4 Marker Functions

The following functions are available for codominant markers only (In these functions the second argument (allele value) must be in single quotes as shown.):

- **`dominant`**`(marker, 'Ai')` or **`dom`**`(marker, 'Ai')`

returns the value 1 or 0 based on the alleles present at the specified marker locus, where A* is any allele other than A_i, as follows[20]:

A_i/A_i, A_i/*A** returns 1

*A**/*A** returns 0,

- **`recessive`**`(marker, 'Ai')` or **`rec`**`(marker, 'Ai')`

  returns the value 1 or 0 based on the alleles present at the specified marker locus, where *A** is any allele other than A_i, as follows:

  A_i/A_i returns 1

  A_i/*A**, *A**/*A** returns 0,

- **`additive`**`(marker, 'Ai')` or **`add`**`(marker, 'Ai')`

  returns the value 2, 1, or 0 based on the alleles present at the specified marker locus, where A* is any allele other than A_i, as follows:

  A_i/A_i returns 2

  A_i/*A** returns 1

  *A**/*A** returns 0,

- **`genotype`**`(marker, 'Ai', 'Aj')` or **`gen`**`(marker, 'Ai', 'Aj')`

  returns the value 1 or 0 based on the alleles present at the specified marker locus, where *A** is any allele other than A_i or A_j, as follows:

  A_i/A_j, A_j/A_i returns 1

  A_i/*A**, A_j/*A**, *A**/A_i, *A**/A_j, *A**/*A** returns 0,

1. An ABO example:

```
function {
   # Creates, from marker ABO, the covariate x whose value is 1 if marker
   # ABO genotypes are AB or BA, and 0 otherwise.
   covariate = x,
   expression = "dom(ABO, 'A') and dom(ABO, 'B')"
}
```

2. Another ABO example:

```
function {
   # Creates, from marker ABO, the covariate x whose value is 1 if marker
   # ABO genotypes are AB or BA, and 0 otherwise.
   covariate = x,
   expression = "gen(ABO, 'A', 'B')"
}
```

   Note The above two functions (from examples 1 and 2) are equivalent. Also, if ABO is missing, the trait x will also be missing.

3. Another marker example:

[20] The examples assume that / is the allele delimiter within genotypes. However, a different delimiter could be used.

```
function {
   # Creates, from marker D42S8 and trait z, a phenotype, y, whose value
   # is z if allele q1 is present at marker D42S8, and 0 otherwise.
   phenotype = y,
   expression = "dominant(D42S8, 'q1') * z"
}
```

2.4.3 Mean-Adjusted and Variance-Adjusted Data

S.A.G.E. provides the option of generating mean-adjusted, variance-adjusted or standardized values for each class of a stratification variable of a given trait, phenotype or covariate. There are two basic steps to creating an adjusted variable:

1. Specify the classes of the stratification variable[21].
2. Define a new variable to be adjusted with respect to these classes.

The newly created variable can then be used in a S.AG.E. analysis.

2.4.3.1 The Binning Algorithm

The method by which data and their associated summary statistics are subdivided into multiple *bins* (i.e., classes) is herein referred to as a *binning algorithm*, and the primary goal of this algorithm is to ensure that each class has a sufficient level of membership to support valid statistical inference. First, we define the following set of variables:

- N - the minimum number of individuals required for each bin
- N_j- the number of individuals occurring in class *j*
- *k* - the number of ordered classes
- x_j - the statistic of interest for the j-the class, *before* the binning algorithm has been applied
- $x_j^{'}$- the statistic of interest for the j-the class, *after* the binning algorithm has been applied

Then the algorithm for allocating data observations across the bins is

1. If $\sum_{j=1}^{k} N_j < N$, then the algorithm cannot be performed.
2. For each class $i \in \{1, 2, \ldots, j\}$ let $a_i$be a starting point and $b_i$be the end point of class i on the number line, where
   $a_i = \begin{matrix} 0, & if\ j = 0 \\ b_{i-1}, & if\ j > 0 \end{matrix}$ and $b_i = a_j + N_i/N$.

[21] A stratification variable is not strictly required, and the data adjustment procedure described here can be used to transform any given trait, phenotype or covariate into its mean- or variance-adjusted values, as explained in Sec. 2.4.3.6.

3. For any class *j* for which $N_j < N$, let *l* be the proportion of $x_j^{'}$ that comes from x_1, ..., x_{j-1}, where
   $l = max\left[min\left(\frac{N-N_j}{2N},\, a_j\right),\, \frac{N-N_j}{2N} - min\left(\frac{N-N_j}{2N},\, n_k - b_j\right)\right]$.
   Then let A = $a_i - l$ and $x_j = \sum_{i=1}^{k} a_i x_i$, where:
   $$a_i = \begin{cases} 0,\, if\, b_i < A\, or\, a_i \geq A+1 \\ b_i - A,\, if\, a_i < A\, and\, b_i > A \\ N_i/N,\, if\, a_i \geq A\, and\, b_i \leq A+1 \\ A+1-a_i,\, if\, a_i \geq A\, and\, b_i > A+1 \end{cases}$$

2.4.3.2 Specifying the Classes

Specify each class within a function block as the values of an `expression` attribute for the `covariate` parameter[22]. The following example shows how to create three classes of a covariate[23] named "Age":

```
function {
   covariate = class1, expression = "(Age <= 15)"
}

function {
   covariate = class2, expression = "(Age > 15 and Age <= 30)"
}

function {
   covariate = class3, expression = "(Age > 30)"
}
```

IMPORTANT! It is essential that the classification scheme partitions the data into exhaustive and mutually exclusive subsets with respect to the classification variable ("Age", in this case). If the data are not partitioned correctly, the resultant mean- and variance-adjusted variables will not be reliable.

2.4.3.3 Creating a Mean-Adjusted Variable

Once the classes of the stratification variable have been defined, the mean-adjusted values of some other trait (or phenotype or covariate) can be calculated using the classes of the stratification variable to define classes of the trait. Specify the mean-adjustment within a function block as the value of an `expression` attribute for the `trait` parameter[24]. Assuming the pedigree data file lists a phenotype called "BP", the following example shows how to create a new mean-adjusted variable named "BP_AgeAdjMean":

[22] The classes could also be created from a trait or phenotype; however, it usually makes more sense to create them from a covariate.

[23] In this example, "Age" is assumed to be a covariate; however the stratification variable could also be a trait or phenotype.

[24] The variable could also be created as a phenotype or covariate.

```
function {
   trait      = BP_AgeAdjMean,
   expression = "mean_adj(Age, BP, 10, class1, class2, class3)"
}
```

Note the use of the special keyword, **mean_adj**. This is what tells S.A.G.E. to add a new set of information to the internal computer representation of the pedigree file.

The value of the third argument to **mean_adj** (*10* in this example) determines the minimum number of items required for the classes. If, after the data have been stratified, any of the resultant classes has less than the minimum number of entries, then a special algorithm is employed to "borrow" values from the ordered list of values in neighboring classes until the minimum number has been reached for the underrepresented class. Note that for the resulting mean-adjusted variable to be meaningful, the classes of the stratification variable must be in natural order. The user can disable this feature by entering a zero (0) as the minimum number.

The variable, *BP_AgeAdjMean*, essentially becomes a new trait, and is surreptitiously added to the internal representation of the pedigree data file (the original file is left unchanged). In this case, there are three different means computed:

- $\bar{x}_1$: the mean BP for individuals whose age is in the range {0, 1, 2, ..., 15},
- $\bar{x}_2$: the mean BP for individuals whose age is in the range {16, 17, ..., 30} and
- $\bar{x}_3$: the mean BP for individuals whose age is in the range {31, 32, 33, ...}.

If BP_i is the blood pressure value for individual *i*, then the value of "BP_AgeAdjMean" *for that individual* will be

- $BP_i - \bar{x}_1$; if the individual's age is in the range {0, 1, 2, ..., 15},
- $BP_i - \bar{x}_2$; if the individual's age is in the range {16, 17, ..., 30} and
- $BP_i - \bar{x}_3$; if the individual's age is in the range {31, 32, 33, ...}.

2.4.3.4 Creating a Variance-Adjusted Variable

The procedure for creating a variance-adjusted variable is analogous. The following example shows how to create a new variable named "BP_AgeAdjVar":

```
function {
   trait      = BP_AgeAdjVar,
   expression = "var_adj(Age, BP, 10, class1, class2, class3)"
}
```

Here, the required keyword is **var_adj**, and the resultant values of BP_AgeAdjVar (for an arbitrary individual *i*) will be:

- BP_i/s_1: if the individual's age is in the range {0, 1, 2, ..., 15},
- BP_i/s_2: if the individual's age is in the range {16, 17, ..., 30} and
- BP_i/s_3: if the individual's age is in the range {31, 32, 33, ...},

where s_i (*i* = 1, 2, 3) is the sample standard deviation of the trait *BP* for age class *i*.

2.4.3.5 Creating a Z-Score Variable

A standardized variable is obtained as follows:

```
function {
   trait      = BP_AgeZScore
   expression = "z_score(Age, BP, 10, class1, class2, class3)"
}
```

In this last example, the required keyword is **z_score**, and the values of BP_AgeZScore (for an arbitrary individual *i*) will be:

- $(BP_i - \bar{x}_1)/s_1$: if the individual's age is in the range {0, 1, 2, ..., 15},
- $(BP_i - \bar{x}_2)/s_2$: if the individual's age is in the range {16, 17, ..., 30} and
- $(BP_i - \bar{x}_3)/s_3$: if the individual's age is in the range {31, 32, 33, ...}.

2.4.3.6 Creating Adjusted Variables Without Classes

It is also possible to create an adjusted variable that does not depend on the classes of a stratification variable. The result is simply the mean-adjusted, variance-adjusted or standardized value of a given variable with respect to the entire sample.

To create a mean-adjusted variable (*BP_AdjMean*) from the variable *BP*, write:

```
function {
   trait      = BP_AdjMean,
   expression = "mean_adj(BP)"
}
```

To create a variance-adjusted variable (*BP_AdjVar*) from the variable *BP*, write:

```
function {
   trait      = BP_AdjVar,
   expression = "var_adj(BP)"
}
```

To create a standardized variable (*BP_Normalized*) from the variable *BP*, write:

```
function {
   trait      = BP_Normalized,
   expression = "z_score(BP)"
}
```

2.4.4 Data Trimming and Winsorization

S.A.G.E. provides a way to minimize the adverse impact of outlier data by creating variables that either *trim* or *Winsorize* the tails of the distributions as shown in the following example:

A set of random numbers:

0.38	.058	0.13	0.15	0.51	0.27	0.10	0.19	0.12	0.86

After sorting and positioning:

0.10	0.12	0.13	0.15	0.19	0.27	0.38	0.51	0.58	0.86
1	2	3	4	5	6	7	8	9	10

After trimming:

-	-	0.13	0.15	0.19	0.27	0.38	0.51	-	-
1	2	3	4	5	6	7	8	9	10

After Winsorization:

0.13	0.13	0.13	0.15	0.19	0.27	0.38	0.51	0.51	0.51
1	2	3	4	5	6	7	8	9	10

Data that are subjected to the trim function are effectively thrown out of the analysis, whereas Winsorized data are revalued to a quantity that corresponds to some critical point along the distribution. Details of the method can be found in Armitage & Colton (1990).

2.4.4.1 Creating a trimmed variable

Create a trimmed variable using the **trim** S.A.G.E. keyword as in the following example:

```
function{trait=LNIGE_trim, expression="trim(LNIGE,0.02)"}
```

The **trim** function takes two arguments:

1. the name of a trait, phenotype or covariate (here “LNIGE”) previously specified in the pedigree block
2. a value $\gamma \in (0, 1)$, representing the “amount” of data to be trimmed (the value 0.02 will result in trimming 1% of the values in each tail of the distribution).

The newly created variable, *LNIGE_trim*, can be used in the same manner as any other trait, covariate or phenotype within S.A.G.E. applications.

2.4.4.2 Creating a Winsorized variable

Create a winsorized variable using the **winsor** S.A.G.E. keyword as in the following example:

```
function{trait=LNIGE_wins, expression="winsor(LNIGE,0.02)"}
```

The **winsor** function takes two arguments:

1. the name of a trait, phenotype or covariate (here "LNIGE") previously specified in the pedigree block
2. a value $\gamma \in (0, 1)$, representing the "amount" of data to be winsorized (the value 0.02 will result in 1% of the values in each tail of the distribution being replaced by the corresponding 1 and 99 percentiles).

The newly created indicator variable, *LNIGE_wins*, can be used in the same manner as any other trait, covariate or phenotype within S.A.G.E. applications.

2.4.5 The Transmitted and Untransmitted Allele Indicators (TAI and UTAI)

The problem of performing a transmission disequilibrium test (TDT) to assess the linkage between a marker locus and a quantitative trait was addressed in a paper by George et al (1999), who proposed a linear-regression approach in which the disease trait (assumed to be continuous) is the dependent variable, *Y*. The primary independent variable in the model, *X*, is an indicator variable that reflects whether or not a given allele was transmitted to the individual from a heterozygous parent (see Figure 1). The authors refer to X as a *transmission status variable* which is referred to here by the slightly more accurate term: *transmitted allele indicator* (TAI).

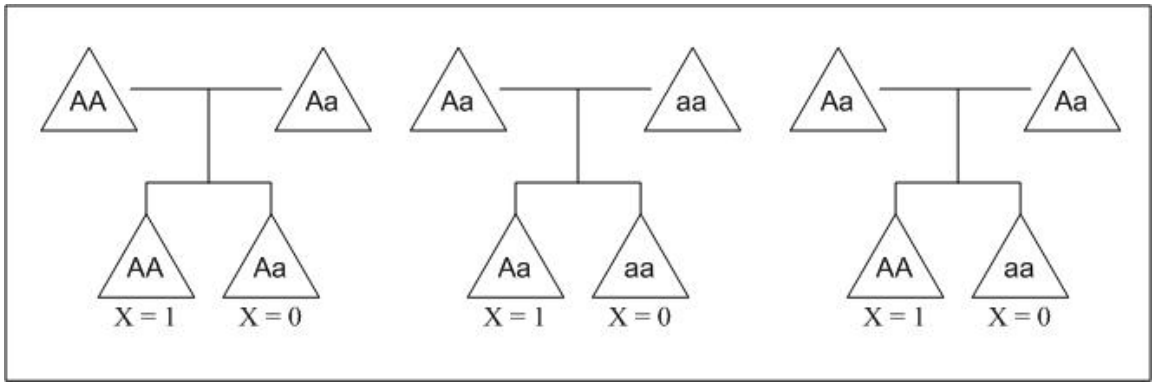


Figure 1: Offspring who are informative for linkage, from relevant parental matings. ***A*** is the associated allele of interest, and *X* is the transmitted allele indicator variable such that *X*=1 if ***A*** was transmitted from a heterozygous parent, and *X*=0 otherwise.

For example, consider a diallelic locus {A, a} and suppose we wish to determine the TAI with respect to allele 'A'. Then the TAI values computed for a given individual would be as shown in the table below, which also indicates the UTAI as well[25]:

[25]When the marker locus has more than two alleles, we appropriately extend this indicator to make use of the maximum amount of information available in an unbiased fashion. See the theory section of the TDTEX program in this manual.

	Parental Genotype	Offspring Genotype	Informative?	TAI Value	UTAI Value
1	AA x AA	AA	N		
2	**AA x Aa**	**AA**	**Y**	**1**	**0**
3	**AA x Aa**	**Aa**	**Y**	**0**	**1**
4	AA x aa	Aa	N		
5	**Aa x Aa**	**AA**	**Y**	**1**	**0**
6	Aa x Aa	Aa	N		
7	**Aa x Aa**	**aa**	**Y**	**0**	**1**
8	**Aa x aa**	**Aa**	**Y**	**1**	**0**
9	**Aa x aa**	**aa**	**Y**	**0**	**1**
10	aa x aa	aa	N		

The following table illustrates the way the TAI variable is created in S.A.G.E. (the table for UTAI variables is similar). The TAI user-defined function generates an internal table, similar to the one depicted below, that associates the value of the indicator status variable with respect to each individual in the pedigree, and for each allele. TAI variables for pedigree founders are simply interpreted as missing information.

Transmitted Allele Indicator (TAI) Table

PID	ID	$M_1A_{1(1)}$	$M_1A_{2(1)}$	...	$M_1A_{k(1)}$	$M_2A_{1(2)}$	$M_2A_{2(2)}$	...	$M_2A_{k(2)}$	...	$M_nA_{1(n)}$	$M_nA_{2(n)}$	...	$M_nA_{k(n)}$
001	001			...				...		...			...	
001	002			...				...		...			...	
001	003	0	0	...	1	0	0	...	1	...	0	0	...	0
001	004	1	0	...	0	0	1	...	0	...	0	1	...	1
001	005	0	0	...	0	1	0	...	0	...	1	0	...	1
002	001			...				...		...			...	
002	002			...				...		...			...	
002	003	0	0	...	1	1	0	...	0	...	0	0	...	0
002	004	1	0	...	0	0	1	...	0	...	0	1	...	1
002	005	0	1	...	0	1	0	...	1	...	1	1	...	0
003	001			...				...		...			...	
003	002			...				...		...			...	
003	003	0	0	...	1	0	0	...	0	...	0	0	...	0
003	004	1	0	...	0	0	0	...	1	...	0	0	...	0
003	005	1	0	...	0	1	0	...	0	...	1	0	...	1

2.4.5.1 Creating TAI and UTAI Variables

To specify TAI and/or UTAI variables for a single marker, create a function block that defines the new variables using the **tai** and **utai** keywords as in the following example:

```
pedigree{
   .
   .
   .
   allele = “M1A”, name = “M1” #marker specified in pedigree block
```

```
   allele = "M1a", name = "M1" #marker specified in pedigree block
}
function{
   trait = M1A_tai, expression="tai(M1, A)"
}
function{
   trait = M1a_tai, expression="tai(M1, a)"
}
function{
   trait = M1A_utai, expression="utai(M1, A)"
}
function{
   trait = M1a_utai, expression="utai(M1, a)"
}
```

The newly created indicator variables, *M1A_tai*, *M1a_tai*, *M1A_utai* and *M1a_utai*, can be used in the same manner as any other trait, covariate or phenotype within S.A.G.E. applications[26].

2.5 Locus Description Files

The marker locus description file and the trait locus description file follow the same format as each other and contain records that define allele frequencies and phenotype to genotype mappings. The marker locus description file contains a record for each marker locus. The trait locus description file contains a record for each discrete trait, or "trait-marker", that is to undergo a model-based linkage analysis. A record must be included in the corresponding locus description file for each marker locus or trait-marker to be analyzed, and these records may appear in any order. All marker loci and trait-markers listed in the parameter file and/or the genome description file should be present in the marker locus description file: THOSE NOT PRESENT THERE ARE IGNORED**.** In the case of fully penetrant and codominant markers, the program FREQ can be used to prepare the marker locus description file.

Optionally, at the beginning of the file, a missing value code may be included. This code indicates which values, if any, indicate a missing marker value. For example, the user may specify one of the following lines as the first line of the marker locus description file:

```
Missing=CODE
missing=CODE
```

or

```
MISSING=CODE
```

[26] S.A.G.E. currently requires the user to create these TAI and UTAI variables one at a time via the `function` block syntax as shown in the example. This would be an admittedly impractical solution when analyzing large numbers of markers (SNPs, for example), and the S.A.G.E. development team is working to provide shortcuts and improvements to this feature in a future release.

where CODE is the case-sensitive missing value code for ALL marker phenotypes or pairs of marker alleles. If, when identifying a marker phenotype by its two alleles, either allele is missing, the phenotype of the individual will be set to missing for that marker. If only one allele is missing, a non-fatal error message will be generated.

The locus description file should contain the following items for **each** locus to be analyzed:

1. The name of the locus.
2. A set of records that give the allele frequencies. The records should follow this format:

   allele_symbol = population allele frequency

   The user should supply the information for the items on both sides of the "=" symbol. There can be any number of spaces before or after the equal sign as long as the allele symbol and its frequency remain on the same line. There should be only one allele symbol per line. The allele symbol can consist of up to 10 characters. It is also permissible to list just the alleles, leaving out " = allele frequency" from every line. When this is done, equal allele frequencies are substituted for each allele listed for that locus[27]. This option is useful when the marker locus description file is used in conjunction with a program that does not use allele frequencies (e.g., ASSOC).
3. A semicolon indicating the end of the alleles. This semicolon can be either on a line by itself or on the same line following the last population allele frequency of the set.
4. A set of records that defines the phenosets (i.e., the sets of genotypes compatible with each phenotype). The records should follow this format:

   phenotype symbol = {$A1_1/A2_1[=P_1]$,...,$A1_m/A2_m[=P_m]$} where

   $A1_1$ is the symbol for allele #1 in the first genotype of this phenoset;

   $A2_1$is the symbol for allele #2 in the first genotype of this phenoset;

   . . .

   $A1_m$ is the symbol for allele #1 in the m-th (last) genotype of this phenoset;

   $A2_m$ is the symbol for allele #2 in the m-th (last) genotype of this phenoset,

and P_1...P_m are the penetrance values of the phenotype, i.e., the probabilities of the phenotype given the genotype. The penetrance values are strictly optional[28].

There can be any number of spaces before or after the equal sign(s). The phenoset should begin with either a left curly brace ({) or less-than symbol (<), and end with a corresponding right curly brace (}) or greater-than symbol (>). The first and second allele of each genotype must be separated by a slash (/) or otherwise specified allele delimiter. Each genotype within the phenoset should be separated by a comma. This record may wrap onto as many lines as necessary. Complete the set by repeating this record for each phenotype at this locus. Any phenotype symbol that is not included here is interpreted as a missing phenotype value. The order of the alleles in a genotype has no effect.

Example:

[27] This is a change from previous versions of S.A.G.E.

[28] This is a change from previous versions of S.A.G.E. If no value is indicated, the phenotype is assumed to be fully penetrant and a value of 1 is assumed.

```
LOCA
A = 0.5                    #(alleles/phenotype names are arbitrary
B = 0.25                   #and need not be sequential)
C = 0.25
;
1 = {A/A,A/B,A/C}          #(A is dominant over B and C, and
2 = {B/B,B/C}              #B is dominant over C)
3 = {C/C}
;
ABO
A1 = 0.1904
A2 = 0.0612
B = 0.0728
O = 0.6756
;
1 = { A1/A1, A1/A2, A1/O } #(A1 is dominant over A2 and O)
2 = { A1/B }
3 = { A2/A2, A2/O }        #(A2 is dominant over O)
4 = { A2/B }
5 = { B/B, B/O }           #(B is dominant over O)
6 = { O/O }
;
```

If a locus is fully penetrant and codominant it is not necessary to include the records for phenotypes mentioned in note 4 above. The program will generate the phenotype symbol by concatenating the two allele symbols of the genotype and putting a delimiter character between them (typically a /, but this can be modified in the parameter file). However, the semicolon indicating the end of the phenotypes still has to be included.

Example:

The following two locus descriptions are equivalent:

```
A
1 = 0.645
2 = 0.223
3 = 0.1325;
1/1 = {1/1}
1/2 = {1/2}
2/2 = {2/2}
1/3 = {1/3}
2/3 = {2/3}
3/3 = {3/3}
;
```

```
A
1 = 0.6455
2 = 0.2230
3 = 0.1325;
;
```

Trait-markers are specified similarly. As an example, suppose we have a trait `"Disease"`, and an underlying model with two disease alleles (allele 1 has frequency 10% and allele 2 has frequency 90%) and two phenotypes (`A` = affected, `U` = unaffected). Suppose that we are assuming that allele 1 predisposes toward the expression of affection, and furthermore that it is recessive to allele 2.

Our penetrance table might look something like this:

	1/1	1/2	2/2
A	0.6	0.01	0.01
U	0.4	0.99	0.99

i.e., 60% penetrance and a sporadic rate of 1%. The trait locus description file would then contain the following entry:

```
Disease
1 = 0.10
2 = 0.90
;
A = { 1/1 = 0.6, 1/2 = 0.01, 2/2 = 0.01 }
U = { 1/1 = 0.4, 1/2 = 0.99, 2/2 = 0.99 }
;
```

Note that the trait need not be binary (any number of phenotypes may be specified), and the locus may have more than two alleles. For any particular phenotype, the sum of all (here two) penetrances must equal 1.

2.6 Genome Description File

The genome description file describes the genomic region(s) used in analyses that require the order of, and distances between, linked marker loci. A genome is defined with at least one genomic region. This region contains the names of sequentially ordered marker loci and the distances or recombination fractions between pairs of adjacent markers. A map function is used to translate genetic map distances to and from recombination fractions. The general form of the file is as follows:

```
genome="genome name"[,map="map function"]
{
   [region1]
   [region2]
   [region3]
   .
   .
   .
}
```

The `genome name` can be any name desired. The map attribute allows specification of a map function, which can be either the Haldane or Kosambi map functions. If no map function is supplied, Haldane is assumed. Map functions are not used during single-point analysis.

Each genomic region is described as follows:

```
region="region name"
{
   [marker and distance parameters]
}
```

The `region name` is used to identify the region being defined. If no name is specified, "`region n`" is used, where n is the number of the region within the genome. The attribute x_linked is needed after region name to indicate the region as X-linked region as follows:

```
region="region name", x_linked
{
   [marker and distance parameters]
}
```

The following parameters are available within a region sub-block:

parameter [, attribute]	**Explanation**
`marker`	Indicates a marker name. If none is specified, the marker is ignored. There should be one marker parameter for each marker in the region. Value Range: Character string Default Value: None Required: Yes Applicable Notes: 1
`missing`	Indicates a marker that is missing from the data, but is included as a placeholder. Value Range: Default Value: None Required: No Applicable Notes: None
`distance`	Specifies distance, in centimorgans, between adjacent marker parameters. Value Range: $(0, \infty)$ Default Value: None Required: Yes Applicable Notes: 2
`theta`	Specifies distance between adjacent markers in terms of the recombination fraction θ. Value Range: [0, 0.5] Default Value: None Required: Yes Applicable Notes: 2

Notes

1. In the program output the first marker in each region is located at an absolute distance of 0.0 cM and all further markers are measured from this location in the map units specified by the map attribute.

2. There is a maximum of one distance or theta (i.e., recombination fraction) parameter between each pair of markers (marker and missing parameters.) When doing multi-point analysis, there must be either a distance or theta parameter between each pair of adjacent markers.

   Here is an example of a typical Genome Description file:

```
Genome {
   # No genome name or map function specified.
   # Haldane map function is assumed
   # No region name specified, so the name is
   # assumed to be "region 1"
   region {
      distance =  154.7100             # Initial distance is measured from pter
      marker   = "D4S2999"             # at  154.7100 cM
      distance =  0.0000000001
      marker   = "D4S3021"             # at  154.7100 cM
      distance =  0.4200000000
```

```
    marker   = "D4S2976"             # at  155.1300 cM
    distance =  0.3200000000
    marker   = "D4S2631"             # at  155.4500 cM
    distance =  0.1700000000
    marker   = "D4S3016"             # at  155.6200 cM
    distance =  0.7000000000
    marker   = "D4S1556"             # at  156.3200 cM
    distance =  1.2300000000
    marker   = "TSC0785934"          # at  157.5500 cM
    distance =  0.0000000001
    marker   = "TSC1312016"          # at  157.5500 cM
    distance =  0.0000000001
    marker   = "TSC0439917"          # at  157.5500 cM
    .
    .
    .
    }
}
```

2.7 IBD Sharing File

The IBD sharing file stores the probability distribution of allele-sharing identical-by-descent (IBD) between pairs of individuals at specific locations. The header of the file contains the n names (L1, L2, , Ln) of the locations at which IBD sharing information is stored for each pair of relatives. These locations are referred to as markers, even though they may not correspond to observed marker loci in a given dataset. The body of the file contains a line for each pair of individuals that includes the following fields:

- pedigree ID
- First individual ID
- Second individual ID
- f_0: The probability that the pair shares 0 alleles IBD at marker L_1
- f_{1m-1p}: The probability that the pair shares 1 maternal allele minus the probability that it shares 1 paternal allele IBD at marker L_1
- f_2: The probability that the pair shares 2 alleles IBD at marker L_1

. . .

- f_0: The probability that the pair shares 0 alleles IBD at marker L_n
- f_{1m-1p}: The probability that the pair shares 1 maternal allele minus the probability that it shares 1 paternal allele IBD at marker L_n
- f_2: The probability that the pair shares 2 alleles IBD at marker L_n

The probability that a pair shares one allele IBD at a given marker is $f_1 = 1 - f_0 - f_2$, where f_0 and f_2 are the probabilities that the given pair shares 0 and 2 alleles IBD at the marker. Similarly, the estimated proportion of alleles shared IBD is $f_2 + \frac{1}{2} f_1$. These probabilities are conditional on the pedigree and marker information available and are usually denoted $\hat{f}$ in the literature.

Notes

1. IBD sharing files are typically generated as a result of some prior analysis and will virtually never need to be constructed manually. IBD sharing files are generated by the program GENIBD and used as input to programs such as SIBPAL.
2. Packages other than S.A.G.E. may be able to use IBD sharing files as input, but the format in S.A.G.E. is subject to change.
3. The number of markers may be very large, so each line of the IBD sharing file can be extremely long. Loading these files into text-editors, especially those that wrap or truncate long lines, is not recommended.
4. IBD sharing files may be extremely large if there are many pairs and markers. When performing analyses on extremely large pedigrees and/or genome screens, IBD sharing files may consume disk space in excess of a gigabyte. Thankfully, IBD sharing files are amenable to many forms of data compression when not in use.

2.8 Information Output Files

An information output file is generated by all S.A.G.E. programs and contains diagnostic output generated during program execution. Typically, this includes information about how pedigree data files were read and diagnostic information on pedigree structure, phenotypes and marker loci. This file is named "program.inf", indicating the name of the specific program that was run[29]. No analysis results are stored in this file.

All S.A.G.E. programs that read trait or marker locus description files or genome description files generate the genome information File. This file contains diagnostic information on each marker or trait phenotype and genotype. This file is named "genome.inf". No analysis results are stored in this file, though errors relating to the markers and traits may be.

2.9 Analysis Output Files

All S.A.G.E. programs produce one or more analysis output files, which contain the results of the analyses. The number of analysis output files, their names and contents are program specific. Analysis output files may even correspond to other S.A.G.E. input file types. E.g., the analysis output file from GENIBD is an IBD sharing file that is an input file for SIBPAL.

[29] Eg., fcor.inf, mlod.inf, segreg.inf, etc.

Chapter 3

PEDINFO

PEDINFO provides many useful descriptive statistics on pedigree structure including means, variances and histograms of family, sibship and pedigree sizes, and counts of each type of relative pair. Statistics based on trait phenotypic status (i.e., limited to traits not having missing values) are can also be requested.

3.1 Limitations

PEDINFO cannot correctly process a pedigree that contains loops; however, the program does indicate the presence of loops within the given pedigree data file.

3.2 Theory

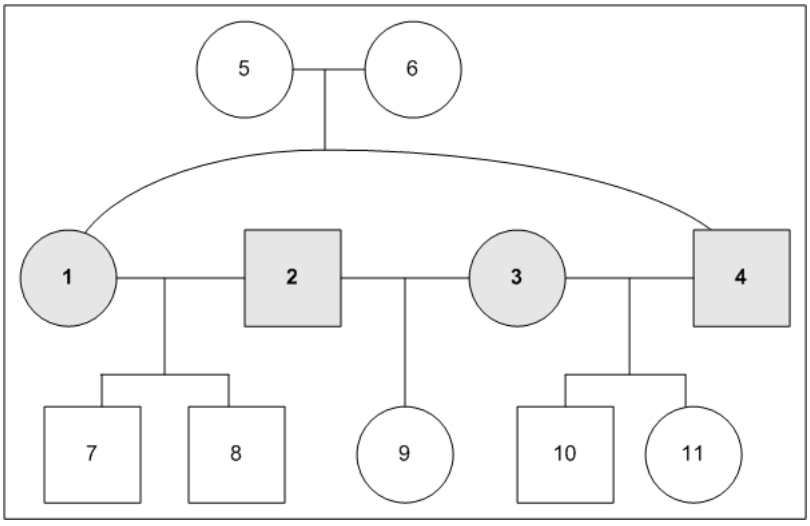
3.2.1 Terminology

PEDINFO operates by iterating over the pedigree structures and keeps counts and distribution information of various elements. The following table defines some terms used in PEDINFO that are not defined elsewhere in this document:

Term	Definition
Brother Pair	A pair of individuals who share the same parents and are both male.
Sister Pair	A pair of individuals who share the same parents and are both female.
Generations	In a pedigree without loops the number of generations is one more than the length of the longest chain of offspring relationships.
Inheritance Vector Bits	For a given pedigree, the maximum value of 2n-f maximized over its constituent pedigrees, where n is the number of non-founders and f is the number of founders in each constituent pedigree. This number represents the largest number of bits in an inheritance vector that would be used in certain types of multi-point analysis algorithms. It is useful for evaluating whether it is feasible to run such algorithms on a given pedigree.

3.2.2 Problematic Family Structures

A *marriage ring* is a chain of spouses who form a cycle, for example, the founders in the following pedigree (individuals #1, 2, 3 and 4):

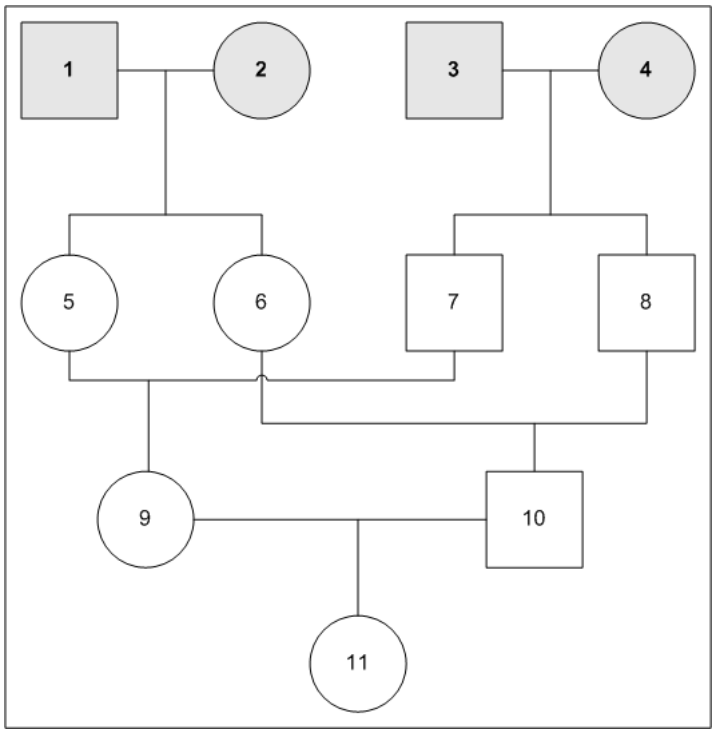


Individuals in a chain of spouses are listed in the output as individuals with multiple mates. Individuals in marriage ring are listed in both the individual column and the mates column.

```
|=========================================================================|
|                      Individuals With Multiple Mates                    |
|(Pedigree, Individual)   Mates                                           |
|-------------------------------------------------------------------------|
|(1, 1)                    4, 2                                           |
|(1, 2)                    1, 3                                           |
|(1, 3)                    4, 2                                           |
|(1, 4)                    3, 1                                           |
|=========================================================================|
```

These rings can cause computational difficulties for current programs using full pedigree structure information and are therefore enumerated so that users can break these rings as they see fit.

Loops indicate either consanguineous (marriage between relatives) or other marriage loops, eg., two brothers married to two sisters:



Consanguineous and other marriage loops can also cause computational difficulties for current programs using full pedigree structure information and may also need to be broken. To facilitate this process, consanguineous matings are listed by pedigree and by the pair of relatives who have mated.

```
|=========================================================================|
|                        Consanguineous Mating Pairs                      |
|Pedigree         Pair                                                    |
|-------------------------------------------------------------------------|
|1                10, 9                                                   |
===========================================================================
```

When there are marriage rings or loops in the pedigree, the pairs are not independent and therefore the pair counts output by PEDINFO may not be accurate.

In the case of a consanguineous pedigree, the number of generations may be indeterminable and “undet” will appear in the generation statistics output by PEDINFO.

Generation Statistics		Nuc Family Statistics		Inh Vector Bit Stats	
# of Gens	# of Peds	# of Nuc Fams	# of Peds	# of Bits	# of Peds
undet.	5	0 - 2	1	3 - 4	2
		3 - 4	4	5 - 8	3

Breaking loops can be done by duplicating individuals or by removing certain connecting individuals.

3.3 Program Input

File Type	Description
PEDINFO parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual including fields for identifiers, sex, parents, trait and marker data.

3.3.1 The `pedinfo` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main PEDINFO parameter.

`parameter [, attribute]`	**Explanation**
`pedinfo`	Starts a PEDINFO analysis block. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension. Value Range: Character string representing a valid file name Default Value: pedinfo.out Required: No Applicable Notes: None

3.3.2 The `pedinfo` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `pedinfo` sub-block.

`parameter [, attribute]`	**Explanation**

`trait`
`phenotype`
`covariate`
	Specifies a variable to be used in the analysis.
	Value Range: Character string representing the name of a trait, phenotype or covariate listed in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 1

`each_pedigree`	Specifies option to calculate statistics on a pedigree-by-pedigree basis Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2
`suppress_general`	Specifies suppression of output for non-trait statistics. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None

Notes

1. The `trait`, `phenotype` and `covariate` parameters are used to specify trait, covariate, or phenotype variables for which statistics are to be calculated. The value of a variable parameter should be set to the name of a variable field read from the pedigree data file or created using a `function` statement. To be included in the statistics an individual must not have a missing value for this variable. More than one variable may be specified in an analysis block, in which case an individual must have non-missing values for each of the specified variables to be included in the statistics. If a single binary variable is specified for analysis, counts of pairs that are concordant unaffected, discordant, concordant affected and uninformative will be displayed. If no variables are specified, only non-variable information (i.e., based on pedigree structure alone) will be used to determine counts.
2. The `each_pedigree` parameter is used to specify whether results should be calculated for each pedigree separately in addition to a set of results for all the pedigrees taken as a whole.

The following are all valid pedinfo statements and could all occur within the same parameter file:

```
# A pedinfo statement that runs with all default values
pedinfo

# A pedinfo statement that runs with all default values
pedinfo
{
}

# A pedinfo statement that specifies the name of an output file
# and requests a separate report for each pedigree
pedinfo,out=allpeds.out
{
   each_pedigree=true
}

# A pedinfo statement that specifies 2 traits,
# for each of which an individual must have no missing data
# to be included in the trait-specific pedigree statistics
pedinfo,out=analysis1 {
   trait=A
   trait=hematocrit
}

# A final example
pedinfo,out=output {
   phenotype=B
   each_pedigree=true
}
```

3.4 Program Execution

PEDINFO is run via a command line interface on the supported UNIX and Windows platforms. This requires that the S.A.G.E. programs are properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running PEDINFO from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line:

```
>pedinfo
S.A.G.E. v5.x -- PEDINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: ./pedinfo <parameters>  <pedigree>
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of PEDINFO may look like the following:

```
>pedinfo pedinfo.par example.ped
S.A.G.E. v5.x -- PEDINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading parameter file.....................done.
Reading pedigree file.....................
from example.ped............done.
Sorting pedigrees.........................done.
Generating statistics.....................done.
Analysis complete!
```

3.5 Program Output

Output files produced by PEDINFO containing results and diagnostic information are:

Filename	File Type	Description
pedinfo.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
pedinfo.out	Analysis output file	Contains a table of summary statistics for all pedigrees combined, and optionally a table for each individual pedigree.

3.5.1 Information Output File

The PEDINFO information output file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read in, are included with all programs in S.A.G.E. and are very useful for debugging many common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be carefully checked to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not run as expected.

3.5.2 Analysis Output File

The PEDINFO analysis output file may contain the following types of tables:

- Tables of statistics pertaining to the structure of all of the data as a whole.
- Tables of statistics pertaining to the structure of a single pedigree.
- Tables of statistics pertaining to a specific trait or set of traits for the data as a whole. For individuals to be counted in these tables they must be informative (not have a missing value) for each relevant trait[1].
- Tables of statistics pertaining to a specific trait or set of traits for a single pedigree. For individuals to be counted in these tables they must be informative (not have a missing value) for each relevant trait.

[1] In the counts of nuclear families per pedigree included in these tables, a nuclear family is defined as a family having at least one informative parent and one informative child.

3.6 Example Output File

Here are some typical examples of PEDINFO output:

```
================================================================================
|                    General Statistics: All Pedigrees                         |
|==============================================================================|
|                 |    Count| Mean Size +/- Std. Dev. (     Min.,      Max.) |
|------------------------------------------------------------------------------|
|Pedigrees        |        2|      6.50 +/-      2.50 (        4,         9) |
|==============================================================================|
|Generation Statistics||  Nuc  Family Statistics ||  Inh  Vector Bit Stats   |
|# of Gens | # of Peds|| # of Nuc Fams |# of Peds||   # of Bits   |# of Peds |
|------------------------------------------------------------------------------|
|         2|         2||      0 -     2|        1||      0 -     2|         2|
|          |          ||      3 -     4|        1||               |          |
|==============================================================================|
|                 |    Count| Mean Size +/- Std. Dev. (     Min.,      Max.) |
|------------------------------------------------------------------------------|
|Sibships         |        4|      1.25 +/-      0.43 (        1,         2) |
|==============================================================================|
|Constituent   |          || Marriage    |          ||             |          |
|Pedigrees     |         4|| Rings       |         0|| Loops       |        0|
|==============================================================================|
|Pairs                   |       Count||  Individuals           |       Count|
|------------------------------------------------------------------------------|
|Parent/Off              |          10||  Male                  |           5|
|Sib/Sib                 |           1||  Female                |           8|
|Sis/Sis                 |           0||  Unknown               |           0|
|Bro/Bro                 |           0||        Total           |          13|
|Bro/Sis                 |           1||                        |            |
|Grandp.                 |           0||  Founder               |           8|
|Avunc.                  |           0||  Non-founder           |           5|
|Half Sib                |           0||  Singleton             |           0|
|Cousin                  |           0||        Total           |          13|
|==============================================================================|
|                      Individuals With Multiple Mates                         |
|none                                                                          |
|==============================================================================|
|                        Consanguineous Mating Pairs                           |
|none                                                                          |
================================================================================
```

```
================================================================================
|            Trait Statistics: All Pedigrees, Trait - HEMATOCRIT               |
|==============================================================================|
|                |  0 Parents w. Data|   1 Parent w. Data|  2 Parents w. Data|
|------------------------------------------------------------------------------|
|Sibships        |                  0|                  0|                  4|
|==============================================================================|
|                |    Count| Mean Size +/- Std. Dev. (     Min.,      Max.) |
|------------------------------------------------------------------------------|
|Sibships        |        4|      1.00 +/-      ---  (        1,          1) |
|------------------------------------------------------------------------------|
|Pedigrees       |        2|      6.00 +/-      3.00 (        3,          9) |
|==============================================================================|
|  Nuc  Family Statistics                                                      |
| # of Nuc Fams |# of Peds                                                     |
|------------------------------------------------------------------------------|
|      0 -     2|        1                                                     |
|      3 -     4|        1                                                     |
|==============================================================================|
|           |         |    Pair       ||           |         |    Pair       |
|Pairs      |    Count|    Correlation||Pairs      |    Count|    Correlation|
|------------------------------------------------------------------------------|
|Parent/Off |        8|           --- ||Grandp.    |        0|           --- |
|Sib/Sib    |        0|           --- ||Avunc.     |        0|           --- |
|Sis/Sis    |        0|           --- ||Half Sib   |        0|           --- |
|Bro/Bro    |        0|           --- ||Cousin     |        0|           --- |
|Bro/Sis    |        0|           --- ||           |         |               |
|==============================================================================|
|Indiv.'s |    Count|      Mean +/- Std. Dev. (     Min.,      Max.) |        |
|------------------------------------------------------------------------------|
|Male     |        4|     40.00 +/-      ---  (    40.00,     40.00) |        |
|Female   |        8|     40.00 +/-      ---  (    40.00,     40.00) |        |
|Unknown  |        0|      ---  +/-      ---  (     --- ,      --- ) |        |
|    All  |       12|     40.00 +/-      ---  (    40.00,     40.00) |        |
|         |         |                                                |        |
|Founder  |        8|     40.00 +/-      ---  (    40.00,     40.00) |        |
|Nonfound.|        4|     40.00 +/-      ---  (    40.00,     40.00) |        |
|Singleton|        0|      ---  +/-      ---  (     --- ,      --- ) |        |
|    All  |       12|     40.00 +/-      ---  (    40.00,     40.00) |        |
================================================================================
```

```
================================================================================
|                     General Statistics: All Pedigrees                        |
|==============================================================================|
|                 |    Count| Mean Size +/- Std. Dev. (     Min.,      Max.) |
|------------------------------------------------------------------------------|
|Pedigrees        |        1|      6.00 +/-      ---  (        6,         6) |
|==============================================================================|
|Generation Statistics||  Nuc  Family Statistics ||  Inh  Vector Bit Stats    |
|# of Gens | # of Peds|| # of Nuc Fams |# of Peds||   # of Bits   |# of Peds |
|------------------------------------------------------------------------------|
|         3|         1||      0 -     2|        1||      3 -     4|         1|
|==============================================================================|
|                 |    Count| Mean Size +/- Std. Dev. (     Min.,      Max.) |
|------------------------------------------------------------------------------|
|Sibships         |        2|      1.50 +/-      0.50 (        1,         2) |
|==============================================================================|
|Constituent   |          || Marriage    |          ||            |         |
|Pedigrees     |         1|| Rings       |         0|| Loops      |        0|
|==============================================================================|
|Pairs                    |       Count||  Individuals           |       Count|
|------------------------------------------------------------------------------|
|Parent/Off               |           6||  Male                  |           3|
|Sib/Sib                  |           1||  Female                |           3|
|Sis/Sis                  |           0||  Unknown               |           0|
|Bro/Bro                  |           0||        Total           |           6|
|Bro/Sis                  |           1||                        |            |
|Grandp.                  |           2||  Founder               |           3|
|Avunc.                   |           1||  Non-founder           |           3|
|Half Sib                 |           0||  Singleton             |           0|
|Cousin                   |           0||        Total           |           6|
|==============================================================================|
|                      Individuals With Multiple Mates                         |
|none                                                                          |
|==============================================================================|
|                        Consanguineous Mating Pairs                           |
|none                                                                          |
================================================================================
```

Trait Statistics: All Pedigrees, Trait - DISEASE					
	0 Parents w. Data		1 Parent w. Data		2 Parents w. Data
Sibships	0		0		2
	Count	Mean Size +/- Std. Dev.		Min.,	Max.)
Sibships	2	1.50 +/-	0.50 (	1,	2)
Pedigrees	1	6.00 +/-	--- (	6,	6)

Nuc Family Statistics	
# of Nuc Fams	# of Peds
0 - 2	1

Pairs	Concord. Unaff.	Discord.	Concord. Aff.	Uninform.	Total	Corr.
Parent/Off	1	3	2	0	6	0.0000
Sib/Sib	0	1	0	0	1	---
Sis/Sis	0	0	0	0	0	---
Bro/Bro	0	0	0	0	0	---
Bro/Sis	0	1	0	0	1	---
Grandp.	0	1	1	0	2	---
Avunc.	0	1	0	0	1	---
Half Sib	0	0	0	0	0	---
Cousin	0	0	0	0	0	---

Indiv.'s	Affected	Unaff.	Missing	Total		
Male	0	3	0	3		
Female	3	0	0	3		
Unknown	0	0	0	0		
Total	3	3	0	6		
Founder	1	2	0	3		
Nonfound.	2	1	0	3		
Singleton	0	0	0	0		
Total	3	3	0	6		

Chapter 4

FCOR

FCOR can estimate multivariate familial correlations, and their asymptotic standard errors, for all pair types available in a set of pedigrees. FCOR also estimates the equivalent count of independent pairs that could theoretically have been used to obtain the same standard error for each correlation. Familial correlations for both subtypes (sex-specific) and main types (pooled sex-specific correlations) are estimated, together with their corresponding asymptotic standard errors. The variance-covariance matrices of the estimated correlations are calculated and a test for homogeneity of correlations among subtypes can be performed.

4.1 Limitations

Further analysis, such as adjusting for covariates, is not supported. Standard errors are based on asymptotic theory and in some cases may not be estimable.

4.2 Theory

The theory underlying all the calculations performed by FCOR is given in Keen and Elston (2003).

4.2.1 Relative Pairs and Treatment of Missing Data

For each type of familial correlation, FCOR uses all pairs of relatives where both members have data on at least one trait in common. All other pairs of that type are excluded from the calculations and output.

4.2.2 Relative Pairs Naming Convention

We call relative pair types that depend on individuals' sexes *subtypes*, and those that do not *main types*. FCOR uses one of the following two naming conventions for each relative pair type, depending on the user's choice. The default is Sex Specific Name.

4.2.2.1 Non-Sex Specific Name for a Pair Type

Each type of relative pair is described by a name for the (non-sex specific) relationship and, additionally, one or two lists of M's (for male) or F's (for female) within square brackets ([ ]) that describe their ancestry. These lists represent a sequence of sexes for the individuals that comprise a lineage connecting the individuals in the pair. Relationships that represent direct descent (that is, parent-offspring, grandparental, great grandparental, and so on) are displayed as a single list starting with the ancestor and ending with the descendant. Relationships that do not represent direct descent (for example, sibling, nephew-uncle, cousin, and so on) are displayed with two lists. The first list begins with the first individual in the pair and terminates at the common ancestral nuclear family. The second list begins with the common nuclear family and terminates at the other individual in the pair. If the common ancestor is a parent of two half siblings (that is, the second-to-last ancestors are half siblings), then the sex of the single common ancestor is displayed between the two lists.

4.2.2.2 Sex Specific Name for a Pair Type

Each type of relative pair is described by the name of the relationship with the sequence of ancestors in the lineage.

4.2.2.3 Examples

Non-Sex Specific Name		Sex Specific Name
parent:offspring	[MF]	father:daughter
grandparent	[MFM]	grandfather-through-mother:grandson
great-grandparent	[MMFM]	great-grandfather-through-mother's-father:great-grandson
sibling	[F,M]	sister:brother
half-sibling	[F,M,F]	paternal-half-sister:half-sister
cousin	[MF,FM]	male-cousin-through-mother:male-cousin-through-mother
half-cousin	[MF,F,MF]	maternal-male-half-cousin-through-mother:female-half-cousin-through-father
second-cousin	[MFF,MFF]	male-second-cousin-through-mother's-mother:female-second-cousin-through-mother's-father
avuncular	[M,FM]	uncle-through-mother:nephew
half-avuncular	[M,M,FM]	paternal-half-uncle-through-mother:half-nephew
great-avuncular	[M,FMF]	great-uncle-through-father's-mother:great-niece
first-cousin-once-removed	[MF,FMF]	male-cousin-through-mother:male-cousin-through-mother, removed-daughter
first-cousin-twice-removed	[MF,FFMM]	male-cousin-through-mother:female-cousin-through-mother, removed-son's-son
second-cousin-once-removed	[MFF,FMFF]	male-second-cousin-through-mother's-mother:female-second-cousin-through-father's-mother,removed-daughter

4.2.3 Correlations

Consider the *N* pairs of the observations of a particular type in the sample as a set of random two-element vectors $\{ (x_i, y_i) \}_{i=1}^{N}$. These vectors are not assumed to be independent or uncorrelated, but the structure of the pairwise correlations among them is known via the pedigree structure. The pedigree correlation between the two random variables x_i and y_i is consistently estimated from a random sample of pedigrees by

$$r_{xy} = \frac{\sum_{i=1}^{N} w_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N} w_i (x_i - \overline{x})^2 \sum_{i=1}^{N} w_i (y_i - \overline{y})^2}} \tag{4.1}$$

where $\overline{x} = \sum_i w_i x_i / \sum_i w_i$ and $\overline{y} = \sum_i w_i y_i / \sum_i w_i$ for arbitrary non-negative weights $\{w_i\}$.

The pedigree correlation r_{xy} can represent either an interclass correlation (if two classes of persons are involved) or an intraclass correlation (if only one class of persons is involved), for either the same trait or different traits. For example, suppose r_{xy} represents an interclass correlation between a trait measured on a woman and a trait measured on her daughter's son. Then we can let the random variable x denote the woman's trait and the random variable y denote the trait on one of her daughter's sons. In this way, grandmother is adopted as one class and daughters' sons as another class. Given a random sample of pedigrees, the pedigrees are scanned to produce *N* pairs from the two classes whereby for the ith pair, x_i equals the value of a woman's trait and y_i equals the value of a trait of one of the woman's daughter's sons. If a woman's daughter has more than one son, then there will be pairs that share the observation of the same grandmother – for which an accounting must be made when calculating the asymptotic standard error. Moreover, a sibling correlation will also need to be accommodated as well. If a woman has more than one daughter, and each has at least one son, then a cousin correlation will need to be accommodated for cousin pairs who share a common grandmother. The situation becomes even more complex when pedigrees contain, for example, pairs of grandmothers as sisters. Note that one or more of the correlations needed to calculate a standard error may not be estimable.

A special case in pedigrees is that of intraclass correlations. These correlations are defined and estimated with respect to, for example: siblings; cousins; brother/brother; and female-cousin/female-cousin. The intraclass correlations are not necessarily restricted to the same random variable, trait, or phenotype. In the situation of relating different random variables with members of the same class of individuals, the correlations are referred to here as intraclass cross-correlations. All possible pairs within a class of individuals are formed with the random variable x representing one trait and the random variable y representing the other trait measured on a different member of the same class.

The user can specify the largest number of generations to be considered when choosing the classes for which correlations are to be calculated. If this is not specified, FCOR can examine the pedigree structure and then decides for itself what pedigree correlations should be calculated for a given random sample of pedigrees (for large pedigrees this calculation can consume a lot of computer time). Thus correlations that are not calculated are those that cannot be adequately estimated from the sample (a minimum of three pairs must be available to estimate any correlation).

4.2.4 Asymptotic Standard Errors of Correlations

The asymptotic standard error of a given correlation is estimated by using a second-order Taylor series expansion and replacing all correlation parameters with their respective estimates. If a required correlation is not estimable, it is replaced by zero or the user can suppress the calculation of such a standard error.

4.2.5 Equivalent Pair Count

The equivalent pair count for a specific familial correlation coefficient estimate is the estimated number of independent pairs of observations that would have a standard error the same as the value estimated for the specific familial correlation. Letting r denote the value of the correlation and s the estimate of its standard error, the equivalent count is estimated by

$$EquivalentCount = \frac{1}{2}\left[N_0 + \sqrt{N_0^2 + \frac{22(1-r^2)}{s^2}r^2}\right], \tag{4.2}$$

where $N_0 = 1 + (1-r^2)^2/s^2$.

4.2.6 Test for Homogeneity of Correlations among Subtypes

This is a test of the hypothesis that all subtypes within a main type have the same correlation.

The main types are grouped by non-sex specific relationship type. For example, the SELF main type relationship contains two subtypes – male self and female self. As another example, the PARENT:OFFSPRING main type has four subtypes – father:son, father:daughter, mother:son, and mother:daughter.

Subtype correlations are always computed first, and then, if requested, main type correlations are calculated by appropriately pooling subtype correlations. After grouping subtypes into main types, chi-square statistics and p-values are calculated to test homogeneity of correlations among the subtypes within each main type. Under the null hypothesis of homogeneity, if only one dependent variable is being analyzed, the test statistic has an approximate chi-square distribution with degrees of freedom equal to the number of subtypes minus one. If multiple dependent variables are being analyzed, the test of homogeneity includes homogeneity of all possible subtype correlations. Thus, if there are *k* subtypes on *p* traits, then the number of degrees of freedom is$(k-1)\,p^2$ for interclass correlations, and$(k-1)\,p\,(p-1)\,/2$ for intraclass correlations.

4.3 Program Input

File Type	Description
FCOR Parameter File	Specifies the parameters and options with which to perform a particular analysis.
Pedigree Data File	Contains delimited records for each individual including fields for identifiers, sex, parents, and trait data.

4.3.1 The `fcor` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main FCOR parameter.

`parameter [, attribute]`	**Explanation**	
`fcor`	Starts an FCOR analysis block	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	fcor
	Required	No
	Applicable Notes	None

4.3.2 The `fcor` Block

The following syntax table specifies the permissible parameter and attribute settings for the `fcor` block.

`parameter [, attribute]`	**Explanation**
`trait`	Specifies a trait to be used in the analysis. Value Range: Character string Default Value: None Required: Yes Applicable Notes: 1
`interclass_weight`	Specifies the weight to be used for interclass correlations. Value Range: **{**pair_wise, pair, uniform, reduced_group, reduced**}** Default Value: pair_wise Required: No Applicable Notes: 2
`intraclass_weight`	Specifies the weight to be used for intraclass correlations. Value Range: **{**pair_wise, pair, uniform, reduced_group, reduced**}** Default Value: pair_wise Required: No Applicable Notes: 2
`type`	Specifies calculation of correlations for subtypes only, or for main relative types only, or for both main relative types and subtypes. Value Range: {subtypes, maintypes, both} Default Value: subtypes Required: No Applicable Notes: 3
`, tabular`	Specifies option to produce an additional output file of alternate tabular structure for all correlation types. Value Range: N/A Default Value: None Required: No Applicable Notes: See 4.6.3
`standard_error`	Option to calculate asymptotic standard errors. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 4

, conservative	Specifies calculation of asymptotic standard errors conservatively. Value Range: N/A Default Value: None Required: No Applicable Notes: 4
, pairs	Specifies option to produce additional output file indicating, for each standard error, the smallest number of pairs used to calculate any of the required correlations. Value Range: N/A Default Value: None Required: No Applicable Notes: See 4.6.4
sex_name	Option to print out relationship name with sex. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 5
generation_limit	Specifies the largest number of steps permissible between a given pair of individuals and their closest common ancestor. Relative pairs who exceed the specified value will be excluded from analysis. Value Range: {1, 2, 3, ...} Default Value: 2 Required: No Applicable Notes: 6
homogeneity_test	Specifies the calculation of chi-square statistics and associated p-values for homogeneity tests. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 7
var_cov	Starts a parameter sub-block to specify the options to print out the variance-covariance matrix of correlation estimates. Value Range: N/A Default Value: None Required: No Applicable Notes: 8
, single	Prints a single matrix for each trait specified. Value Range: N/A Default Value: None Required: No Applicable Notes: 8

`, joint`	Prints a joint matrix for each pair of traits specified. Value Range: N/A Default Value: None Required: No Applicable Notes: 8

Notes

1. The value of a `trait` parameter should be set to the name of a trait, phenotype or covariate field either read from the pedigree data file or created by a `function` statement. If no valid `trait` parameters are listed, then all trait fields are used. Note that this can lead to long runs for highly multivariate data, and that the test for homogeneity among subtypes considers all specified traits jointly.

2. This parameter is used to specify the weight to be used to compute the correlations. The value of the `interclass_weight` or `intraclass_weight` parameter should be one of those from the following table. If no value has been selected by the user, or an invalid value has been selected, the default **pair_wise** will be used. For the **pair_wise** weighting scheme, the contributions to the sums of squares and cross-products for a given pedigree is proportional to the number of pairs of a main type or subtype in the pedigree; for the uniform weighting scheme, the contributions from each pedigree have the same weights, regardless of the number of pairs of the designated type in the pedigree.

   In the following table, p_j denotes the number of pairs of a given type in the jth pedigree. For example, in the case of brother-sister correlations, p_j would denote the total number of brother-sister pairs in the jth pedigree.

Parameter Value	Description	Calculation
pair_wise `---or---` **pair**	Each pair of the given type in the jth pedigree has the same weight regardless of p_j.	$w_j = 1$
uniform	Each pair has weight inversely proportional to the number of such pairs, p_j, in the j^{th} pedigree.	$w_i = \frac{1}{p_j}$, if $p_j \geq 1$,
reduced_group `---or---` **reduced**	Each pair has weight inversely proportional to the number of such pairs, p_j, in the j^{th} pedigree but pedigrees that contain only one pair of the type are excluded.	$w_j = \begin{matrix} \frac{1}{p_j} & \text{, if } p_j > 1 \\ 0 & \text{, if } p_j = 1 \end{matrix}$

3. The `type` parameter is used to specify whether to calculate correlations for subtypes only, or for main relative types only, or for both main relative types and subtypes. If the value of `type` is set to **subtypes**, then correlations of subtypes will be computed. If the value of `type` is set to **maintypes**, then correlations of main types will be computed. If the value of `type` is set to **both**, then both correlations of subtypes and main types will be computed. The default value is **subtypes**.

4. The `standard_error` parameter is used to specify whether to calculate asymptotic standard errors of correlations. By default, any standard error for which a required

correlation is nonestimable is calculated by setting the value of that required correlation to a value of 0, and appears within [ ] in the output. This usually overestimates the standard error. The optional `conservative` attribute specifies that if any required correlation is nonestimable, then that standard error is not calculated. The default value for `standard_error` is **true**.

5. If the value of `sex_name` is set to **false**, then non-sex specific names will be printed in output tables. The default value is **true**.

6. The `generation_limit` is the largest number of steps between the pair of individuals and their closest common ancestor. For example, a `generation_limit` value of one would include only parent offspring, sibling and half sibling pair types. A `generation_limit` value of 2 would include all first-and second-degree relationships, cousins and half avuncular pairs.

7. The `homogeneity_test` parameter is used to specify calculation of chi-squares and p-values to test for homogeneity of subtypes within main types. The values of `interclass_weight` and `intraclass_weight` are use to calculate all required correlations. The default value of `homogeneity_test` is **false**.

8. The `var_cov` parameter block is used to specify options to print variance-covariance matrices of subsets of the correlations. The `single` attribute is used to print all matrices for all traits, one trait at a time, and the `joint` attribute is used to print a joint matrix for two traits. If no `attribute` is specified, then `single` is used as the default. The traits are specified in the `var_cov` parameter block as `trait` parameters.

4.3.3 The `var_cov` Sub-Block

The following lists all parameters that may occur in a `var_cov` sub-block.

`parameter [, attribute]`	**Explanation**
`trait`	Names a trait, phenotype or covariate for which a variance-covariance matrix is to be printed. Value Range: Character string. Default Value: None Required: No Applicable Notes: 1
`correlation`	Specifies a set of correlations between two relative types for a variance-covariance matrix. Value Range: Any of the entries listed in the codes table below. Default Value: None Required: No Applicable Notes: 2

Notes

1. The value of a `trait` parameter should be set to the name from a `trait` parameter that is used in the `fcor` block. If no valid `trait` parameters are listed, then all trait fields used in the `fcor` block are used.

2. The value of `correlation` should be set to one of following codes or names, and can be repeated.

Main Types		**Subtypes**	
Code	**Name**	**Code**	**Name**
0	**self**	**m**	male-self
		f	female-self
1	**mother:father**	**m,f**	mother:father
10	**parent:offspring**	**mm**	father:son
		fm	mother:son
		mf	father:daughter
		ff	mother:daughter
11	**sibling**	**m,m**	brother:brother
		f,m	sister:brother
		f,f	sister:sister
11h	**half-sibling**	**m,m,m**	paternal-half-brother:half-brother
		f,m,m	paternal-half-sister:half-brother
		f,m,f	paternal-half-sister:half-sister
		m,f,m	maternal-half-brother:half-brother
		f,f,m	maternal-half-sister:half-brother
		f,f,f	maternal-half-sister:half-sister
20	**grandparental**	**mmm**	grandfather-through-father:grandson
		fmm	grandmother-through-father:grandson
		mfm	grandfather-through-mother:grandson
		ffm	grandmother-through-mother:grandson
		mmf	grandfather-through-father:granddaughter
		fmf	grandmother-through-father:granddaughter
		mff	grandfather-through-mother:granddaughter
		fff	grandmother-through-mother:granddaughter
21	**avuncular**	**m,mm**	uncle-through-father:nephew
		f,mm	aunt-through-father:nephew
		m,fm	uncle-through-mother:nephew
		f,fm	aunt-through-mother:nephew
		m,mf	uncle-through-father:niece
		f,mf	aunt-through-father:niece
		m,ff	uncle-through-mother:niece
		f,ff	aunt-through-mother:niece
22	**cousin**	**mm,mm**	male-cousin-through-father:male-cousin-through-father
		mf,mm	male-cousin-through-mother:male-cousin-through-father
		mf,fm	male-cousin-through-mother:male-cousin-through-mother
		fm,mm	female-cousin-through-father:male-cousin-through-father

Main Types		Subtypes	
Code	Name	Code	Name
		fm,fm	female-cousin-through-father:male-cousin-through-mother
		ff,mm	female-cousin-through-mother:male-cousin-through-father
		ff,fm	female-cousin-through-mother:male-cousin-through-mother
		fm,mf	female-cousin-through-father:female-cousin-through-father
		ff,mf	female-cousin-through-mother:female-cousin-through-father
		ff,ff	female-cousin-through-mother:female-cousin-through-mother

4.3.4 FCOR Examples

The following are all valid `fcor` statements:

```
fcor
{
}

fcor
{
   interclass_weight=uniform
}

fcor, out=test
{
   trait=TRAIT1
   trait=TRAIT2
   trait=TRAIT3
   interclass_weight=uniform
   intraclass_weight=pair
   standard_error=true
   sex_name=false
   type=maintypes
   homogeneity_test=true
   generation_limit=3

   var_cov, single { # This will calculate separate variance-
                     # covariance matrices for TRAIT1 and TRAIT2
      trait=TRAIT1   # parent:offspring type correlations.
      trait=TRAIT2
      correlation=parent:offspring
   }
   var_cov, joint { # This will calculate the joint variance-
                    # covariance matrices for TRAIT1 and
      trait=TRAIT1  # TRAIT2, TRAIT1 and TRAIT3, and TRAIT2 and
      trait=TRAIT2  # TRAIT3 father:son and mother:son
      trait=TRAIT3  # correlations.
      correlation=mm # father:son
      correlation=fm # mother:son
   }
}
```

4.4 Program Execution

FCOR is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running FCOR from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>fcor
S.A.G.E. v5.x -- FCOR
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: fcor <parameters> <pedigree>
Command line parameters:
  parameters   - parameter file
  pedigree     - pedigree data file
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of FCOR may look like the following:

```
>fcor data.par data.ped
S.A.G.E. v5.x -- FCOR
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading Parameter File.....................done.
Reading pedigree file.....................
              from data.ped..............done.
Sorting pedigrees.........................done.
  No analyses specified.
Performing FCOR default analysis...
Computing subtypes correlations...........done.
Computing standard errors.................done.
Writing output files......................done.
Analysis complete!
```

4.5 Program Output

FCOR produces six types of output files that contain results and diagnostic information:

File Name	Description
fcor.inf	Contains informational diagnostic messages, warnings and program errors. No calculation results are stored in this file.
fcor.sub	Contains tables of correlations and standard errors with used pair counts and equivalent pair counts for each pair of traits for each subtype of relative up to 2nd generation (by default) or the generation specified by `generation_limit`. Generated when `pair_type` value is **subtypes** or **both**.
fcor.main	Contains tables of correlations and standard errors, with used pair counts and equivalent pair counts, for each pair of traits for each main type of relative up to 2nd generation (by default) or up to the generation specified by `generation_limit`. Generated when `type` value is **maintypes** or **both** .
fcor.htest	Contains chi-square values and p-values. Generated when `homogeneity_test` value is true.
fcor.alt	Contains tables of correlations and standard errors, with used pair counts and equivalent pair counts in the alternate tabular form. Generated when `type` has an attribute `tabular`.
fcor.pair	Contains tables of the smallest number of pairs used to calculate any of the required correlations for each standard error. Generated when `standard_error` has the attribute `pairs`.
fcor.cov	Contains the variance-covariance matrix of correlation estimates. Generated when there is a `var_cov` sub-block within the FCOR block.

4.5.1 Information Output File

The FCOR information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read from the pedigree data file, are included with all programs in S.A.G.E. and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "fcor.inf" should be checked for errors and diagnostic information after each run of the program.

4.5.2 Correlations and Standard Errors: Subtypes & Main Types

The FCOR subtypes and main types output file prints the tables of correlations and standard errors for interclass and intraclass relative types, within two generations by default. An additional output file contains the alternate tabular form for the tables.

4.5.3 Smallest Pair Numbers

The FCOR pair number output prints the tables indicating, for each standard error, the smallest number of pairs used to calculate any of the required correlations.

4.6 Example Output Files

4.6.1 Correlations and Standard Errors: Subtypes

Here is a typical example of an FCOR subtype output file for the father-son relationship when the `sex_name` option is set equal to **true**:

```
================================================================================
  Tables of Correlations +/- Asymptotic Standard Errors for Subtypes
================================================================================

    Number of pedigrees : 207
    Number of traits    : 2

    Weight method used for interclass : Equal Weight to Pairs
    Weight method used for intraclass : Uniform Weight to Pedigrees

    Legend :
      ******  : Value is not estimable.
      &&&&&&  : Value is greater than or equal to 100000.
      @@@@@@  : Standard error is greater than or equal to 10.0.
      ######  : Equivalent pair count is greater than or equal to 10000.
      [StdErr]: Calculated by setting the value of an inestimable required
                correlation to a value of 0.
================================================================================
.
.
.
Relationship Type : Row:Column
                    father:son
Pairs Found       : 241
-------------------------------------------------------
                    TRAIT1               TRAIT2
  INTERCLASS ----------------------------------------
               Count  Correlation   Count  Correlation
              EqvCnt  +/-  StdErr  EqvCnt  +/-  StdErr
-------------------------------------------------------
   TRAIT1      241        -0.0011   241        -0.0265
               180.3  +/- [0.0747]  440.0  +/- [0.0477]
-------------------------------------------------------
   TRAIT2      241        -0.0494   241        -0.0585
               440.8  +/- [0.0476]  168.4  +/- [0.0770]
-------------------------------------------------------
.
.
.
```

4.6.2 Correlations and Standard Errors: Main Types

Here is a typical example of the FCOR main type output file for the parent-offspring relationship:

```
================================================================================
  Tables of Correlations +/- Asymptotic Standard Errors for Maintypes
================================================================================

    Number of pedigrees : 207
    Number of traits    : 2

    Weight method used for interclass : Equal Weight to Pairs
    Weight method used for intraclass : Uniform Weight to Pedigrees

    Legend :
      ******  : Value is not estimable.
      &&&&&&  : Value is greater than or equal to 100000.
      @@@@@@  : Standard error is greater than or equal to 10.0.
      ######  : Equivalent pair count is greater than or equal to 10000.
      [StdErr]: Calculated by setting an inestimable constituent correlation
                to a value of 0.
================================================================================
.
.
.
Relationship Type : parent:offspring
  Subtypes Pooled : father : son
                    mother : son
                    father : daughter
                    mother : daughter

Total Pairs Found : 913
--------------------------------------------------------
                    TRAIT1                TRAIT2
 INTERCLASS  ----------------------------------------
               Count  Correlation   Count  Correlation
              EqvCnt  +/-  StdErr  EqvCnt  +/-  StdErr
--------------------------------------------------------
   TRAIT1      913         0.1825   913         0.1302
               620.7  +/- [0.0388]  609.3  +/- [0.0399]
--------------------------------------------------------
   TRAIT2      913         0.1136   913         0.0730]
               650.6  +/- [0.0387]  638.9  +/- [0.0394]
--------------------------------------------------------
.
.
.
```

4.6.3 Output File of the Alternate Tabular Form

Here is a typical example of the alternate tabular form output tables:

```
Relationship Type : father:son

Pairs Found       : 241

-----------------------------------------------------
                  Count  Correlation  EqvCnt  StdError
-----------------------------------------------------
 TRAIT1 - TRAIT1    241   0.0496812    190.9  0.072387
 TRAIT1 - TRAIT2    241   0.0443138    168.6  0.077100
 TRAIT2 - TRAIT1    241   0.0258354    190.8  0.072536
 TRAIT2 - TRAIT2    241   0.0295229    168.6  0.077172
-----------------------------------------------------
```

4.6.4 Output File of the Smallest Pair Numbers

Here is a typical example of the FCOR pair numbers output tables:

```
============================================================================
  Tables of the Smallest Number of Pairs Used in
    Calculating Required Correlations for Subtypes
============================================================================

    Number of pedigrees : 207
    Number of traits    : 2

    Weight method used for interclass : Equal Weight to Pairs
    Weight method used for intraclass : Uniform Weight to Pedigrees
    [      ] : Excluded the number of pairs for inestimable required
               correlations.
============================================================================
.
.
.
Relationship Type : Row:Column
                    father:son
Pairs Found       : 241
-------------------------------------
 INTERCLASS     TRAIT1      TRAIT2
-------------------------------------
   TRAIT1           178         178
-------------------------------------
   TRAIT2           178         178
-------------------------------------
.
.
.
```

4.6.5 Homogeneity Test Results Output File

The FCOR homogeneity test output file for main types of relatives has chi-square values and p-values. Here is a typical example of the FCOR homogeneity test output file for the parent-offspring relationship when the `homogeneity_test` option is **true**:

```
Notes :
    ****** : Value is not estimable.
    [    ] : Calculated by setting nonestimable required correlations
             to a value of 0.

================================================================================
Relationship Type : parent:offspring
  Subtypes Pooled : father:son
                    mother:son
                    father:daughter
                    mother:daughter

Total Pairs Found : 420

Chi-Square = [1.82247] with 3 degree(s) of freedom
P-Value    = [0.610058]
--------------------------------------------------------------------------------
```

4.6.6 Variance-Covariance Matrix Output File

Here is a typical example of the variance-covariance matrices for TRAIT1 and TRAIT2 parent-offspring correlations:

```
==================================================
  Variance-Covariance Matrix for Correlations of
     PARENT:OFFSPRING
       with
     PARENT:OFFSPRING
     trait(s) : TRAIT1 TRAIT2  SINGLY
  **** : Value is not estimable.
  [  ] : Calculated by setting an inestimable
         required correlation to a value of 0.
==================================================

  Legend :
      [R1] PARENT:OFFSPRING  TRAIT1:TRAIT1
      [C1] PARENT:OFFSPRING  TRAIT1:TRAIT1
  ---------------------
     \         [C1]
  ---------------------
    [R1]    0.0052217
  ---------------------
  The Smallest Number of Pairs Used in
  Calculating Required Correlations
  ---------------------
     \         [C1]
  ---------------------
    [R1]           86
  ---------------------
  Legend :
      [R1] PARENT:OFFSPRING  TRAIT2:TRAIT2
      [C1] PARENT:OFFSPRING  TRAIT2:TRAIT2
  ---------------------
     \         [C1]
  ---------------------
    [R1]  [ 0.0062607]
  ---------------------
  The Smallest Number of Pairs Used in
  Calculating Required Correlations
  ---------------------
     \         [C1]
  ---------------------
    [R1]           86
  ---------------------
```

Here is another typical example of the joint variance-covariance matrices for TRAIT1, TRAIT2, and TRAIT3 father:son and mother:son correlations.

```
================================================
  Variance-Covariance Matrix for Correlations of
     FATHER:SON
       with
     MOTHER:SON
     trait(s) : TRAIT1 TRAIT2 TRAIT3  JOINTLY
  **** : Value is not estimable.
  [  ] : Calculated by setting an inestimable
         required correlation to a value of 0.
================================================

  Legend :
      [R1] FATHER:SON - TRAIT1:TRAIT1
      [R2] FATHER:SON  TRAIT1:TRAIT2
      [R3] FATHER:SON  TRAIT2:TRAIT1
      [R4] FATHER:SON  TRAIT2:TRAIT2
      [C1] MOTHER:SON - TRAIT1:TRAIT1
      [C2] MOTHER:SON - TRAIT1:TRAIT2
      [C3] MOTHER:SON  TRAIT2:TRAIT1
      [C4] MOTHER:SON  TRAIT2:TRAIT2
  ------------------------------------------------------------
     \         [C1]         [C2]         [C3]         [C4]
  ------------------------------------------------------------
    [R1]   -0.0004526   -0.0000822   -0.0001792   -0.0000288
    [R2]   -0.0000854   -0.0004590    0.0000230   -0.0001697
    [R3]    0.0003632    0.0001635    0.0000327    0.0001361
    [R4]    0.0000577    0.0002804    0.0001000    0.0001008
  ------------------------------------------------------------
  The Smallest Number of Pairs Used in
  Calculating Required Correlations
  ------------------------------------------------------------
     \         [C1]         [C2]         [C3]         [C4]
  ------------------------------------------------------------
    [R1]          178          178          178          178
    [R2]          178          178          178          178
    [R3]          178          178          178          178
    [R4]          178          178          178          178
  ------------------------------------------------------------
  Legend :
      [R1] FATHER:SON - TRAIT1:TRAIT1
      [R2] FATHER:SON - TRAIT1:TRAIT3
      [R3] FATHER:SON  TRAIT3:TRAIT1
      [R4] FATHER:SON  TRAIT3:TRAIT3
      [C1] MOTHER:SON - TRAIT1:TRAIT1
      [C2] MOTHER:SON - TRAIT1:TRAIT3
      [C3] MOTHER:SON  TRAIT3:TRAIT1
      [C4] MOTHER:SON  TRAIT3:TRAIT3
  ------------------------------------------------------------
    \          [C1]         [C2]         [C3]         [C4]
  ------------------------------------------------------------
    [R1]   -0.0004526    0.0000009   -0.0000811   -0.0000411
    [R2]    0.0000646   -0.0005089    0.0000746   -0.0000916
    [R3]    0.0003006    0.0001168   -0.0003247    0.0002001
    [R4]   -0.0000465    0.0003295   -0.0000248   -0.0002056
  ------------------------------------------------------------
```

```
The Smallest Number of Pairs Used in
Calculating Required Correlations
------------------------------------------------------------
   \         [C1]         [C2]         [C3]         [C4]
------------------------------------------------------------
  [R1]          178          178          178          178
  [R2]          178          178          178          178
  [R3]          178          178          178          178
  [R4]          178          178          178          178
------------------------------------------------------------
Legend :
    [R1] FATHER:SON  TRAIT2:TRAIT2
    [R2] FATHER:SON  TRAIT2:TRAIT3
    [R3] FATHER:SON  TRAIT3:TRAIT2
    [R4] FATHER:SON  TRAIT3:TRAIT3
    [C1] MOTHER:SON  TRAIT2:TRAIT2
    [C2] MOTHER:SON  TRAIT2:TRAIT3
    [C3] MOTHER:SON  TRAIT3:TRAIT2
    [C4] MOTHER:SON  TRAIT3:TRAIT3
------------------------------------------------------------
   \         [C1]         [C2]         [C3]         [C4]
------------------------------------------------------------
  [R1]    0.0001008   -0.0003228   -0.0006320    0.0000243
  [R2]    0.0001108   -0.0000552    0.0005476   -0.0005515
  [R3]    0.0003562    0.0005819   -0.0003420   -0.0002770
  [R4]    0.0000523    0.0005094   -0.0000148   -0.0002056
------------------------------------------------------------
       The Smallest Number of Pairs Used in
         Calculating any Required Correlations
------------------------------------------------------------
   \         [C1]         [C2]         [C3]         [C4]
------------------------------------------------------------
  [R1]          178          178          178          178
  [R2]          178          178          178          178
  [R3]          178          178          178          178
  [R4]          178          178          178          178
------------------------------------------------------------
```

Chapter 5

SEGREG

5.1 Introduction

SEGREG is a very general program that can be used for, among other things, commingling analysis, segregation analysis and to produce penetrance files for model-based linkage analysis (for use in the programs LODLINK and MLOD). The most significant improvements over the programs REGC, REGD and REGTL of the previous versions of S.A.G.E. are as follows:

1. It is no longer necessary to provide initial parameter estimates (but these can be provided if desired).
2. It is no longer necessary (or possible) to specify parameters that control the maximizing process.
3. Several related analyses can be automatically performed in a single run.
4. When a transformation of the data is performed, all location parameter estimates refer to the data on their original scale of measurement - but parameter estimates of dispersion still refer to the transformed variable.
5. All covariates are initially centered, and the centering (average) values are given as part of the output.

5.2 Limitations

As with most S.A.G.E. programs, SEGREG cannot currently be used in the presence of pedigree loops.

Further, if the sample size is small relative to the number of parameters being estimated, the likelihood may have multiple maxima. There is no guarantee that in such a situation the maximum found and reported by the program is also the global maximum. Also, situations can occur in which it is not numerically possible to calculate the variance-covariance matrix of the estimates.

5.3 Theory

The segregation of a possible major locus is allowed for by letting one or more parameters depend on an unobserved (latent) qualitative factor u = AA, AB or BB. Following Go et al. (1978), we call u an individual's *type*. In this context, type is best defined in terms of the expected distribution of an individual's offspring. Two individuals have the same type if and only if the expected phenotypic distributions of their offspring by a mate of a given type are identical, and this is true for every type of mate. The same concept, but not with this definition, was denoted *ousiotype* by Cannings et al. (1978). Genotypes are the special case of types, or ousiotypes, that transmit to offspring in Mendelian fashion.

Thus we use the term *type* to allow for many kinds of discrete transmission, whether Mendelian or not. When there is no transmission from one generation to the next, the model can include the existence of only one type as defined above. In this situation, it will nevertheless be convenient to refer to several types, each with its own phenotypic distribution, but it must be understood that the model then essentially allows for only a single type, the corresponding phenotypic distribution being a mixture distribution. The incorporation of types introduces two sets of parameters, type frequencies[1] and transmission[2] parameters. The population frequencies of the types are designated ψ_u, for u = AA, AB, BB, and satisfy the condition:

$$\sum_u \psi_u = 1.$$

If the type frequencies are in Hardy-Weinberg equilibrium proportions, then they are defined in terms of q_A = frequency of (component allele) A. Thus:
$\psi_{AA} = q_A^2$; $\psi_{AB} = 2q_A(1 - q_A)$; $\psi_{BB} = (1 - q_A)^2$.

Each transmission parameter τ_u is the probability that a parent of type u transmits allele (more generally, *component*) A to offspring, for u = AA, AB, BB. Mendelian transmission corresponds to the case in which τ_{AA} = 1, τ_{AB} = 0.5, and τ_{BB} = 0. These parameters give rise to transition[3] probabilities. The transition probability $Pr(u|u_F, u_M)$ is the probability that parents of types u_F and u_M produce offspring of type u. Transition probabilities are determined by the transmission probabilities as follows:

$$\begin{aligned}
Pr(AA|u_F, u_M) &= \tau_{u_F}\tau_{u_M}, \\
Pr(AB|u_F, u_M) &= \tau_{u_F}(1 - \tau_{u_M}) + \tau_{u_M}(1 - \tau_{u_F}), \\
Pr(BB|u_F, u_M) &= (1 - \tau_{u_F})(1 - \tau_{u_M}).
\end{aligned}$$

However, in the case that there is homogeneity of the phenotypic distributions between generations and no parent-offspring transmission of type, we define Pr(u|u_F,u_M)=τ_u, for u = AA, AB, BB. In order to have homogeneity across generations when there is parent-offspring transmission of type,

1. the type frequencies must be in Hardy-Weinberg equilibrium proportions, and

[1] We use the word *frequencies* in the sense used by geneticists, i.e., *relative frequencies* that sum to 1.
[2] SEGREG uses the terms *transmission probability* and *transition probability* as defined by Elston and Stewart (1971).
[3] ditto

2. τ_{AB} must be a function of τ_{AA}, τ_{BB} and the allele frequency q_A (Demenais and Elston, 1981).

Details of the pedigree likelihoods that are calculated, on the assumption of random mating, are given below. It should be noted that singletons (unrelated individuals) may be included in the data. Although SEGREG counts and treats them separately for convenience, they are in fact simply one-person pedigrees and, as such, require no special treatment in the model. However, note that these singletons are not considered to be founders. Estimation is performed by maximum likelihood and standard errors are obtained by numerical double differentiation of the log likelihood surface. The output contains the overall ln(likelihood), -2ln(likelihood) and Akaike's *A* information criterion (AIC)[4] for each of the models that has been maximized in a run. When the model consists of two or three types, a table is produced indicating the respective likelihood ratio statistic for each type.

Transmission models are compared and p-values quoted according to the asymptotic distribution of the likelihood ratio as shown in the table below (Self and Liang, 1987).

Distributions of the Segregation Analysis Test Statistic Used by SEGREG

	Homo-No-Tran	**Homo-Mendelian**	**Homo-General**	**Tau-AB-Free**
Homo-Mendelian	—			
Homo-General	χ^2_2 (2t, 3t-hwe)	$\left(\frac{1}{4}\right)+\left(\frac{1}{2}\right)\chi^2_1+\left(\frac{1}{4}\right)\chi^2_2$ (2t, 3t-hwe)		
Tau-AB-Free	—	$\left(\frac{1}{2}\right)+\left(\frac{1}{2}\right)\chi^2_1$ (2t, 3t-hwe)	—	
General	χ^2_3 (2t, 3t-hwe, 3t-nhwe)	$\left(\frac{1}{4}\right)\chi^2_1+\left(\frac{1}{2}\right)\chi^2_2+\left(\frac{1}{4}\right)\chi^2_3$ (2t, 3t-hwe, 3t-nhwe)	χ^2_1 (2t, 3t-hwe)	χ^2_2 (2t, 3t-hwe, 3t-nhwe)

Legend		
	—	Not Applicable
	2t	Two-type models
	3t-hwe	Three-type models with HWE
	3t-nhwe	Three-type models without HWE

[4]Contrary to popular belief, the acronym AIC stands for *A Information Criterion*, and not *Akaike's Information Criterion*.

5.3.1 Segregation Models

Certain aspects of the models available in SEGREG are common to all traits and models, and are described here. Later sections describe the aspects that are specific to regressive models for continuous traits, regressive multivariate logistic models for binary traits, the finite polygenic mixed model, and models for binary traits with variable age of onset.

5.3.1.1 Single Ascertainment and/or Conditioning on a Subset

Instead of being sampled at random, a pedigree may be included in the analysis because one or more members of the pedigree have particular trait values or are in a certain *sampling frame*[5]. It may be desired to condition the likelihood on the phenotypes of these individuals or, more generally, on the phenotypes and/or structure of any prespecified subset C of the pedigree. This *conditioned subset* may be

1. the set of founders (members of the pedigree whose parents are not included in the pedigree),
2. the set of pedigree members in the pedigree proband sampling frame, or
3. the union of these two sets.

Currently, no model is assumed for the ascertainment and, for results to be correct, the observed pedigree must contain all members of the pedigree proband sampling frame. This subsumes both simplex and multiplex single ascertainment (see Elston and Bonney, 1986) as a special case. See Ginsberg et al (2003) for a discussion of what is meant by "pedigree", "correct results" and "pedigree proband sampling frame".

If no conditioned subset is indicated for a particular pedigree (either explicitly as a user-specified set or implicitly as the founders), random sampling is assumed for that pedigree. In general, the likelihood for a randomly sampled pedigree (L) is divided by a correction L_C, defined in one of three possible ways.

1. **Random Sampling**

   In this case, no correction is necessary, so C is empty and we define L_C = 1.

2. **Conditioning on Actual Phenotypes**

   In this case, the likelihood is conditioned on the available phenotype of each member of the conditioned subset. The correction L_C is then taken to be L computed as though all individuals not in C are missing.

3. **Conditioning on Phenotypes Being Above or Below a Threshold Value**

   In this case, the likelihood is conditioned, for each member of the conditioned subset for whom a phenotype is available, on that member's phenotype being at least as large as a threshold T_U, or at most as large as a threshold T_L.

[5]The pedigree sampling frame can include pedigree members for whom the trait value is missing. (See Ginsberg et al (2003) for further discussion.)

5.3.1.2 Type Probabilities and Penetrance Functions

Given a model with established parameter values, we can estimate the probability of each possible type for every individual conditional on all the sample data. We define the following terms:

- L(•) is a likelihood
- *S* is the set of all sampled data in the pedigree,
- t_i is the analysis trait of individual *i*,
- u_i is the type of individual *i*,
- u_{i_M} is the type of individual *i*'s mother, and
- u_{i_F} is the type of individual *i*'s father.

Then the posterior probability for a given individual is computed (using maximum likelihood estimates of unknown parameters) as:

$$L(u_i|S) = \frac{L(u_i,\ S)}{L(S)}.$$

Note that the denominator is the likelihood L computed for the whole pedigree to which *i* belongs.

If u_i is a genotype, SEGREG can prepare files of penetrance functions that can be used as input into LODLINK and MLOD using maximum likelihood estimates of all unknown parameters. These are of the form $\Pr(t_i|u_i)$.

5.3.2 Regressive Models for Continuous Traits

Regressive models (Bonney, 1984; 1998) are those models in which distributions over pedigrees are specified by conditioning each individual's trait value on those of antecedent individuals. For continuous trait they assume (possibly after transformation) multivariate normality across pedigree members of the underlying individual residuals from the type means. Two classes of regressive models for continuous traits are implemented in SEGREG. Class A models assume that siblings are dependent only because of common parentage, while class D models assume that the sibling correlations are equal, but not necessarily due to common parentage alone. For a continuous trait, SEGREG assumes a model that is a close approximation to multivariate normal for the underlying individual residuals. The approximation used is a generalization of approximation 6 in Demenais et al (1990).

The following non-zero residual correlations are allowed in all the models: ρ_{FM} for father-mother (spouse), ρ_{MO} for mother-offspring, ρ_{FO} for father-offspring, and ρ_{SS} for the correlation between any two siblings (for a class D model). A class A model also includes, indirectly, a sibling correlation ρ_{SS} that satisfies the condition

$$\rho_{SS}=\frac{\rho_{MO}^2+\rho_{FO}^2-2\rho_{FM}\rho_{MO}\rho_{FO}}{1-\rho_{FM}^2}.$$

The residual correlations between half siblings are assumed to be zero, conditional on the common parent. Missing values are handled according to the formulas in Bonney (1984, 1998).

In the correlation structure indicated above, the means and variances of the underlying normal distribution can be dependent on covariates. All covariates are centered prior to inclusion in the likelihood.

When types are incorporated into the model, the correlation parameters (ρs) are the correlations of the residual multivariate normal distribution. Thus the inference of a major gene can be made allowing for the cumulative effect, assumed to be multivariate normally distributed for the transformed trait, of various factors (such as polygenes, cultural, and other environmental factors) that are not separately distinguished.

5.3.2.1 Composite Trait

The trait, or phenotype, to be analyzed may be a single variate, the *main phenotype* (y^* = y) or a linear function of the main phenotype (with coefficient 1) and p covariates (with coefficients κ_i):

$$y^* = y + \kappa_1 x_1 + \kappa_2 x_2 + ... + \kappa_{p_\kappa} x_p,$$

where the parameters κ_i may be estimated.

5.3.2.2 Transformation of the Phenotype

The phenotype y^*, however composed, may be transformed by one of two transformations. For commingling analysis and segregation analysis, the first (Cox and Box) transformation is recommended.

The first possible transformation is:

$$t = h(y^*) = \begin{cases} \frac{(y^*+\lambda_1)^{\lambda_1}-1}{\lambda_1 (y^*_{G1})^{(\lambda_1-1)}} & if \lambda_1 \neq 0, \\ y^*_{G1} \ln(y^*+\lambda_2) & if \lambda_1 = 0 \end{cases}$$

where

$$y^*_{G1} = \left[\prod_{i=1}^{N} (y_i^*+\lambda_2)\right]^{\frac{1}{N}},$$

and N = number of individuals in the full data set (possibly including more than one pedigree) with complete phenotype and covariate values (nothing missing). This is the standardized Box and Cox (1964) transformation with power parameter $\lambda_1$and shift parameter λ_2.

The second possible transformation is:

$$t = h(y*) = \begin{cases} \frac{sign(y*+\lambda_2)\left[(|y*+\lambda_2|+1)^{\lambda_1}-1\right]}{\lambda_1 (y^*_{G2})^{(\lambda_1-1)}} & if\ \lambda_1 \neq 0 \\ y^*_{G2} sign(y*+\lambda_2)\ln(|y*+\lambda_2|+1) & if \lambda_1 = 0 \end{cases}$$

where

$$y^*_{G2} = \left[\prod_{i=1}^{N}(|y^*_i+\lambda_2|+1)\right]^{\frac{1}{N}}.$$

This is the standardized generalized modulus power transformation (George and Elston, 1988) with power parameter λ_1 and shift parameter λ_2.

We call the transformed phenotype *t* the *analysis trait*. When a transformation is applied it is applied to *both sides* (Carroll and Ruppert, 1984), so that all location parameters are median unbiased on the original scale of measurement.

5.3.2.3 Likelihood for a Randomly Sampled Pedigree

Let the pedigree contain n individuals (i = 1, ..., n) on each of whom we observe a value of the analysis trait. An individual is considered missing if the value of any variate for that individual, required for calculating the likelihood, is unknown. For individual i, let

t_i = analysis trait value of i

x_{ij} = j-th covariate value of i

u_i = type of i

S_i = spouse of i

M_i = mother of i

F_i = father of i

B_{ij} = j'th observed elder sibling of i

n_{iB} = number of observed elder siblings of i.

We let the expected value of *t* conditional on type *u* be

$$\theta_u(i) = h(\beta_u + \xi_1 x_{i1} + \xi_2 x_{i2} + ... + \xi_{p_\xi} x_{ip_\xi})$$

and the variance of *t* conditional on type *u* be

$$\eta_u^2(i) = \sigma_u^2 + \varsigma_1 x_{i1} + \varsigma_2 x_{i2} + ... + \varsigma_{p_\varsigma} x_{ip_\varsigma}$$

Because the expected value of *t* conditional on type *u* undergoes the same transformation as is used to produce *t* ("transformation of both sides", see Carroll & Ruppert, 1984), the estimates of parameters in this conditional expectation are median unbiased on the same scale of measurement as the original untransformed data. However, the residual variance that is calculated, and all the covariate coefficients pertaining to it, are on the scale of the analysis trait. Further general quantities that apply to regressive models are defined as follows:

$$\alpha_{iS} = \begin{cases} \rho_{FM} & if\ specific\ spouse\ of\ i\ is\ observed, \\ 0 & otherwise, \end{cases}$$

$$\alpha_{iM} = \begin{cases} \frac{\rho_{MO} - \rho_{FO}\rho_{FM}}{1-\rho_{FM}^2} & if\ both\ parents\ of\ i\ are\ observed, \\ \rho_{MO} & if\ mother,\ but\ not\ father,\ of\ i\ is\ observed, \\ 0 & if\ mother\ of\ i\ is\ not\ observed, \end{cases}$$

$$\alpha_{iF} = \begin{cases} \frac{\rho_{FO} - \rho_{MO}\rho_{FM}}{1-\rho_{FM}^2} & if\ both\ parents\ of\ i\ are\ observed, \\ \rho_{FO} & if\ father,\ but\ not\ mother,\ of\ i\ is\ observed, \\ 0 & if\ father\ of\ i\ is\ not\ observed, \end{cases}$$

$$\delta_i = \alpha_{iM}\rho_{MO} + \alpha_{iF}\rho_{FO} = \begin{cases} \rho_{SS} & if\ both\ parents\ of\ i\ are\ observed, \\ \rho_{MO}^2 & if\ mother,\ but\ not\ father,\ of\ i\ is\ observed, \\ \rho_{FO}^2 & if\ father,\ but\ not\ mother,\ of\ i\ is\ observed, \\ 0 & if\ neither\ parent\ of\ i\ is\ observed, \end{cases}$$

$$\phi(z_i, w_i) = \frac{1}{\sqrt{2\pi w_i}} \exp[-z_i^2/(2w_i)],$$

where the arguments z_i and w_i are defined differently for each of the model classes. For a class A model, the arguments of the normal density function are defined in SEGREG as

$$z_i = t_i - \theta_u(i) - b_{iS}V_{iS_i}(t_{S_i} - \theta_u(S_i)) - b_{iM}V_{iM_i}(t_{M_i} - \theta_u(M_i)) - b_{iF}V_{iF_i}(t_{F_i} - \theta_u(F_i))$$

and

$$w_i = \eta_u^2(i)(1 - b_{iS}\rho_{FM} - b_{iM}\rho_{MO} - b_{iF}\rho_{FO}),$$

where

$$V_{ij} = \eta_u(i)/\eta_u(j)$$

$$b_{iS} = \alpha_{iS},$$

$$b_{iM} = \alpha_{iM}\left(\frac{1-\rho_{SS}}{1-\rho_{SS}+n_{iB}(\rho_{SS}-\delta_i)}\right)$$

$$b_{iF} = \alpha_{iF}\left(\frac{1-\rho_{SS}}{1-\rho_{SS}+n_{iB}(\rho_{SS}-\delta_i)}\right),$$

with

$$\rho_{SS} = \frac{\rho_{MO}^2 + \rho_{FO}^2 - 2\rho_{MO}\rho_{FO}\rho_{FM}}{1-\rho_{FM}^2}.$$

For a class D model, the arguments of the normal density function are defined as:

$$\begin{aligned} z_i &= t_i - \theta_u(i) - b_{iS}V_{iS_i}(t_{S_i} - \theta_u(S_i)) - b_{iM}V_{iM_i}(t_{M_i} - \theta_u(M_i)) \\ &\quad -b_{iF}V_{iF_i}(t_{F_i} - \theta_u(F_i)) - b_{iB}\sum_{j=1}^{n_{iB}}\hat{V}_{iB_{ij}}(t_{B_{ij}} - \hat{\mu}_{B_{ij}}), \end{aligned}$$

and

$$w_i = \eta_u^2(i)(1 - b_{iS}\rho_{FM} - b_{iM}\rho_{MO} - b_{iF}\rho_{FO} - n_{iB}b_{iB}\rho_{SS}),$$

where

$$\begin{aligned} \hat{\mu}_j &= \sum_{u_j}\theta_u(j)f_{uj}/\sum_{u_j}f_{uj} \\ f_{uj} &= Pr(u_j|u_{F_j}, u_{M_j})exp\{-(t_j - \theta_u(j))^2/(2\eta_u^2(j))\}/\eta_u(j), \\ \hat{\sigma}_j^2 &= \sum_{u}f_{uj}\sigma_u^2/\sum_{u}f_{uj}, \\ \hat{V}_{ij} &= \sum_{u_j}f_{uj}/\sum_{u_j}f_{uj}\eta_u^2(j) \\ b_{iS} &= \alpha_{iS} \\ b_{iM} &= \alpha_{iM}\left(\frac{1-\rho_{SS}}{1-\rho_{SS}+n_{iB}(\rho_{SS}-\delta_i)},\right) \end{aligned}$$

To indicate all the potential variables in $\phi(z_i, w_i)$, except covariates, denote it

$$Pr(t_i|u_i, u_S, u_M, u_F, t_{S_i}, t_{M_i}, t_{F_i}, t_{B_{il}}, ..., t_{B_{in_{iB}}}).$$

(This quantity is a conditional phenotypic density function, sometimes referred to as a penetrance function.)

Using the components defined above, and letting

$$p_i(u_i, u_{M_i}, u_{F_i}) = \begin{cases} Pr(u_i|u_{F_i}, u_{M_i}) & if\ the\ parents\ of\ i\ are\ included\ in\ pedigree, \\ \psi_{u_i} & otherwise, \end{cases}$$

$$H_i(u_i, u_{S_i}, u_{M_i}, u_{F_i}, t_i, t_{S_i}, t_{M_i}, t_{F_i}, t_{B_{il}}, ..., t_{B_{in_{iB}}})$$

$$= \begin{cases} p_i(u_i, u_{M_i}, u_{F_i}) & if\ i\ missing, \\ p_i(u_i, u_{M_i}, u_{F_i}) Pr(t_i|u_i, u_{S_i}, u_{M_i}, u_{F_i}, t_{S_i}, t_{M_i}, t_{F_i}, t_{B_{i1}}, ..., t_{B_{in_{iB}}}) & otherwise \end{cases}$$

under random mating the likelihood for a randomly sampled pedigree is

$$L = \left[\sum_{u_1} ... \sum_{u_n} \prod_{i=1}^{n} H_i(u_i, u_{S_i}, u_{M_i}, u_{F_i}, t_i, t_{S_i}, t_{M_i}, t_{F_i}, t_{B_{il}}, ..., t_{B_{in_{iB}}}) \right].$$

5.3.2.4 Allowing for Ascertainment

Ascertainment is allowed for as indicated in 5.3.1.1. In order to condition on phenotypes being at least as large as T_U or at most as large as T_L, the correction L_C is taken to be the likelihood defined in 5.3.1.1 computed as though all individuals not in the prespecified subset C are missing, but with $Pr(t_i|.)$, for each individual i in C replaced by

$$\int_{T_U}^{\infty} Pr(t|.)dt = \Phi(-z_{iU}/\sqrt{w_i}), or \int_{-\infty}^{T_L} Pr(t|.)dt = \Phi(z_{iL}/\sqrt{w_i}),$$

where z_{iU} or z_{iL} is identical to z_i with $h(T_U)$ or $h(T_L)$, respectively, substituted for t_i. However, z_i is always left unchanged for any founders not included in the proband sampling frame.

5.3.3 Regressive Multivariate Logistic Models for Binary Traits

The multivariate logistic model for a binary trait was described by Karunaratne and Elston (1998) for nuclear family data. It is implemented in SEGREG for pedigree data by making the regressive model assumption that, conditional on the phenotype and major type of any individual who belongs to two nuclear families, the likelihoods for those two nuclear families are independent. In this model, unlike in Bonney's (1986) multiple logistic model, the marginal probability that any pedigree member has a particular phenotype is the same for all members who have the same values of any covariates in the model. This marginal probability, which we call susceptibility, is given by the cumulative logistic function

$$\gamma = \frac{e^{\theta(i)t_i}}{1 + e^{\theta(i)}},$$

where t_i, the analysis trait of the i-th individual, is 1 for an affected individual and 0 for an unaffected individual; and $\theta(i)$, the logit of the susceptibility for the i-th individual, can depend on both major type (u) and covariate values $x_{i1}, x_{i2}, ..., x_{ip}$:

$$\theta_u(i) = \beta_u + \xi_1 x_{i1} + ... + \xi p x_{ip}.$$

Composite traits and phenotype transformation are not relevant for a binary trait; nor is a Class A model possible.

Nuclear family residual association parameters, analogous to the correlation parameters in regressive models for continuous traits, are incorporated into the model. These are denoted in 5.3.4 below as δ_{FM} for father-mother (spouse), δ_{MO} for mother-offspring, δ_{FO} for father-offspring, and δ_{SS} for any two siblings. In the case of the multivariate logistic distribution these association parameters correspond to second-order correlations; it is assumed that all higher correlations are zero. The actual correlations are calculated from these associations measures for specific logit values [see Karunaratne and Elston (1988)].

For a binary trait, information about the population prevalence of the trait (for a binary trait with variable age of onset, the probability of having been affected since birth) can be incorporated into the likelihood as an independent factor. This is done by specifying that a sample of N independent individuals have been observed, of whom R have been affected for given values of the covariates (and/or up to a specified age), and this may be repeated for different sets of covariate values (The corresponding factor(s) in the likelihood are not shown in the next section). Similarly, the program can output the prevalence, for given sets of covariate values (and/or up to a specified age), estimated from the model using the maximum likelihood estimates of all parameters.

5.3.3.1 Likelihood for a Randomly Sampled Nuclear Family

Let t_F, t_M and t_i be the phenotypes of the father, mother and i-th child, i = 1, 2, ... , n and u_F, u_M and u_i be the types of the father, mother and i-th child. Then the likelihood for a nuclear family is

$$\sum_{u_F}\sum_{u_M}\sum_{u_1}...\sum_{u_n} Pr(u_F)Pr(u_M|u_F)\prod_{i=1}^{n} Pr(u_i|u_F,u_M)L(t_F,t_M,t_1,...,t_n|u_1,...,u_n),$$

where $Pr(u_F)Pr(u_M|u_F)$is the joint probability of types u_F and u_M in the population; $Pr(u_i|u_F,u_M)$is the probability that a sib has type u_i, given the parents' types are u_F and u_M; and the penetrance function $L(t_F,t_M,t_1,...,t_n|u_F,u_M,u_1,...,u_n)$ is given by

$$\begin{aligned}\prod_{i=F,M,1}^{n}\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\{1+\rho_{FO}\left(1-\frac{e^{\theta_u(F)t_F}}{1+e^{\theta_u(F)}}\right)\sum_{i=1}^{n}(-1)^{t_F+t_i}\left(1-\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\right)\\ +\rho_{MO}\left(1-\frac{e^{\theta_u(M)t_M}}{1+e^{\theta_M(u)}}\right)\sum_{i=1}^{n}(-1)^{t_M+t_i}\left(1-\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\right)\\ +\rho_{SS}\sum_{1\le i<j\le n}(-1)^{t_i+t_j}\left(1-\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\right)\left(1-\frac{e^{\theta_u(j)t_j}}{1+e^{\theta_u(j)}}\right)\\ +(-1)^{t_M+t_F}\rho_{MF}\left(1-\frac{e^{\theta_u(M)t_M}}{1+e^{\theta_u(M)}}\right)\left(1-\frac{e^{\theta_u(F)t_F}}{1+e^{\theta_u(F)}}\right)\Big\}.\end{aligned}$$

5.3.4 Finite Polygenic Mixed Model

The finite polygenic mixed model (Fernando et al, 1994; Lange, 1997) can be used for either continuous or binary traits, the only difference being in the particular penetrance function used. It can also be used for binary traits with variable age of onset.

In addition to type (AA, AB or BB), we assume the presence of k diallelic polygenic loci in the model. The alleles at each such locus are a and b, with effects α and β, and frequencies *p* and 1-*p* (the default value of *p* is 0.5). The polygenic effect is the sum of the effects of alleles at all k loci. Thus, if a pedigree member has v a alleles and $(2k-v)$ b alleles, then the polygenic effect is

$$\mu_v = v\alpha + n(2k-v)\beta,$$

where v is called the polygenic number, and α and β are chosen to make the mean polygenic effect zero. It follows that

$$\mu_v = \frac{v-2pk}{1-p}\sqrt{\frac{\sigma_v^2(1-p)}{2pk}},$$

where σ_v^2 is the variance of the polygenic effect.

We assume that, conditional on the polygenic numbers of two parents, the polygenic number of any pedigree member is independent of the polygenic numbers of all other pedigree members. This allows us to use the Elston-Stewart (1971) algorithm summing over the 2k +1 possible genetic numbers times the three possible types. Although this is not strictly consistent with Mendelian inheritance, it leads to a conditional correlation of zero between the polygenic numbers of any two pedigree members.

It is possible to analyze a composite trait and to transform the phenotype in the case of a continuous trait. As for regressive models for continuous traits, the type mean and/or variance can depend on covariates. For a continuous trait, let t_i be the analysis trait for individual i, with expectation conditional on type u:

$$\theta_u(i) = h(\beta_u + \xi_1 x_{i1} + \xi_2 x_{i2} + ... + \xi_{p_\xi} x_{ip_\xi})$$

and let the variance of *t* conditional on type *u* be

$$\eta_u^2(i) = \sigma_u^2 + \varsigma_1 x_{i1} + \varsigma_2 x_{i2} + ... + \varsigma_{p_\varsigma} x_{ip_\varsigma}.$$

Then we define the penetrance function for a continuous trait to be

$$Pr(t_i|u_i, v_i) = \varphi(t_i - \theta_u(i) + \mu_{v_i}, \sigma_i^2),$$

with polygenic variance equal to the variance of μ_{v_i}.

In the case of a binary trait, we define the penetrance function to be the cumulative logistic function

$$Pr(t_i|u_i, v_i) = \frac{e^{\theta_u(i)}}{1 + e^{\theta_u(i)}},$$

where, conditional on type *u*, we have the logit

$$\theta_u(i) = \beta_u + \mu_{v_i} + \xi_1 x_{i1} + \xi_2 x_{i2} + ... + \xi_{p_\xi} x_{ip_\xi}.$$

5.3.4.1 Likelihood for a Randomly Sampled Pedigree

Using the penetrance functions defined above, and letting

$P_i(u_i, u_{Mi}, u_{Fi}, v_i, v_{Mi}, v_{Fi}) =$

$$\begin{cases} Pr(u_i, u_{Mi}, u_{Fi}, v_i, v_{Mi}, v_{Fi}) & if\ the\ parents\ of\ i\ are\ included\ in\ the\ pedigree \\ \psi_i & otherwise \end{cases},$$

and

$$H_i(u_i, v_i, z_i) = \begin{cases} Pr(u_i, u_{M_i}, u_{F_i}, v_i, v_{M_i}, v_{F_i}) & if\ i\ is\ missing, \\ Pr(u_i, u_{M_i}, u_{F_i}, v_i, v_{M_i}, v_{F_i}) Pr(t_i|u_i v_i) & otherwise \end{cases}$$

under random mating the likelihood for a randomly sampled pedigree is

$$L = \sum_{u_1} ... \sum_{u_n} \sum_{v_1} ... \sum_{v_n} \prod_{i=1}^{n} H_i(u_i, v_i, z_i).$$

5.3.5 Binary Traits with Variable Age of Onset

SEGREG currently uses the finite polygenic mixed model for binary traits with variable age of onset.

In general terms, letting a be age of onset and a' the age at examination (for an unaffected person, the last age at which a person is known to be so), the penetrance functions are:

- $\gamma(f(a))$ for an affected person with known age of onset a,
- $\gamma(F(a'))$ for an affected person with unknown age of onset, age at examination a', and
- $1 - \gamma(F(a'))$ for an unaffected person with age at examination a',

where γ is the susceptibility and $f(a)$ is the age of onset density with cumulative distribution $F(a')$.

The mean and variance of f, and susceptibility γ, can each be made dependent on covariates and/or type, in the same way as for a continuous analysis trait and a binary trait, respectively. However, age of onset is assumed to follow a logistic density function rather than a normal density function; because this causes the variance of the age of onset distribution to depend on the mean, it is not permissible for the mean and variance to depend on the same covariate. Letting β be a baseline parameter and α the age coefficient, the density and cumulative distribution functions are:

$$f(a) = \frac{\alpha e^{\beta+\alpha a}}{(1+e^{\beta+\alpha a})^2}$$

$$F(a') = \frac{e^{\beta+\alpha a'}}{1+e^{\beta+\alpha a'}} = [1+e^{-(\beta+\alpha a')}]^{-1}$$

For this distribution, the mean = $-\frac{\beta}{\alpha}$, and the variance = $\frac{\pi^2}{3\alpha^2}$.

As for any other continuous trait, the mean and variance can each depend linearly on covariates, and transformation of "both sides" is possible. Using the logistic distribution has the advantage that the parameters α and β can be interpreted as increases in log odds in the susceptible population (in the whole population if γ = 1).

The susceptibility γ is modeled by a cumulative logistic, in the same way as a binary trait is modeled. In order to avoid confounding among the parameters, there are restrictions on how age of onset and susceptibility depend on type and polygenic number. The following six possibilities are permissible:

1. Age of onset depends on major genotype alone, susceptibility depends on neither major genotype nor polygenic number

2. Age of onset depends on both major genotype and polygenic number, susceptibility depends on neither
3. Age of onset depends on major genotype alone, susceptibility depends on polygenic number alone
4. Susceptibility depends on major genotype alone, susceptibility depends on neither
5. Susceptibility depends on both major genotype and polygenic number, age of onset depends on neither
6. Susceptibility depends on major genotype, age of onset depends on polygenic number alone

As for a binary trait without variable age of onset, information about prevalence (probability of having been affected since birth) can be incorporated into the likelihood, or estimated from the model fitted.

5.4 Program Input

File Type	Description
Parameter File	Specifies the parameters and options with with to perform a particular analysis.
Pedigree Data File	Contains delimited records for each individual including fields for identifiers, sex, parents, trait and marker data.

5.4.1 The `segreg` Parameter

The following table shows the main SEGREG syntax:

`parameter [, attribute]`	**Explanation**	
`segreg`	Starts a SEGREG analysis block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the "root" name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	An alphanumeric constant representing a valid filename
	Default Value	"segreg"
	Required	No
	Applicable Notes	None

5.4.2 The `segreg` Parameter Block

The following lists all parameters that may occur in a SEGREG block (see note 1).

`parameter [, attribute]`	**Explanation**
`title`	Specifies a title for the analysis Value Range: Character string Default Value: None Required: No Applicable Notes: 1
`trait`	Specifies the name of a primary trait. Must be the name of a trait, covariate, or phenotype in the pedigree data file or created by means of a function block. Value Range: Character string Default Value: None Required: Yes Applicable Notes: 1
`, type`	Primary trait type Value Range: {continuous, binary, age_onset} Default Value: continuous, if trait is continuous binary, if trait is binary Required: No Applicable Notes: 2
`composite_trait`	Starts a sub-block for specifying composite trait covariates. Value Range: N/A Default Value: None Required: No Applicable Notes: 3
`type_mean`	Starts a sub-block for specifying type means. Value Range: N/A Default Value: None Required: No Applicable Notes: 4, 19
`type_var`	Starts a sub-block for specifying type variances. Value Range: N/A Default Value: None Required: No Applicable Notes: 5, 19
`type_suscept`	Starts a sub-block for specifying type susceptibilities or penetrances. Value Range: N/A Default Value: None Required: No Applicable Notes: 6, 20

`mean_cov`	Starts a sub-block for specifying covariates for the mean. Value Range: N/A Default Value: None Required: No Applicable Notes: 7, 19
`var_cov`	Starts a sub-block for specifying covariates for the variance. Value Range: N/A Default Value: None Required: No Applicable Notes: 8, 19
`suscept_cov`	Starts a sub-block for specifying covariates for the trait susceptibility or penetrance. Value Range: N/A Default Value: None Required: No Applicable Notes: 9, 20
`class`	Specifies the model class Value Range: {A, D, FPMM, MLM} Default Value: D, for continuous traits MLM, for binary traits FPMM, for age-of-onset traits Required: No Applicable Notes: 10
`fpmm`	Starts a sub-block for specifying an FPMM model. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`resid`	Starts a sub-block for specifying residual correlations (or associations). Value Range: N/A Default Value: None Required: No Applicable Notes: 11
`transformation`	Starts a sub-block for specifying transformation options. Value Range: N/A Default Value: None Required: No Applicable Notes: 12, 19

`geno_freq`	Starts a sub-block for specifying the founder genotype frequency model. Value Range: N/A Default Value: None Required: No Applicable Notes: 13
`transmission`	Starts a sub-block for the specifying the transmission model Value Range: N/A Default Value: None Required: No Applicable Notes: 14
`ascertainment`	Starts a sub-block for specifying the pedigree ascertainment options. Value Range: N/A Default Value: None Required: No Applicable Notes: 15
`prev_constraints`	Starts a sub-block for specifying the constraints on the population prevalence of a binary trait. Value Range: N/A Default Value: None Required: No Applicable Notes: 16
`prev_estimate`	Starts a sub-block for specifying population prevalence model parameters for a binary trait. Value Range: N/A Default Value: None Required: No Applicable Notes: 17
`output_options`	Starts a sub-block for specifying output options. Value Range: N/A Default Value: None Required: No Applicable Notes: 18

Notes

1. Each of the `title` and `trait` parameters is defined by its own block. Except when a binary trait with variable age of onset is being analyzed, no sub-blocks are required (a commingling analysis is automatically performed in this case). Whenever a sub-block is included, there can be required parameters.

2. Only necessary if a trait with variable age of onset is being analyzed, or a binary trait is to be analyzed as a continuous trait.

3. The trait analyzed can be a linear function of the primary trait (with coefficient 1) and other

covariates whose coefficients are fixed or estimated. This linear function is called a composite trait. Without this sub-block a composite trait is not formed. All covariates are centered, the centering (average) value being included as part of the output. The covariates can be any covariate, phenotype or trait (other than the primary trait) listed in the pedigree data file. Note: This sub-block is not applicable to binary traits.

4. This sub-block refers to means of continuous traits. Without this sub-block, one, two and three types are fitted successively (see notes 2 and 3 following the `type_mean` sub-block for an interpretation of the type means).

5. This sub-block refers to variances of continuous traits conditional on type. Without this sub-block, one common variance is fitted.

6. This sub-block refers to logits of susceptibilities (or of penetrances). Without this sub-block, one, two and three types are fitted successively (see notes 2 and 3 following the type_suscept sub-block for an interpretation of the type susceptibilities). Note that it is not possible to fit more than one type when either the **no_trans** or **homog_no_trans** option for transmission is used unless the model includes non-zero residual correlations.

7. This sub-block indicates which covariates are to (linearly) modify the means indicated in the `type_mean` sub-block. Without this sub-block, no such covariates are included in the analysis. All covariates are centered, the centering (average) value being included as part of the output.

8. This sub-block indicates which covariates are to (linearly) modify the variances in the `type_var` sub-block. Without this sub-block, no such covariates are included in the analysis. All covariates are centered, the centering (average) value being included as part of the output.

9. This sub-block indicates which covariates are to (linearly) modify the logits of susceptibilities (or of penetrances) indicated in the `type_suscept` sub-block. Without this sub-block, no such covariates are included in the analysis. All covariates are centered, the centering (average) value being included as part of the output.

10. The values of **A** and **D** denote Bonney's class A and D regressive models, respectively. **FPMM** is the finite polygenic mixed model. Without this parameter, a class D regressive model is used for continuous traits and a multivariate logistic model is used for binary traits. **FPMM** is the only option currently available for binary traits with variable age of onset.

11. This sub-block is not relevant for the FPMM (finite polygenic mixed model). Residual correlations are relevant for continuous traits and residual associations are relevant for binary traits. We use the term "correlations" to cover both situations. Without this sub-block, the usual genetic mixed model assumption of no marital correlation and equal sib-sib and parent-offspring correlations is used.

12. Without this sub-block, the Box-Cox power parameter that provides the best fit to a normal distribution (logistic distribution for age of onset) conditional on type is estimated. An error message will be returned if any value of the analysis trait is at any time necessarily negative. When a composite trait is being analyzed this is avoided as much as possible.

13. Without this sub-block, it is assumed that there is no genotype correlation between spouses, and that there are Hardy-Weinberg equilibrium proportions when fitting three types.

14. Without this sub-block, homogeneity across generations, no transmission, and Hardy-Weinberg equilibrium proportions are assumed.

15. Without this sub-block, it is assumed that the pedigrees are randomly sampled.

16. Without this sub-block, the estimate of population prevalence (more correctly, for a binary trait with variable age of onset, the probability of having been affected since birth) is not constrained by data extraneous to the pedigree file.

17. Without this sub-block, the population prevalence (for a binary trait with variable age of onset, the probability of having been affected since birth) is not calculated.

18. Without this sub-block, the output contains the overall ln(likelihood), -2ln(likelihood) and Akaike's AIC criterion for each of the models that has been maximized in a run.

19. This sub-block is not applicable to binary traits, but does apply to the age-of-onset distribution of a binary trait with variable age of onset.

20. This sub-block is not applicable to continuous traits.

5.5 Sub-Block Syntax: composite_trait

The following table shows the syntax for the composite_trait sub-block (see note 1):

parameter [, attribute]	Explanation	
covariate	Specifies the name of a covariate used to form a composite trait as a linear function of the primary trait and the covariate. This parameter may be specified multiple times. A covariate that is specified in this sub-block may not be also specified in a mean_cov sub-block.	
	Value Range	Character string representing the name of a trait, covariate or phenotype in the pedigree data file or created by means of a function block.
	Default Value	None
	Required	No
	Applicable Notes	1, 3
, val	Specifies the value of the covariate coefficient.	
	Value Range	(-∞, ∞)
	Default Value	None
	Required	No
	Applicable Notes	2
, fixed	Specifies option to fix the covariate coefficient.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	2

Notes

1. This sub-block is not relevant for a binary trait (with or without variable age of onset).
2. If the fixed attribute is set to **true**, the attribute val must be included. If set to **false** and the attribute val is included, this determines the initial value of the variable to be used in the maximization process. If set to **false** and the attribute val is not included, then the program supplies various initial values for the maximization process.
3. A particular trait may not be specified as both a mean covariate and composite trait.

5.6 Sub-Block Syntax: `type_mean`

The following table shows the syntax for the `type_mean` sub-block:

parameter [, attribute]	**Explanation**	
`option`	Specifies `type_mean` option	
	Value Range	one two three two_dom two_rec three_add three_dec three_inc
	Default Value	one
	Required	No
	Applicable Notes	1, 2, 3
`mean`	Specifies the effect of a type. This parameter may be specified as many times as necessary to indicate the values appropriate for the option chosen.	
	Value Range	AA (means β_{AA}) AB (means β_{AB}) BB (means β_{BB}) A* (means β_{AA}=β_{AB}) B* (meansβ_{BB}=β_{AB}) ** (means β_{AA}=β_{AB}=β_{BB})
	Default Value	None
	Required	No
	Applicable Notes	3
`, val`	Specifies value of a given mean.	
	Value Range	(-∞, +∞)
	Default Value	None
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix the given value.	
	Value Range	true false
	Default Value	false
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.

Notes

1. This option refers to the number of types fitted to a continuous trait. Note that if a

`type_mean` sub-block is not included, the program successively fits one, two and three types (see note 4 of the `segreg` block). On the other hand, if a `type_mean` sub-block is included without specifying an option, then only one type is fitted. This sub-block is only relevant for continuous traits. It is relevant for the age of onset distribution of a binary trait with variable age of onset, but is not otherwise relevant for a binary trait.

2. When specified in this sub-block, the type effects are means of continuous distributions. For a binary trait with variable age of onset, they are the mean values of age of onset.

3. Denoting the three type effects β_{AA}, β_{AB}, and β_{BB}, the options correspond to:

Option	Estimated or Fixed
one	$\beta = \beta_{AA} = \beta_{AB} = \beta_{BB}$
two	$\beta_1 = \beta_{AA} = \beta_{AB}$, $\beta_2 = \beta_{BB}$
	$\beta_1 = \beta_{AA}$, $\beta_2 = \beta_{AB} = \beta_{BB}$
three	β_{AA}, β_{AB}, β_{BB}
two_dom	$\beta_{AA} = \beta_{AB}$, β_{BB}
two_rec	β_{AA}, $\beta_{AB} = \beta_{BB}$
three_add	β_{AA}, $\beta_{AB} = (\beta_{AA} + \beta_{BB}) / 2$, β_{BB}
three_dec	$\beta_{AA} \geq \beta_{AB} \geq \beta_{BB}$
three_inc	$\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$

For example,

```
type_mean
{
   option=three_inc
   mean="A*", val=5.0, fixed=false
   mean="BB", val=12.0, fixed=false
}
```

sets initial estimates $\beta_{AA} = \beta_{AB}$= 5.0 and β_{BB}= 12.0 when estimating $\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$.

5.7 Sub-Block Syntax: `type_var`

The following table shows the syntax for the `type_var` sub-block (see note 1):

`parameter [, attribute]`	**Explanation**	
`option`	Specifies `type_var` option	
	Value Range	one two three two_dom two_rec three_add
	Default Value	one
	Required	No
	Applicable Notes	1,2
`var`	Specifies the effect of a type. This parameter may be specified as many times as necessary to indicate the values appropriate for the option chosen.	
	Value Range	AA (means σ^2_{AA}) AB (means σ^2_{AB}) BB (means σ^2_{BB}) A* (means σ^2_{AA}= σ^2_{AB}) B* (means σ^2_{BB} = σ^2_{AB}) ** (means σ^2_{AA} = σ^2_{AB} = σ^2_{BB})
	Default Value	None
	Required	No
	Applicable Notes	2
`, val`	Specifies value of the variance	
	Value Range	[0, ∞)
	Default Value	None
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix the given value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.

Notes

1. This sub-block is only relevant for continuous traits. It is relevant for the age of onset distribution of a binary trait with variable age of onset, but is not otherwise relevant for a binary trait. There can be at most one variance for each type specified in the `type_mean` sub-block.

2. Denoting the three variances σ^2_{AA}, σ^2_{AB} and σ^2_{BB}, the six options are analogous to the first six options in the `type_mean` sub-block (see note 3 of the type_mean sub-block) with σ^2 replacing β.

   For example,

   ```
   type_var
   {
      option=three
      var="AA", val=5.0, fixed=false
      var="B*", val=30.0, fixed=true
   }
   ```

   sets the initial estimate σ^2_{AA}=5.0 and fixed values $\sigma^2_{BB} = \sigma^2_{AB}$=30.0, when estimating only σ^2_{AA}.

5.8 Sub-Block Syntax: `type_suscept`

The following table shows the syntax for the `type_suscept` sub-block:

`parameter [, attribute]`	**Explanation**	
`option`	Specifies `type_suscept` option	
	Value Range	one two three two_dom two_rec three_add three_dec three_inc
	Default Value	one
	Required	No
	Applicable Notes	1, 2, 3
`suscept`	Specifies the effect of a type. This parameter may be specified as many times as necessary to indicate the values appropriate for the option chosen.	
	Value Range	AA (means β_{AA}) AB (means β_{AB}) BB (means β_{BB}) A* (means β_{AA}= β_{AB}) B* (means β_{BB}= β_{AB}) ** (means β_{AA}= β_{AB}=β_{BB})
	Default Value	None
	Required	No
	Applicable Notes	3, 4
`, val`	Specifies value of a given mean.	
	Value Range	(-∞, +∞)
	Default Value	None
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix the given value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.

Notes

1. This option refers to the number of types fitted to a binary trait. Note that if a `type_suscept` sub-block is not included, the program successively fits one, two and three

types (see note 6 of the `segreg` block). On the other hand, if a `type_suscept` sub-block is included without specifying an option, then only one type is fitted.

2. The type effects are mean logits of susceptibilities (of penetrances for a binary trait).

3. Denoting the three type effects β_{AA}, β_{AB}, and β_{BB}, the options correspond to:

Option	Estimated or Fixed
one	$\beta = \beta_{AA} = \beta_{AB} = \beta_{BB}$
two	$\beta_1 = \beta_{AA} = \beta_{AB},\ \beta_2 = \beta_{BB}$
	$\beta_1 = \beta_{AA},\ \beta_2 = \beta_{AB} = \beta_{BB}$
three	$\beta_{AA},\ \beta_{AB}, \beta_{BB}$
two_dom	$\beta_{AA} = \beta_{AB},\ \beta_{BB}$
two_rec	$\beta_{AA},\ \beta_{AB} = \beta_{BB}$
three_add	$\beta_{AA},\ \beta_{AB} = \frac{1}{2}(\beta_{AA} + \beta_{BB}),\ \beta_{BB}$
three_dec	$\beta_{AA} \geq \beta_{AB} \geq \beta_{BB}$
three_inc	$\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$

4. For example,

```
type_suscept
   {
      option=three_inc
      suscept="A*", val= -1.0, fixed=false
      suscept="BB", val= -2.0, fixed=false
   }
```

sets initial estimates $\beta_{AA} = \beta_{AB}$=5.0 and β_{BB}=12.0 when estimating $\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$.

5.9 Sub-Block Syntax: mean_cov

The following table shows the syntax for the mean_cov sub-block:

parameter [, attribute]	**Explanation**
covariate	Covariate to modify the mean of a continuous trait. This parameter may be specified multiple times. A covariate that is specified in this sub-block may not be used in either a composite_trait sub-block or a suscept_cov sub-block.
	Value Range: Character string representing the name of a trait, covariate or phenotype from the pedigree data file or a name created by means of a function block.
	Default Value: None
	Required: No
	Applicable Notes: 1, 3
, val	Specifies the value of the covariate coefficient.
	Value Range: (-∞, +∞)
	Default Value: None
	Required: No
	Applicable Notes: See note 2 of the composite_trait sub-block.
, fixed	Specifies option to fix the given value.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: 3 See note 2 of the composite_trait sub-block.
, interaction	Specifies whether interaction effects are to be estimated.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: 2

Notes:

1. The default is to include no covariates in the analysis. The means indicated in the type_mean sub-block are a linear function of this covariate. All covariates are centered, the centering (average) value being included as part of the output. This sub-block is only

relevant for continuous traits. It is relevant for the age of onset distribution of a binary trait with variable age of onset, but is not otherwise relevant for a binary trait. In the case of an age of onset distribution, the same covariate cannot be specified to modify both the mean and the variance, or both the mean and the susceptibility. In the case of a continuous trait, the same trait cannot be specified as both the mean and composite trait.

2. The `interaction` attribute refers to an interaction with type; the default is to assume no interaction. If there is no interaction, we estimate β_{AA}, β_{AB}, β_{BB} (as many as are specified in the `type_mean` sub-block) and one overall "mean" covariate coefficient for each covariate. If there is interaction, then for this covariate we estimate an additional two interaction effects that sum to 0 if two β parameters are being fitted; and an additional three interaction effects that sum to 0 if three β parameters are being fitted.

3. When analyzing covariates for an age of onset model the `mean_cov` sub-block is allowed, regardless of whether the dependent trait is continuous or binary.In this case, the covariates affect the continuous component of the age of onset model.

5.10 Sub-Block Syntax: `var_cov`

The following table shows the syntax for the `var_cov` sub-block:

parameter [, attribute]	**Explanation**
`covariate`	Covariate to modify the variance (conditional on type) of a continuous trait. Allowable values are the names of traits, covariates or phenotypes in the pedigree data file or created by means of a function block. This parameter may be specified multiple times. Value Range: N/A Default Value: None Required: No Applicable Notes: 1
`, val`	Specifies value of the covariate coefficient Value Range: (-∞, +∞) Default Value: None Required: No Applicable Notes: See note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix the given value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: See note 2 of the `composite_trait` sub-block.
`, interaction`	Specifies whether interaction effects to be estimated. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2

Notes

1. The default is to include no covariates in the analysis. The variances indicated in the `type_var` sub-block are a linear function of this covariate . All covariates are centered, the centering (average) value being included as part of the output. This sub-block is only relevant for continuous traits. It is relevant for the age of onset distribution of a binary trait with variable age of onset, but is not otherwise relevant for a binary trait. In the case of an age of onset distribution, the same covariate cannot be specified to modify both the mean and the variance.

2. The `interaction` attribute refers to an interaction with type; the default is to assume no interaction. If there is no interaction, we estimate σ^2_{AA}, σ^2_{AB}, σ^2_{BB} (as many as are specified

in the type_var sub-block) and one overall "var" coefficient for each covariate. If there is interaction, then for this covariate we estimate an additional two interaction effects that sum to 0 if two σ^2 parameters are being fitted; and an additional three interaction effects that sum to 0 if three σ^2 parameters are being fitted.

5.11 Sub-Block Syntax: `suscept_cov`

The following table shows the syntax for the `suscept_cov` sub-block:

`parameter` **`[, attribute]`**	**Explanation**
`covariate`	Specifies the name of a covariate coefficient to be used in calculating mean logit of susceptibility, or of penetrance, as a linear function of the covariate. A covariate specified in this sub-block may not be used in a mean_cov sub-block. This parameter may be specified multiple times.
	Value Range: Character string representing the name of a trait, phenotype or covariate listed within the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 1
`, val`	Specifies the value of the covariate coefficient.
	Value Range: (-∞, ∞)
	Default Value: None
	Required: No
	Applicable Notes: See note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix this value.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: See note 2 of the `composite_trait` sub-block.
`, interaction`	Specifies option to assume interaction with type.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: 2

Notes

1. The default is to include no covariates in the analysis. The `suscept_cov` sub-block indicates which covariates are to modify the logits of susceptibilities or penetrances indicated in the `type_suscept` sub-block. All covariates are centered, the centering (average) value being included as part of the output. In the case of an age-of-onset distribution, the same covariate cannot be specified as both the mean and the type susceptibility.

2. The “interaction” attribute refers to an interaction with type; the default is to assume no interaction. If there is no interaction, we estimate β_{AA}, β_{AB}, β_{BB} (as many as are specified in the `type_suscept` sub-block) and one overall “suscept” covariate coefficient for each covariate. If there is interaction, then for this covariate we estimate an additional two interaction effects that sum to 0 if two β parameters are being fitted; and an additional three interaction effects that sum to 0 if three β parameters are being fitted.

5.12 Sub-Block Syntax: `fpmm`

The following table shows the syntax for the `fpmm` sub-block:

parameter [, attribute]	**Explanation**
`loci`	Specifies number of polygenic loci. Value Range: {1, 2, 3, ..., 9} Default Value: 3 Required: No Applicable Notes: None
`freq`	Specifies allele frequency of polygenic loci. Value Range: (0,1) Default Value: 0.5 Required: No Applicable Notes: None
`var`	Specifies the variance of the polygenic variable. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`, val`	Specifies the value of the polygenic variance Value Range: (0, $+\infty$) Default Value: None Required: No Applicable Notes: See note 2 below. See also note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix the given value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: See note 2 below. See also note 2 of the `composite_trait` sub-block.
`onset`	Starts a sub-block for a binary trait with variable age of onset. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 3

Notes

1. The following is an example of an `fpmm` sub-block,

```
fpmm {
```

```
      loci=6
      freq=.4
      var, val=10.3, fixed=false
      onset { # See onset sub-block below.
         type_dependent=A
         multi_dependent=N
         status=disease
         age_onset=age
         age_exam=age
      }
   }
```

2. The attribute `val` is optional. Thus, the following two lines are equivalent.

```
var, val=0.2, fixed=false
var=0.2, fixed=false
```

3. This sub-block, nested within the `fpmm` sub-block, is required to analyze a disease trait with variable age of onset: a bivariate trait in which one trait is binary (affected versus unaffected) and the other is continuous (age of onset) censored for unaffected persons. The current version of S.A.G.E. can only analyze such disease traits under the finite polygenic mixed model.

5.13 Sub-Block Syntax: onset

The following table shows the syntax for the onset sub-block:

parameter [, attribute]	**Explanation**
type_dependent	Specifies what is dependent on type. Value Range: {A, S} Default Value: A Required: No Applicable Notes: 1, 2
multi_dependent	Specifies what is dependent on a polygenic component. Value Range: {N, A, S} Default Value: N Required: No Applicable Notes: 2, 3
age_onset	Specifies the age of onset. Value Range: Name of a continuous trait, covariate or phenotype that is either listed in the pedigree data file or created by means of a function block. Default Value: None Required: No Applicable Notes: 4
age_exam	Specifies the age at exam. Value Range: Name of a continuous trait, covariate or phenotype that is either listed in the pedigree data file or created by means of a function block. Default Value: None Required: No Applicable Notes: 4

Notes

1. The type_dependent values have the following meanings:

   A – Age of onset depends on type.

   S – Susceptibility depends on type.

   The option chosen will not cause any dependence on type if A is specified and the type_mean sub-block specifies an option value of **one**, or if S is specified and the type_suscept sub-block specifies an option value of **one**.

2. See note 1 of the `fpmm` sub-block for an example.

3. The multi_dependent values have the following meanings:

   N – There is no polygenic component.

   A – Age of onset has a polygenic component.

   S – Susceptibility has a polygenic component.

4. It is permissible for the `age_onset` and `age_exam` parameters to specify the same continuous trait, in which case the value of this trait is assumed to be age of onset for affected persons and age at exam for unaffected persons.

5.14 Sub-Block Syntax: `resid`

The following table shows the syntax for the `resid` sub-block (see note 1):

parameter [, attribute]	**Explanation**	
`option`	Specifies residual familial correlations.	
	Value Range	equal_po_ss equal_po arb
	Default Value	equal_po_ss
	Required	No
	Applicable Notes	2, 3
`fm`	Specifies the correlation between the residuals of father and mother	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None
`, val`	Specifies value of the residual correlation (ρ).	
	Value Range	(-1, +1), for cont. traits N/A, for binary traits
	Default Value	0
	Required	No
	Applicable Notes	4, See note 2 of the `composite_trait` sub-block.
`, fixed`	Option to fix the given value.	
	Value Range	{true, false}
	Default Value	true
	Required	No
	Applicable Notes	4, See note 2 of the `composite_trait` sub-block.
`mo`	Specifies the correlation between the residuals of mother and offspring.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None

, val	Specifies value of the residual correlation (ρ). Value Range: (-1, +1), for cont. traits; N/A, for binary traits Default Value: 0 Required: No Applicable Notes: 4, See note 2 of the composite_trait sub-block.
, fixed	Option to fix the given value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: See note 2 of the composite_trait sub-block.
fo	Specifies initial correlation between the residuals of father and offspring. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
, val	Specifies value of the residual correlation (ρ). Value Range: (-1, +1), for cont. traits; N/A, for binary traits Default Value: 0 Required: No Applicable Notes: 4, See note 2 of the composite_trait sub-block.
, fixed	Option to fix the given value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: See note 2 of the composite_trait sub-block.
ss	Specifies the correlation between the residuals of siblings. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

, val	Specifies value of the residual correlation (ρ).	
	Value Range	(-1, +1), for cont. traits N/A, for binary traits
	Default Value	0
	Required	No
	Applicable Notes	4, See note 2 of the composite_trait sub-block.
, fixed	Option to fix the given value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	See note 2 of the composite_trait sub-block.

Notes

1. This sub-block is not relevant for the FPMM (finite polygenic mixed model). Residual correlations are relevant for continuous traits and residual associations are relevant for binary traits. We use the term "correlations" to cover both situations.

2. The default option value, **equal_po_ss**, corresponds to the usual genetic mixed model assumption of no marital correlation and equal sib-sib and parent-offspring correlations (only one of the parameters from among mo, fo and ss may be specified)

3. With the second value of the option parameter, **equal_po**, mother-offspring and father-offspring correlations are equal while the father-mother (marital) correlation and sibling-sibling correlation are functionally independent of the parent-offspring correlation and of each other (fm and ss may be specified as well as either mo or fo). With the option value of **arb**, all four correlations: father-mother, mother-offspring, father-offspring, and sibling-sibling are functionally independent of each other and any combination of these correlations may have their attributes specified.

4. The residual value range of (-1, +1) is valid only when modeling continuous data. In the multivariate logistic model used for non-continuous data with residuals, the range is a calculated value that changes based on parameter estimates.

5.15 Sub-Block Syntax: `transformation`

The following table shows the syntax for the `transformation` sub-block (see note 1):

parameter [, attribute]	**Explanation**
`option`	Specifies transformation type. Value Range: none, box_cox, george_elston Default Value: box_cox Required: No Applicable Notes: None
`lambda1`	Specifies of the power parameter, λ_1 Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`, val`	Specifies the value for λ_1. Value Range: (-∞, +∞) Default Value: 1.0 Required: No Applicable Notes: See note 2 of the `composite_trait` sub-block.
`, fixed`	Specifies option to fix $\lambda_1$at the given value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: See note 2 of the `composite_trait` sub-block.
`, lower_bound`	Specifies lower bound for λ_1. Value Range: (-∞, +∞) Default Value: -1 Required: No Applicable Notes: None
`, upper_bound`	Specifies upper bound for λ_1. Value Range: (-∞, +∞) Default Value: ∞ Required: No Applicable Notes: None
`lambda2`	Specifies the shift parameter, λ_2 Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

, val	Specifies the value for λ_2. Value Range: (-∞, +∞) Default Value: 0 Required: No Applicable Notes: See note 2 of the composite_trait sub-block.
, fixed	Option to fix $\lambda_2$at the given value. Value Range: {true, false} Default Value: true Required: No Applicable Notes: See note 2 of the composite_trait sub-block.

Notes

1. This block is not relevant for a binary trait.

5.16 Sub-Block Syntax: geno_freq

The following table shows the syntax for the geno_freq sub-block:

parameter [, attribute]	Explanation
option	Specifies whether Hardy Weinberg equilibrium proportions are to be assumed. Value Range: {hwe, nhwe} Default Value: hwe Required: No Applicable Notes: 1
prob	Specifies probability of a given genotype. This parameter should be specified at most twice and is ignored if the option value is set to **hwe** Value Range: {AA, AB, BB} Default Value: None Required: No Applicable Notes: 2
, val	Specifies value for given probability type. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 2
probs_fixed	Option to fix or not fix genotype probabilities or the probability (relative frequency) of allele (component) A. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 3
freq_A	Specifies the relative frequency of allele A. Used when option value is set to **hwe**, or when option is set to **nhwe** and no prob parameters are specified. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
, val	Specifies value for the allele frequency. Value Range: (0,1) Default Value: None Required: No Applicable Notes:

Notes

1. The **hwe** option imposes Hardy-Weinberg equilibrium proportions, **nhwe** does not.

2. If two `prob` parameters are specified, their sum must be less than 1.

3. If **true**, sufficient information (`val` attributes of `probs_fixed` or `allele_freq_A`, depending on the option chosen) must be specified to fully cover all probabilities. If **false** and a sufficient number of vals are included to specify all probabilities, they determine initial values of the probabilities. If **false** and a sufficient number of vals are not included, the program supplies the necessary initial values for the maximization process.

5.17 Sub-Block Syntax: `transmission`

The following table shows the syntax for the `transmission` sub-block (see note 1):

parameter [, attribute]	**Explanation**	
`option`	Specifies transmission type.	
	Value Range	homog_no_trans homog_mendelian homog_general tau_ab_free general no_trans
	Default Value	homog_no_trans
	Required	No
	Applicable Notes	2, 3, 4
`tau`	Specifies a transmission probability. The `tau` parameter may be specified as many times as necessary to indicate the appropriate values for the model chosen.	
	Value Range	AA (means τ_{AA}) AB (means τ_{AB}) BB (means τ_{BB}) A* (means τ_{AA}= τ_{AB}) B* (means τ_{BB}= τ_{AB}) ** (means τ_{AA}= τ_{AB}= τ_{BB})
	Default Value	None
	Required	No
	Applicable Notes	None
`, val`	Specifies a value for the parameter.	
	Value Range	[0, 1]
	Default Value	None
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.
`, fixed`	Option to fix the given value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	See note 2 of the `composite_trait` sub-block.

no_bounds	Option to remove the range restriction on the transmission probabilities when the option value is set to either **homog_general** or **general**.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	6

Notes

1. This sub-block can only be used if two or three distinct types are specified in either the type_mean or type_suscept sub-block. If this sub-block is missing and a type_mean or type_suscept sub-block is included that specifies two or three types, then all of the option values of this sub-block, with the exception of **no_trans,** are automatically performed.

2. Defining the transmission probability τ_j to be the probability that a person of type j transmits A, and q_A to be the relative frequency of A, these options correspond to:

Option	Estimated or Fixed
homog_no_trans	$\tau_{AA} = \tau_{AB} = \tau_{BB}$ = q_A
homog_mendelian	τ_{AA}= 1, τ_{AB} = .5, τ_{BB} = 0
homog_general	$0 \leq \tau_{AA}$, $\tau_{BB} \leq 1$ τ_{AB}= $(q_A - q_A^2\tau_{AA} - (1-q_A)^2\tau_{BB})/2q_A(1-q_A)$
general	$0 \leq \tau_{AA}, \tau_{AB}, \tau_{BB} \leq 1$
tau_ab_free	$\tau_{AA} = 1, 0 \leq \tau_{AB} \leq 1$,$\tau_{BB} = 0$

3. For the 3 "homogeneous" options **hwe** must be specified in the geno_freq sub-block (or, equivalently, a geno_freq sub-block must not be included).

4. This default is appropriate for commingling analysis with the assumption of Hardy-Weinberg equilibrium proportions.

5. This option together with the option value of **hwe** in the geno_freq sub-block will give the same result as the option value of **homog_no_trans**.

6. Does not apply to a tau parameter for which fixed = **true** or to user-specified initial values. The initial values of the val attribute, if specified, must always lie in the closed interval [0, 1].

5.18 Sub-Block Syntax: `ascertainment`

The following table shows the syntax for the `ascertainment` sub-block:

parameter [, attribute]	**Explanation**
`cond_set`	Specifies the subset of persons on whom ascertainment conditioning is performed.
	Value Range: none, founders, psf, founders_plus_psf
	Default Value: **psf** if `psf_indic` is given a valid value, **none** otherwise.
	Required: No
	Applicable Notes: 1
`psf_indic`	Specifies the proband sampling frame indicator. Must be the name of a trait, covariate or phenotype, binary or continuous, in the pedigree data file or created by means of a function block.
	Value Range: Character string.
	Default Value: None
	Required: No
	Applicable Notes: 1
`psf_indic_include`	Value of the proband sampling frame indicator that is interpreted to mean an individual is included in the proband sampling frame. May be repeated as many times as needed. Any other value of the proband sampling frame indicator, including a missing value, means that the individual is not part of the proband sampling frame.
	Value Range: $(-\infty,+\infty)$
	Default Value: 1
	Required: No
	Applicable Notes: None
`cond_val`	Specifies how a trait value is used to determine the conditioning on a person's phenotype.
	Value Range: actual, gte_thresh, lte_thresh, thresh_indic
	Default Value: actual
	Required: No
	Applicable Notes: 2, 3

, thresh	Threshold value to be used if cond_val is **gte_thresh** or **lte_thresh**. If not specified, the value of thresh is estimated by the program. Value Range: (-∞, +∞) Default Value: None Required: No Applicable Notes: None
, thresh_indic_high	Specifies the value for the greater-than-or-equal-to threshold if cond_val is set to **thresh_indic**. If not specified, the value of thresh_indic_high is estimated by the program. Value Range: (-∞, +∞) Default Value: None Required: No Applicable Notes: 4
, thresh_indic_low	Specifies the value for the less-than-or-equal-to threshold if cond_val is set to **thresh_indic**. If not specified, the value of thresh_indic_high is estimated by the program. Value Range: (-∞, +∞) Default Value: None Required: No Applicable Notes: 4
thresh_indic	Specifies the threshold indicator variable. Must be the name of a continuous trait, covariate or phenotype in the pedigree data file or created by means of a function block. Value Range: Character string Default Value: None Required: No Applicable Notes: 5
, thresh	Specifies the cutoff value for using or not using thresh_indic Value Range: (-∞, +∞) Default Value: 0 Required: No Applicable Notes: 5
onset_option	Specifies the type of conditioning when a binary trait with variable age of onset is being analyzed. Value Range: {actual, by_onset} Default Value: actual Required: No Applicable Notes: 6

Notes

1. This parameter determines whose phenotypes are conditioned on (the "conditioned subset") when calculating a conditional likelihood that allows for ascertainment as follows:

   - A value of **none** indicates that unconditional likelihoods are calculated (i.e. no correction for ascertainment).
   - A value of **psf** indicates the members of the pedigree proband sampling frame and is only permissible if a `psf_indic` parameter is included in the sub-block.
   - A value of **founders** indicates all founder members of the pedigree.
   - A value of **founders_plus_psf** indicates all the founder members and the members of the pedigree proband sampling frame, and is only permissible if a `psf_indic` parameter is included in the sub-block.

2. The `cond_val` parameter is relevant for continuous traits only, and is ignored for binary traits, composite traits, and for age of onset models (for which the onset_option parameter should be used). In the case of binary and composite traits, the default value of **actual** is always used. Also, **actual** is the value used for all founders not included in the proband sampling frame.

3. The meanings of the values of `cond_val` are as follows:

Value	Meaning
actual	Indicates that conditioning is on the actual trait value.
gte_thresh	Indicates that conditioning is on the trait value being greater than or equal to a threshold value.
lte_thresh	Indicates that conditioning is on the trait value being less than or equal to a threshold value.
thresh_indic	Indicates that for each person an indicator variable determines whether to apply the value **gte_thresh** or **lte_thresh.**

4. If the value (specified or estimated) of `thresh_indic_low` is greater than the value of `thresh_indic_high`, a warning message is printed.

5. If `cond_val` is set equal to the value **thresh_indic**, then the value of the threshold indicator variable determines, separately for each individual, which `cond_val` option to apply. The threshold indicator variable should:

   - be equal to `thresh` for those individuals for whom **actual** is to be applied.
   - be greater than or equal to `thresh` for those individuals for whom **gte** is to be applied.
   - be less than or equal to `thresh` for those individuals for whom **lte** is to be applied.

6. This parameter is required if a binary trait with variable age of onset is being analyzed (unless random sampling is to be assumed). If set equal to **actual**, the likelihood is conditioned on the binary phenotype and actual age of onset for each member of the conditioned subset, if available, otherwise by the age at exam. If the value **by_onset** is specified, the likelihood is conditioned on the binary phenotype of each member of the conditioned subset and by the actual age of onset, if available, otherwise by the age at exam. However, **actual** is the value used for all founders not included in the proband sampling frame.

5.19 Sub-Block Syntax: `prev_constraints`

The following table shows the syntax for the `prev_constraints` sub-block:

`parameter` **`[, attribute]`**	**Explanation**	
`constraint`	Starts a sub-block for specifying a particular prevalence constraint. May be repeated as many times as needed.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	1

Notes:

1. Beginning with S.A.G.E. version 4.6, this represents a change from previous versions for the specification of the `prev_constraints` parameter (see 5.19.1).

5.19.1 Sub-Block Syntax: `constraint`

The following table shows the syntax for the `constraint` sub-block:

parameter [, attribute]	**Explanation**
`covariate`	Specifies a covariate on which prevalence (probability of having been affected since birth) depends. Allowable values are the names of traits, covariates or phenotypes in the pedigree data file or created by means of a function block. This parameter may be specified multiple times. Value Range: Character string Default Value: None Required: No Applicable Notes: 1, 2, 5
`, val`	Specifies a value for the covariate. Value Range: (-∞, +∞) Default Value: None Required: Yes Applicable Notes: 2
`R`	Specifies the number of affected persons in a random sample. Value Range: (0, N) Default Value: None Required: No Applicable Notes: 3, 5
`N`	Specifies the sample size. Value Range: (R, $+\infty$) Default Value: None Required: No Applicable Notes: 3, 5
age	Specifies the age at which prevalence (probability of having been affected since birth) should be computed. Required for age of onset traits, and disallowed otherwise. Value Range: (0, $+\infty$) Default Value: None Required: No Applicable Notes: 2, 4, 5

Notes

1. Any covariate in this sub-block must also appear in the `mean_cov, var_cov` or `suscept_cov` sub-blocks.

2. Any covariate (including age) upon which prevalence depends and which is not specified as a covariate parameter, or for which a value is not specified, is set at its mean value.

3. It is assumed that, independent of the pedigree data, R of N persons are affected by the specified values of the covariates. R and N need not be integers.

4. The literal string **`infinity`** must be entered to indicate an "infinite" age.

5. The following example illustrates the constraint syntax:

```
segreg, out = myAnalysis {
   trait = BMI, type = continuous
   trait = aff, type = age_onset
   .
   .
   .
   prev_constraints {
      constraint {
         covariate = height
         covariate = weight
         age = infinity
         N = 1000
         R = 100
      }
      constraint {
         covariate = smoking
         covariate = drinking
         age = infinity
      }
      .
      .
      .
   }
}
```

5.20 Sub-Block Syntax: `prev_estimate`

The following table shows the syntax for the `prev_estimate` sub-block:

parameter [, attribute]	**Explanation**
`covariate`	Specifies a covariate on which prevalence depends. Allowable values are the names of traits, covariates or phenotypes in the pedigree data file or created by means of a function block. This parameter may be specified multiple times. Value Range: Character string Default Value: None Required: No Applicable Notes: 1, 2
`, val`	Specifies a value for the covariate. Value Range: (-∞, +∞) Default Value: None Required: No Applicable Notes: 2
age	Specifies the age at which prevalence (probability of having been affected since birth) should be computed. Required for age of onset traits, and disallowed otherwise. Value Range: (0, $+\infty$) Default Value: None Required: No Applicable Notes: 2, 3

Notes

1. Any covariate in this sub-block must also appear in the mean_cov or suscept_cov sub-blocks. Age of onset (or age at exam) may also be included as a covariate if an onset sub-block is included, and then prevalence is interpreted as the probability of having been affected since birth up to the specified age.

2. Any covariate upon which prevalence depends, but is not specified as a covariate parameter is set at its mean value.

3. The literal string **`infinity`** may be entered to indicate an “infinite” age.

5.21 Sub-Block Syntax: output_options

The following table shows the syntax for the output_options sub-block:

parameter [, attribute]	**Explanation**
type_prob	Specifies option to calculate type probabilities. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 1
pen_func_out	Specifies option to create penetrance function output file. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2

Notes

1. Type probabilities can only be calculated if two or three types are specified in the type_mean sub-block. In either case three probabilities (summing to 1) are output for an individual: the probabilities of being AA, AB or BB conditional on the model and all the pedigree information available, substituting maximum likelihood estimates for all unknown parameters.

2. This file can be used as input to LODLINK or MLOD. Because this only makes sense if the transmission option **homog_mendelian** has been chosen, it will only be produced if that option is among those chosen in the transmission sub-block; otherwise, the option is ignored.

5.22 Program Execution

SEGREG is run via a command line interface on the supported UNIX and Windows platforms. This requires that the S.A.G.E. programs are properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running SEGREG from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line:

```
>segreg
S.A.G.E. v5.x -- SEGREG
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: ./segreg <parameters>  <pedigree>
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of SEGREG may look like the following:

```
>segreg segreg.par example.ped
S.A.G.E. v5.x -- SEGREG
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading parameter file.....................done.
Reading pedigree file.....................
from example.ped............done.
Sorting pedigrees.........................done.
Generating statistics.....................done.
Analysis complete!
```

5.23 Program Output

Output files produced by SEGREG containing results and diagnostic information are:

File Name	File Type	Description
segreg.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
segreg.sum	SEGREG summary output file	Contains the table of final estimates of the parameters and their standard errors and other results.
segreg.det	SEGREG detailed output file	Contains the table of final estimates and variance-covariance matrix of the parameters.
segreg.typ	Trait genotype probability output file	Contains the individual specific type probabilities conditional on the model and all the pedigree information available, and is suitable for input into model-based linkage programs such as MLOD and LODLINK.
segreg.pen	Trait penetrance function output file	Contains individual specific penetrance information. Will only be produced if the **homog_mendelian** option of the `transmission` parameter has been enabled.

5.23.1 Information Output File

The SEGREG Information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read from the pedigree data file, are included with all programs in S.A.G.E. Release and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that this file be checked for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "segreg.inf" should be checked for errors and diagnostic information after each run of the program.

5.23.2 Summary Output File

The SEGREG summary output file stores the table of final estimates of the parameters with model information.

5.23.3 Detailed Output File

The SEGREG Detailed output file stores the variance-covariance matrix as well as what the Summary File has.

5.24 Example Output File

5.24.1 Summary Output File

Here is a typical example of a SEGREG summary output file.

```
================================================================================================
SEGREG Analysis for Trait : dbh
================================================================================================
  # Model Specification

      Model Class A
      Type means                          : three means
      Type variances                      : one variance
      Genotype frequency                  : Hardy-Weinberg equilibrium
      Residual correlations               : no spouse correlation,
                                            parent-offspring and sib-sib correlations equal
      Transmission                        : homogeneous mendelian
      Transformation                      : Box-Cox
      Covariate means
          cov1        :  (Mean =      0.16146)
          cov2        :  (Mean =      0.68576)
          cov3        :  (Mean =      0.15278)

  # Number of constituent pedigrees       : 4
  # Number of singletons                  : 0

  # Final Estimates :

  Parameter           Parameter Est.  Standard Err.   First Deriv.   Status
------------------------------------------------------------------------------------------------
  mean_AA              8.37728311      1.58425845      0.00001649     IND, MAY VARY
  mean_AB             24.83059248      2.25001207      0.00000000     IND, MAY VARY
  mean_BB             43.13638675      1.93512951      0.00001281     IND, MAY VARY
  variance           105.3057234     14.03526432     -0.00000393     IND, MAY VARY
  prob_AA              0.16311761      0.05099851                     DEPENDENT
  prob_AB              0.48152120      0.02427498                     DEPENDENT
  prob_BB              0.35536119      0.07527349                     DEPENDENT
  freq_A               0.40387821      0.06313600      0.00027621     IND-FN, MAY VARY
  genotypic corr.      0.00000000      0.00000000                     FIXED EXTERNALLY
  marital resid c      0.00000000      0.00000000                     FIXED EXTERNALLY
  po = ss resid c      0.07655435      0.13603358      0.00027621     IND, MAY VARY
  transm prob_AA       1.00000000      0.00000000                     FIXED EXTERNALLY
  transm prob_AB       0.50000000      0.00000000                     FIXED EXTERNALLY
  transm prob_BB       0.00000000      0.00000000                     FIXED EXTERNALLY
  lambda_one           0.53976055      0.05940720     -0.00025587     IND, MAY VARY
  lambda_two           0.00000000      0.00000000                     FIXED EXTERNALLY
  cov1                -3.71081358     13.74889789      0.00007443     IND, MAY VARY
  cov2                 1.13614517     13.54930873     -0.00012156     IND, MAY VARY
  cov3                 1.58174041     13.87732080      0.00000000     IND, MAY VARY
------------------------------------------------------------------------------------------------

        LN(Likelihood) : -1244.44
     -2 LN(Likelihood) : 2488.87
    Akaike's AIC score : 2526.87
================================================================================================
```

5.24.2 Detailed Output File

Here is a typical example of a SEGREG detailed output file.

```
================================================================================================
SEGREG Analysis for Trait : dbh
================================================================================================
  # Model Specification

      Model Class A
      Type means                          : three means
      Type variances                      : one variance
      Genotype frequency                  : Hardy-Weinberg equilibrium
      Residual correlations               : no spouse correlation,
                                            parent-offspring and sib-sib correlations equal
      Transmission                        : homogeneous mendelian
      Transformation                      : Box-Cox
      Covariate means                     :
          cov1        :  (Mean =      0.16146)
          cov2        :  (Mean =      0.68576)
          cov3        :  (Mean =      0.15278)

  # Number of constituent pedigrees       : 4
  # Number of singletons                  : 0

  # Final Estimates :
.
.
.
  # Variance-Covariance Matrix :
             |     mean_AA         mean_AB         mean_BB         variance   . . . . . .
-------------------------------------------------------------------------------------------
  mean_AA    |      2.50987483      2.16787316      0.82168246      7.45485406
  mean_AB    |      2.16787316      5.06255433      1.28400089     11.11488621
  mean_BB    |      0.82168246      1.28400089      3.74472620     -5.88210645
  variance   |      7.45485406     11.11488621     -5.88210645    196.9886444
  prob_AA    |      0.04014903      0.07066268      0.04384616      0.10221457
  prob_AB    |      0.01911069      0.03363501      0.02087050      0.04865352
  prob_BB    |     -0.05925972     -0.10429769     -0.06471666     -0.15086809
  freq_A     |      0.04970438      0.08748019      0.05428141      0.12654133
  po = ss res|      0.06303292      0.07646595      0.00760523      0.82768163
  lambda_one |      0.03847106      0.03240227      0.03610166      0.03125240
  cov1       |     -0.92646184     -1.53058196      0.91183742     -6.97375559
  cov2       |      0.14707161      0.20870282     -0.24557958      0.17034505
  cov3       |     -0.41230018     -0.36967265     -0.55733522     -2.43238363
-------------------------------------------------------------------------------------------

        LN(Likelihood) : -1244.44
     -2 LN(Likelihood) : 2488.87
    Akaike's AIC score : 2526.87
================================================================================================
```

Chapter 6

MARKERINFO

MARKERINFO detects Mendelian inconsistencies in pedigree data. These inconsistencies are sorted by marker, by pedigree, and by whether one or more than one nuclear family is involved in the inconsistency.

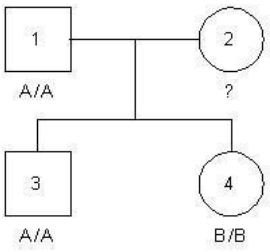
6.1 Limitations

MARKERINFO analyses one marker at a time and is only guaranteed to detect all errors in the absence of loops. Mendelian inconsistencies cannot be localized beyond the nuclear family in which they are first detected (see theory).

6.2 Theory

The phenoset of an individual is the set of all genotypes consistent with that individual's phenotype. Individuals labeled as missing are considered to be consistent with all possible phenotypes. MARKERINFO detects Mendelian inconsistencies in pedigree data by reducing the set of possible genotypes for each individual to the minimal possible subset on the basis of both the individual's phenoset and the phenosets of surrounding individuals. An empty minimal subset of genotypes for any individual indicates a Mendelian inconsistency.

Example 1

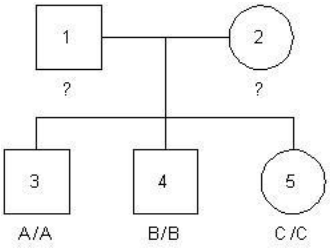


In this example pedigree, individuals 1 and 3 have a phenoset consisting of genotype A/A, while individual 4 has a phenoset consisting of genotype B/B. Individual 2 is unknown, so her phenoset includes all possible genotypes: {A/A, A/B, B/B, etc.}

These phenosets are reduced based on Mendelian inheritance from parent to child. Under Mendelian inheritance, a parent having marker genotype A/B can transmit either the A or the B allele to the child, but cannot transmit any other allele at that marker. Any genotype for which there is no valid transmission from a parent or to a child is removed from the phenoset. In this way, the subset of possible genotypes for individual 2 becomes A/B and that for individual 1 become empty.

MARKERINFO detects two sorts of inconsistencies, those involving one, and those involving more than one, nuclear family. In the above example, there is no valid transmission from individual 1 to individual 4 because 4 must receive a B allele from both parents and 1 has no B allele. In this and the next example it is sufficient to inspect a single nuclear family to detect an inconsistency.

Example 2:

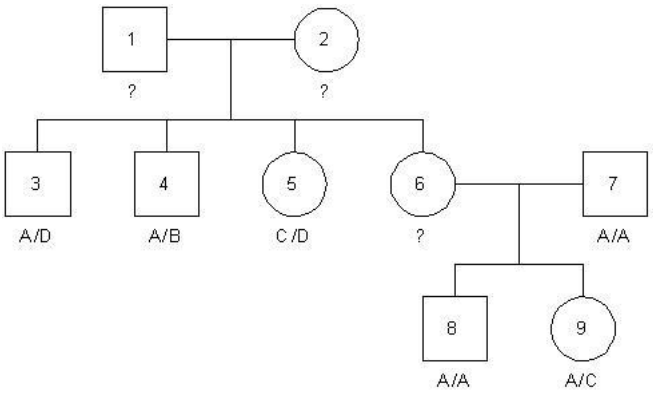


In this example, each of the children must receive a different set of alleles from each of their parents, but each parent has only two alleles. At least one child must be inconsistent with the parents, but it is impossible to determine which one.

Inconsistencies Involving More than one Nuclear Family

Often, a single nuclear family appears consistent until new information is added from surrounding nuclear families. Consider example 3.

Example 3:



Looking at only the nuclear family with parents 1 and 2, we see that this family is consistent, with 1 and 2 each having subset of possible genotypes A/C, B/D. Note here that if 1 is A/C, 2 must be B/D and vice versa. From this, we can deduce that the subset of possible genotypes for 6 is A/D, A/B, C/D, B/C.

Similarly, the nuclear family with parents 6 and 7 is consistent, with the subset of possible genotypes for 6 being A/C. However, A/C is not present in the subset of possible genotypes for 6 as derived from the first nuclear family. There is no genotype present in both subsets, so the minimal subset is empty. Because the sequence in which MARKERINFO traverses the pedigree depends on several factors, the inconsistency could be first detected in either of the nuclear families, and only one of them will be reported as being inconsistent.

6.3 Program Input

MARKERINFO requires the following input files in order to run:

File Type	Description
MARKERINFO parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual including fields for identifiers, parents, trait and marker data.

6.3.1 The `markerinfo` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main MARKERINFO parameter.

`parameter [, attribute]`	**Explanation**	
`markerinfo`	Starts MARKERINFO block	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	markerinfo
	Required	No
	Applicable Notes	None

6.3.2 The `markerinfo` Block

The following lists all parameters that may occur in a `markerinfo` block.

`parameter` **`[, attribute]`**	**Explanation**
`sample_id`	Specifies an extra ID field to be printed in the analysis output file.
	Value Range: Character string representing the name of a field in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 1
`consistent_out`	Specifies that consistent nuclear family members should be added to the output.
	Value Range: true, false
	Default Value: false
	Required: No
	Applicable Notes: 2

Notes

1. The value of `sample_id` should be set equal to the name of a field read from the pedigree data file. This can be used to indicate the location where a sample is stored.
2. If `consistent_out` is set to **true**, then the nuclear family members who are not inconsistent are added to the output with [] around them.

6.4 Program Execution

MARKERINFO is run via a command line interface on the supported UNIX and Windows platforms. This requires that the S.A.G.E. programs are properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running MARKERINFO from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement.

```
>MARKERINFO
S.A.G.E. v5.x -- MARKERINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: markerinfo <parameters> <pedigree>
Command line parameters:
  parameters   - parameter file
  pedigree     - pedigree data file
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of MARKERINFO may look like the following:

```
>markerinfo par example.ped
S.A.G.E. v5.x -- MARKERINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading Parameter File...................done.
Reading pedigree file.....................
              from example.ped............done.
Sorting pedigrees........................done.
Markerinfo analysis.......
Processing pedigree '1'..................done.
Processing pedigree '102'................done.
Processing pedigree '104'................done.
Processing pedigree '105'................done.
Processing pedigree '106'................done.
Processing pedigree '107'................done.
Processing pedigree '108'................done.
.
.
.
```

6.5 Program Output

Output files produced by MARKERINFO containing results and diagnostic information are:

Filename	Filetype	Description
`markerinfo.inf`	Information output file	Contains informational diagnostic messages, warnings and program errors. No calculation results are stored in this file.
`markerinfo.out`	Analysis output file	Contains Mendelian inconsistency information on markers (See note)

Note:

Two types of Mendelian inconsistencies are differentiated: those which occur within a single nuclear family, and those in which members of more than one nuclear family are involved – i.e. the inconsistency can only be detected if two or more nuclear families are simultaneously examined. In the latter case, only one of the nuclear families that could be involved is shown in the output, followed by *.

6.5.1 Information Output File

The MARKERINFO information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read from the pedigree data file, are included with all programs in S.A.G.E. Release and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results

of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "markerinfo.inf" should be checked for errors and diagnostic information after each run of the program.

6.6 Example Output File

Here is a typical example of MARKERINFO output:

```
S.A.G.E. v5.x -- MARKERINFO
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
=============================================================================
  MARKERINFO Analysis Output
=============================================================================
-----------------------------------------------------------------------------
    Part 1: Number of Inconsistencies per pedigree/marker
-----------------------------------------------------------------------------
=============================================================================
                                          Number of Markers
Pedigree                          Incon.     Informative     Total
----------                      ----------  -------------  ----------
124                                    3            323         324
155                                    3            321         324
.
.
.
=============================================================================
=============================================================================
                                        Number of Pedigrees
Marker                            Incon.     Informative     Total
----------                      ----------  -------------  ----------
F13A1                                 61             94          94
1s225                                  3             93          94
D1S245                                 2             94          94
D14S608                                1             94          94
D2S1328                                1             94          94
D2S1334                                1             91          94
=============================================================================
-----------------------------------------------------------------------------
    Part 2: Inconsistencies
-----------------------------------------------------------------------------
    missing code = 0
    * More than one nuclear family must be examined to detect
      the inconsistency.
=============================================================================
Table 1 with Marker F13A1  1s225  D1S245
=============================================================================
Pedigree    Individual            F13A1         1s225        D1S245
----------  ----------          ----------    ----------    ----------
2           2          Mother    0
2           1          Father    1/1
2           3
2           4                    4/5
.
.
.
124         2          Mother                * 0           * 0
124         1          Father                * 0           * 0
124         4                                * 2/9         * 4/6
124         5                                * 8/13        * 1/4
124         6                                * 0           * 0
```

```
.
.
.
=======================================================================
=======================================================================
Table 2 with Marker D14S608  D2S1328  D2S1334
=======================================================================
Pedigree    Individual          D14S608       D2S1328       D2S1334
----------  ----------          ----------    ----------    ----------
66          16         Mother   * 3/3
66          17         Father   * 0
66          27                  * 3/5
66          28                  * 3/4
66          29                  * 3/4
.
.
.
```

Chapter 7

FREQ

FREQ is a program that estimates allele frequencies from marker data among related individuals with known pedigree structure and generates marker locus description files, needed by GENIBD, MLOD, and other S.A.G.E. programs. Future versions will also have the ability to estimate genotype and haplotype frequencies in the presence of allelic disequilibrium.

7.1 Limitations

Maximum likelihood estimates of allele frequencies can only be calculated using information from pedigrees without mating rings or other loops. Any pedigrees with loops will automatically be skipped for maximum likelihood estimation. Sometimes numerical problems occur and standard errors of the frequency estimates cannot be calculated. Also, the computational time required to calculate maximum likelihood frequency estimates increases greatly with the number of alleles at any locus.

7.2 Theory

7.2.1 Initial Frequency Estimator

FREQ begins its analysis by computing allele frequencies using only founders and singletons (unrelated and unconnected individuals) from each pedigree for all codominant marker phenotypes (Boehnke 1991). These estimates are calculated by summing the number of times each allele appears and dividing by the total number of observed alleles. This estimator tends to be sub-optimal since much of the data are not used and, in many datasets, founders are not typed.

A second estimator is provided that attempts to use marker information from non-founders and non-singletons by assuming that they are independent. Calculation is performed the same way as for the founders, by counting the number of times each allele appears and dividing by the total number of observed alleles. These estimates can be reported directly, or combined with the founder-only based estimates by giving the `founder_weight` parameter a value. When the founder weight is not set, the founder and non-founder frequencies are combined by adding the number of times each allele appears in both founders and non-founders and dividing by the total number of observed

alleles from both. When `founder_weight` is set to a number between 0 and 1, say w, then a weighted average of the founder and non-founder frequencies is taken, with weights w and $1 - w$, respectively. Setting `founder_weight` to 1 generates founder-only frequency estimates, while setting `founder_weight` to 0 results in non-founder-only frequency estimates.

These methods provide consistent but statistically inefficient frequency estimates which can be used for datasets that have many pedigrees with loops or markers with too many alleles for the frequencies to be computed efficiently, as well as automatically provide initial estimates for maximum likelihood estimation.

7.2.2 Maximum Likelihood Estimator

The likelihood formulation assumes that, with respect to the marker loci, the pedigrees are randomly ascertained from a single random mating population, and that genotypes occur with Hardy-Weinberg equilibrium frequencies. The likelihood for the data at each marker in the whole sample is numerically maximized over possible allele frequencies to obtain the maximum likelihood estimates for that marker. Standard errors are computed by double differentiation of the log likelihood. Those frequencies that maximize the likelihood are then reported. Non-codominant markers are fully supported, provided that the phenotype to genotype mapping is provided in a locus description file. It should be noted that singletons (unrelated and unconnected individuals) may be included in the data; they are simply one-person pedigrees with parent information missing and, as such, require no special treatment in the model.

7.3 Program Input

FREQ requires the following input files in order to run:

File Type	Description
FREQ Parameter File	Specifies the parameters and options with which to perform a particular analysis.
Pedigree Data File	Contains delimited records for each individual including fields for identifiers, sex, parents, trait and marker data.
Marker Locus Description File	Lists the alleles, allele frequencies and phenotype to genotype mapping for each marker locus. Only needed in FREQ for non-codominant markers.

7.3.1 The `freq` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main FREQ parameter.

`parameter [, attribute]`	**Explanation**	
`freq`	Starts FREQ analysis sub-block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	freq.out
	Required	No
	Applicable Notes	None

7.3.2 The `freq` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `freq` sub-block.

`parameter [, attribute]`	**Explanation**
`founder_weight`	The weight used for founders to combine founder-only and approximate non-founder frequency estimates. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 1
`skip_mle`	Specifies whether to skip maximum likelihood estimate computation of allele frequencies. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`marker`	Names a marker to be included in the current analysis. Value Range: Character string representing the name of a marker listed in the pedigree data file. Default Value: None Required: No Applicable Notes: 2

Notes

1. This parameter is useful when consistent (but inefficient) estimates are required from a dataset with many alleles. When not specified, the estimates labeled as "All Pedigree Members" are obtained on the assumption that all observed alleles are independent.
2. The value of a `marker` parameter should be set to the name of a marker for which allele frequencies are to be estimated. If no valid marker parameters are listed, then all markers are used.

7.4 Program Execution

FREQ is run via a command line interface on the supported UNIX and Windows platforms. This requires that the S.A.G.E. programs are properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running FREQ from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line:

```
>freq
S.A.G.E. v5.x -- FREQ
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY
usage: freq <parameters>  <pedigree> [locus]
Command line parameters:
parameters - Parameter File
pedigree - Pedigree Data File
locus - Locus Description File (optional)
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of FREQ may look like the following:

```
> freq params ped
S.A.G.E. v5.x -- FREQ
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY
Reading Parameter File.....................done.
Reading Pedigree File.....................
from ped.....................done.
Sorting pedigrees.........................done.
Estimating allele frequencies. (default analysis)
================================================
Detailed output file: freq.out
Locus description file: freq.loc
Processing marker: D5G1
Processing marker: D5G2
Processing marker: D5G3
Processing marker: D5G4
Processing marker: D5G5
Processing marker: D5G6
Processing marker: D5G7
Processing marker: D5G8
Processing marker: D5G9
Processing marker: D5G10
Processing marker: D5G11
Processing marker: D5G12
Processing marker: D5G13
Processing marker: D5G14
Processing marker: D5G15
Processing marker: D5G16
Processing marker: D5G17
Processing marker: D5G18
```

7.5 Program Output

Output files produced by FREQ containing results and diagnostic information are:

Filename	Filetype	Description
`freq.inf`	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
`freq.out`	Analysis output file	Contains detailed tables of analysis options and results.
`freq.loc`	Locus Description File	An output file that presents the allele frequencies estimated by FREQ in a form that may be read directly into any other S.A.G.E. program that requires it.

7.5.1 Information Output File

The FREQ information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read from the pedigree data file, are included with all programs in S.A.G.E. and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "freq.inf" should be checked for errors and diagnostic information after each run of the program.

7.5.2 Locus Description File

Allele frequency estimates are output into the locus description file according to the following priority:

1. Maximum likelihood estimates
2. Weighted estimates
3. Non-weighted/naive estimates

For example, if `skip_mle` = **true**, and `founder_weight` is set to some nonzero value, then the locus description file will contain weighted estimates. On the other hand, if `skip_mle` is set equal to **false**, then the locus description file will contain maximum likelihood estimates.

7.5.3 Example Output File

Here is a typical example of FREQ output:

```
S.A.G.E. v4.3 -- FREQ
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY
File generated on : Tue Oct  8 16:55:13 2002
Estimating allele frequencies. (default analysis)
================================================
Locus description file: freq.loc
Allele frequency estimates for marker: D5G1
                Frequency Estimates:
                Founders      All Pedigree  Maximum      Standard
  Allele        Only          Members       Likelihood   Error
  ------------  ------------  ------------  ------------ ------------
  B               0.2918410     0.2912371     0.2918410  ( 0.0147032)
  C               0.3033473     0.2993986     0.3033473  ( 0.0148680)
  D               0.3012552     0.3058419     0.3012552  ( 0.0148388)
  A               0.0575314     0.0640034     0.0575314  ( 0.0075311)
  E               0.0460251     0.0395189     0.0460251  ( 0.0067770)
Allele frequency estimates for marker: ABO
  Marker was determined to be non-codominant, so only
  Maximum Likelihood estimates were computed.
                Frequency Estimates:
                Maximum      Standard
  Allele        Likelihood   Error
  ------------  ------------ ------------
  A1              0.1894737  ( 0.0290427)
  A2              0.0754243  ( 0.0212204)
  B               0.0428382  ( 0.0145766)
  O               0.6922637  ( 0.0345212)
Allele frequency estimates for marker: PGD
                Frequency Estimates:
                Founders      All Pedigree  Maximum      Standard
  Allele        Only          Members       Likelihood   Error
  ------------  ------------  ------------  ------------ ------------
  A               0.9837398     0.9699499     0.9727053  ( 0.0089518)
  C               0.0162602     0.0300501     0.0272947  ( 0.0089518)
Allele frequency estimates for marker: C3
                Frequency Estimates:
                Founders      All Pedigree  Maximum      Standard
  Allele        Only          Members       Likelihood   Error
  ------------  ------------  ------------  ------------ ------------
  F               0.2032520     0.2128548     0.1972428  ( 0.0217400)
  S               0.7926829     0.7838063     0.7941248  ( 0.0220807)
  F'              0.0040650     0.0033389     0.0086324  ( 0.0049628)
```

Chapter 8

GENIBD

GENIBD is a program for generating identity-by-descent (IBD) sharing distributions from family data and genetic marker loci by a variety of algorithms tuned for various types of pedigrees. Three methods of generating IBD sharing are provided: the Single Marker IBD Analysis (single-point only), the Exact IBD Analysis (single- or multi-point), and the Simulation IBD Analysis (single- and multi-point). Control of which algorithms are used in a given analysis is provided to the user through convenient automatic switching parameters. IBD sharing distributions are generated for five types of relative pairs: sibling, half sibling, avuncular, grandparental and cousin. In future versions there will be options to generate more types of pairs. The resulting output file(s) list at each location the probability of each pair sharing 0 or 2 alleles IBD, and the difference between the paternal and maternal probability of sharing 1 allele IBD, conditional on the marker data available. This file can then be read into other programs (e.g. SIBPAL) for analyses.

8.1 Limitations

IBD sharing for only five pair types can be generated:

1. full sib,
2. half sib,
3. grandparental,
4. avuncular and
5. first cousin.

There are three methods currently implemented that generate IBD sharing distributions. Each method has distinct capabilities and limitations:

8.1.1 Single Marker IBD Analysis

The Single Marker IBD Analysis uses complete information at each single marker to generate the IBD distributions for each pair of relatives at that marker. It is strictly a single-point method, and does not support pedigrees with loops.

8.1.2 Exact IBD Analysis

The Exact IBD Analysis computes the likelihood of each inheritance vector at one or several markers (including locations interpolated between markers) to generate IBD distributions for each pair of the five supported types of relative pairs at each marker. It can be used for either single- or multi-point analysis in pedigrees with or without loops and is not restricted in the type of relative pair for whom IBD is computed. It is, however, restricted to small pedigrees due to the exponential nature of the algorithm related to the number of individuals in the pedigree. The time and space complexity of the algorithm is largely characterized by the exponent $2n - f$, the number of bits in an inheritance vector, where n is the number of non-founders and f is the number of founders in a pedigree. During parameter specification the maximum value of $2n - f$ may be set; any pedigree that has a value larger than the limit will use another of the analysis methods, if possible, or be skipped.

8.1.3 Simulation IBD Analysis

The Simulation IBD analysis uses a Markov chain Monte Carlo (MCMC) simulation over the space of possible inheritance vectors for each pedigree to estimate the IBD distribution for each pair of the five supported pair types at each marker, without interpolation at locations between markers. Several batches are run to ensure coverage of the state space. Generation of IBD distributions at points between markers can be accomplished putting markers with no data at those locations.

Also note that since this is a simulation method, values differ between runs of the program. This method may be quite time consuming so it is only used when pedigrees are too large for the exact IBD analysis.

8.2 Theory

Let $\hat{f}_{imj}$ be the probability, conditional on the marker data available, that relative pair j shares exactly i alleles IBD at marker m,where i= 0, 1 or 2. GENIBD calculates $\hat{f}_{imj}$ for each marker locus of interest for each of five types of relative pair in the data set as follows.

Given the marker data I_m for a single pedigree at marker m

$$\hat{f}_{imj} = \frac{P\left(I_m \left| pair\ j\ shares\ i\ alleles\ IBD\right.\right) P\left(pair\ j\ shares\ i\ alleles\ IBD\right)}{L\left(I_m\right)} \tag{8.1}$$

or

$$\hat{f}_{imj} = \frac{P\left(I_m \left| pair\ j\ shares\ i\ alleles\ IBD\right.\right) P\left(pair\ j\ shares\ i\ alleles\ IBD\right)}{L\left(I_m\right)} \tag{8.2}$$

where L (I_m) is the likelihood for the pedigree at marker m and Pr(pair j shares i alleles IBD) is the prior probability that depends on relationship alone. L (I_m) does not depend on the individual pair and is thus only calculated once for each pedigree at each marker locus.

In the case of full sib and half sibs, for i = 1 and pair j, the components $\hat{f}_{1mj-maternal}$ and $\hat{f}_{1mj-paternal}$ of $\hat{f}_{1mj}$ are calculated separately, depending on the sex of the parent from whom

the sharing allele is descended, as follows:

$$\hat{f}_{1mj-maternal} = \frac{P(I_m|pair\ j\ shares\ 1\ maternal\ allele\ IBD)P(pair\ j\ shares\ 1\ maternal\ allele\ IBD)}{L(I_m)}$$

$$\hat{f}_{1mj-paternal} = \frac{P(I_m|pair\ j\ shares\ 1\ paternal\ allele\ IBD)P(pair\ j\ shares\ 1\ paternal\ allele\ IBD)}{L(I_m)}$$

or

$$\hat{f}_{1mj-maternal} = \frac{P(I_m, pair\ j\ shares\ 1\ maternal\ allele\ IBD)}{L(I_m)}$$

$$\hat{f}_{1mj-paternal} = \frac{P(I_m, pair\ j\ shares\ 1\ paternal\ allele\ IBD)}{L(I_m)}$$

.

The difference $(\hat{f}_{1mj-maternal} - \hat{f}_{1mj-paternal})$ is reported in the GENIBD output for every marker location, denoted in the output as f1m-f1p.

The methods used to calculate these values depend on the type of analysis used.

8.2.1 Single Marker Analysis

In the case of single marker analysis, only information at a single locus is used, with L (I_m) calculated using the recursive methods described in Fernando, Stricker and Elston (1993).

To calculate $\hat{f}_{imj}$ for sib pairs, we use equation 8.1, while for other pair types we use equation 8.2. For sib pairs, we use the *counting* method suggested by Amos, Dawson and Elston (1990). To evaluate equation 8.2 for other pair types, we condition upon a set of individuals in the pedigree that includes the pair and a chain of individuals connecting the pair genetically. This chain includes the parents of each member of the pair and the parents shared by any two individuals already in the chain [See Amos, Dawson and Elston (1990) for more detail.] We know that

$$P\left(I_m, pair\ j\ shares\ i\ alleles\ IBD\right) = \sum_{g \in G} P\left(I_m,\ pair\ j\ shares\ i\ alleles\ IBD,\ g\right),$$

where G is the set of all possible genotype configurations of the individuals in the conditioned set. We therefore calculate L $(I_m,$ pair j shares i alleles i.b.d.,$g)$ for each possible genotype configuration in G. We use the recursive methods of Fernando, Stricker and Elston (1993) to calculate the likelihood for the sections of the pedigree not in the conditioned set and reuse them for each likelihood calculation.

8.2.2 Exact IBD Analysis

The exact IBD analysis is used for both single- and multi-point analysis. It uses the exact multi-point algorithm to generate likelihoods of inheritance vectors at target locations. These likelihoods are then summed separately for inheritance vectors corresponding to a given pair sharing 0, 1, and 2 alleles IBD.

8.2.2.1 The Exact Multi-point Algorithm

The general algorithm used by MLOD and GENIBD to generate multi-point likelihoods and other statistics is called the exact multi-point algorithm. This algorithm takes a chromosomal region and generates likelihoods of all the possible inheritance patterns at each marker in the region. These likelihoods can then be combined to generate identity-by-descent statistics.

8.2.2.2 Single-point IBD Sharing

For single-point, a likelihood vector is generated for each marker of interest. For each inheritance pattern, the number of alleles shared by a given inheritance pattern can be determined by tracking which founder alleles each pair of individuals receives. By summing the likelihoods of all inheritance patterns that share a specific number of alleles IBD, and dividing by the total likelihood of the pedigree at that marker (equation 8.2 above), we obtain the probability of the pair sharing that number of alleles IBD.

8.2.2.3 Multi-Point IBD Sharing

The multi-point algorithm is essentially the same as single-point. For each location of interest along the chromosome, we generate a multi-point likelihood vector incorporating all the information provided by the markers. This vector can then be summed, as in the single-point case above, to give us the multi-point probability of sharing 0, 1 and 2 alleles IBD.

8.2.3 Simulation IBD Analysis

The simulation IBD analysis uses a modified Sobel and Lange (Sobel and Lange, 1996) algorithm to generate random inheritance patterns at each marker in the state space. A multi-point likelihood for all markers is generated, assuming no crossover interference. For each generated state, IBD values are noted. Heuristic methods are used to determine the number of states to be generated, as well as the number of batches and how much dememorization to perform.

8.2.3.1 Calculating the Amount of Simulation

By default, GENIBD determines the amount of simulation to perform for each pedigree. It does this by multiplying the number of individuals in the pedigree by the number of markers in the region being simulated. This number is then multiplied by several factors, one each for the number of dememorization steps per batch, the number of simulation steps per batch, and the number of batches. The default factors have been set, based upon extensive in-house testing, to the following:

dememorization steps per batch	15
simulation steps per batch	150
batch factor	30

These values have been found to be sufficient in most cases, but may be changed.

8.3 Program Input

File Type	Description
GENIBD Parameter File	Specifies the parameters and options with which to perform a particular analysis.
Pedigree Data File	Contains delimited records for each individual including fields for identifiers, sex, parents, trait and marker data.
Marker Locus Description File	Lists the alleles, allele frequencies and phenotype to genotype mapping for each marker locus.
Genome Description File	Contains a description of the linked marker regions, including distances between markers. This file is not required for single-point analysis.

8.3.1 The genibd Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main GENIBD parameter.

parameter [, attribute]	**Explanation**	
genibd	Starts GENIBD analysis block.	
	Value Range	N/A
	Default Value	N/A
	Required	Yes
	Applicable Notes	None
, out	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	None
	Required	No
	Applicable Notes	None

Notes

The out attribute controls the filenames generated by the analysis. For each region in the analysis, a separate IBD file is generated. These filenames are in the format: "*out_region*.ibd" where *out* is the value of the out attribute, and *region* is the region name. If no out attribute is specified, the analysis title is used instead.

8.3.2 The genibd Block

The following syntax table specifies the permissible parameter and attribute settings for the genibd block.

parameter [, attribute]	**Explanation**	
title	Specifies title of the run	
	Value Range	Character string.
	Default Value	None
	Required	Yes
	Applicable Notes	None

`region`	A region to be analyzed. As many region parameters may be specified as required to specify which regions are to be analyzed. **If no region parameters are specified, all regions in the genome description file are analyzed.** Regions with no valid markers (as may be the case when data for only one or two chromosomes are present in the pedigree data file) are skipped.
	Value Range: Character string.
	Default Value: None
	Required: No
	Applicable Notes: None
`output_pair_types`	Specifies pair types to be generated
	Value Range: siblings, all_sibs, relatives
	Default Value: all_sibs
	Required: No
	Applicable Notes: 1
`max_pedigree`	The largest $2n - f$ value to be processed for a pedigree while performing exact multi-point or single-point analysis.
	Value Range: { 1, 2, 3, ...}
	Default Value: 18
	Required: No
	Applicable Notes: None
`scan_type`	Indicates whether to compute IBD sharing at the observed markers or at the markers and intervals between them.
	Value Range: {**markers**, **intervals**}
	Default Value: **markers**
	Required: No
	Applicable Notes: None
`,distance`	Sets the interval, in cM, to use as basis for computing IBD sharing probabilities between observed markers. Only applicable when value of `scan_type` parameter is set to **intervals**.
	Value Range: [0, +INF]
	Default Value: 2.0
	Required: No
	Applicable Notes: None
`allow_loops`	Allows pedigrees with loops to be processed while performing single-point analysis.
	Value Range: {**true**, **false**}
	Default Value: **false**
	Required: No
	Applicable Notes: None

`ibd_mode`	Selects either single- or multi-point IBD generation. Value Range: {**singlepoint**, **multipoint**} Default Value: **multipoint** Required: No Applicable Notes: 2
`split_pedigrees`	Option to allow pedigrees that are too large for the exact analysis to be split into nuclear families before processing. Setting the value to **always** means that all pedigrees will be split in this fashion. Value Range: {**yes**, **no**, **always**} Default Value: **no** Required: No Applicable Notes: None
`simulation` `use_simulation`	Starts a sub-block for specifying simulation options. Value Range: {**yes**, **no**, **always**} Default Value: **yes** Required: No Applicable Notes: 3

Notes

1. `output_pair_types` may be set to any of three values: **siblings** if only full sibling pairs are desired, **all_sibs** if both full and half sibling pairs, or **relatives** if all five relative pair types (sibs, half sibs, avuncular, grand-parental and cousin) are desired.

2. If **singlepoint** is selected, only data at each marker are used to calculate the IBD sharing at a marker. If **multipoint** is selected, all the marker data in a region is used to calculate the IBD sharing at each point, assuming no interference.

3. The `simulation` sub-block allows simulation on pedigrees that are too large for the exact methods. Setting the value of this parameter to **always** means that all pedigrees will use simulation. For example:

```
genibd, out = autism_study_01 {
   title      = “Autism Study #1: IBD Results”
   region     = “Chrom1”
   ibd_mode   = multipoint
   scan_type  = intervals, distance = 1.0
   simulation = always {
      use_factoring = true
      sim_steps = 100000
   }
}
```

8.3.3 Sub-Block Syntax: `simulation`

The following syntax table specifies the permissible parameter and attribute settings for the `simulation` sub-block.

parameter [, attribute]	**Explanation**
`sim_local_marker`	The proportion of times during simulation that a marker adjacent to the current marker being simulated is chosen for simulation during the next simulation step. Value Range: [0, 1] Default Value: 0.75 Required: No Applicable Notes: None
`use_factoring`	Controls whether the simulation scaling factors are used. If they are not used, simulation uses a constant number of steps regardless of pedigree size. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 1
`base_factor`	The base scaling factor provides a method of adjusting all three scaling factors together. Will be ignored if `use_factoring` is set to **false**. Value Range: $[0, \infty)$ Default Value: None Required: No Applicable Notes: None
`dememorization_factor`	The dememorization scaling factor. This controls the number of dememorization steps done during each batch. Will be ignored if `use_factoring` is set to **false**. Will be set to 0.5 x `base_factor` if `base_factor` > 0. Value Range: {1, 2, 3, ...} Default Value: 15 Required: No Applicable Notes: 1
`sim_factor`	The simulation step scaling factor. This controls the number of simulation steps during each batch. Will be ignored if `use_factoring` is set to **false**. Will be set to 10 x `base_factor` if `base_factor` > 0. Value Range: {1, 2, 3, ...} Default Value: 150 Required: No Applicable Notes: 1

`sim_batch_factor`	The simulation batch count scaling factor. This controls the number of batches of simulation to perform. Will be ignored if `use_factoring` is set to **false**. Will be set to `base_factor` if `base_factor` > 0.
	Value Range: {1, 2, 3, ...}
	Default Value: 30
	Required: No
	Applicable Notes: 1
`sim_steps`	The number of simulation steps during each batch.
	Value Range: {1, 2, 3, ...}
	Default Value: 200000
	Required: No
	Applicable Notes: 1
`dememsteps`	The number of dememorization steps during each batch.
	Value Range: {1, 2, 3, ...}
	Default Value: 50000
	Required: No
	Applicable Notes: 1
`batch_count`	The number of batches of simulation to perform.
	Value Range: {1, 2, 3, ...}
	Default Value: 100
	Required: No
	Applicable Notes: 1

Notes

1. When calculating identity-by-descent values by simulation, it is usually unnecessary to specify the amount of simulation to be performed. GENIBD does this automatically for each pedigree being analyzed. However, an option to specify the amount of simulation is provided. There are two methods of doing this:

   - The first, called *factoring*, calculates the amount of dememorization, the amount of simulation, and the number of batches based upon pedigree size and number of markers in the region being simulated. It is selected by setting `use_factoring` to **true** (default). The user may set the value of `base_factor` (which automatically determines the values of `dememfactor`, `sim_factor` and `sim_batch_factor` as described in the syntax table) or may set the values of `dememfactor`, `sim_factor` and `sim_batch_factor` directly.
   - The second method uses the same number of steps and batches for every pedigree. It is used when `use_factoring` is set to **false**. Setting `dememsteps`, `sim_steps`, and `batch_count` parameters sets, respectively, the amount of dememorization per batch, the amount of simulation per batch, and the number of batches.

8.4 Program Execution

GENIBD is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running GENIBD from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line including any that are optional.

```
>GENIBD
S.A.G.E. v5.x -- GENIBD
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: ./GENIBD <parameters> <pedigree> <locus> [map]
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
locus - locus description file
map - Genome Map File (optional for single point analysis)
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of GENIBD may look like the following:

```
>GENIBD data.par data.ped data.loc data.map
S.A.G.E. v5.x -- GENIBD
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Loading Map file...
Validating Analysis...
ANALYSES
================================================================================
Single-point : Singlepoint analysis on region (CHR5):
* - Pedigrees with loops will not be processed.
Multi-point : Multipoint analysis on region (CHR5):
* - Maximum size for exact method is set to 18.
* - Pedigrees larger than 18 will be simulated.
* - Pedigrees will be simulated based on pedigree size and the number
of markers.
Processing Analyses....
================================================================================
Single-point : Singlepoint analysis on region (CHR5):
* - Pedigrees with loops will not be processed.
Processing Region: CHR5
================================================================================
Single-point: Pedigree 1
Generating Single Point Likelihoods...................Done.
Single-point: Pedigree 10
Generating Single Point Likelihoods...................Done.
Single-point: Pedigree 100
Generating Single Point Likelihoods...................Done.
.
.
.
```

8.5 Program Output

GENIBD produces several output files that contain results and diagnostic information:

File Name	File Type	Description
GENIBD.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
genome.inf	Genome Information File	Contains diagnostic information on the genetic map data and the marker loci that were provided for analysis. No analysis results are stored in this file.
analysis_region.ibd	IBD sharing files	There is one IBD sharing file for each region processed by each analysis performed by GENIBD. These files contain the IBD distribution of each pair of relatives for each marker in the analysis (see 2.7).

8.5.1 Information Output File

The GENIBD information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read in, are included with all programs in S.A.G.E. and are very useful for debugging many common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not run as expected. The file "`genibd.inf`" should be checked for errors and diagnostic information after each run of the program.

8.5.2 Genome Information File

This file includes warnings and errors produced while parsing the marker locus description file, as well as a table for each marker listing allele and genotype population frequencies, assuming Hardy-Weinberg equilibrium. If allele frequencies do not sum to 1.0, they are standardized to 1.0, so these frequencies may not be identical to those in the marker locus description files.

8.5.3 IBD Sharing Files

The IBD sharing file stores the IBD probability distribution of allele-sharing identical-by-descent between pairs of individuals at specific locations.

8.6 Example Output File

The IBD sharing file is generated as output from GENIBD and is used as input to other programs, such as SIBPAL. It contains the following information (see 2.7):

- a list of the markers at which the IBD sharing distributions are generated.
- a table that contains a line for each relative pair and the probabilities of sharing 0 or 2 alleles at each marker (designated as f_0 and f_2, respectively). Additionally, the table shows the value of the difference $f_{1m} - f_{1p}$ to support analysis of parent-of-origin effects. The table includes up to five types of relative pairs: sibling, half sibling, avuncular, grand-parental and cousin.

In our example, two IBD sharing files are generated, one using single- and one using multi-point analysis. Although the numerical results are generally different, the files are similar in structure. The following is a portion of the single-point file:

```
IBD File 1.9 : This File is automatically generated.  Do NOT edit!
#=====================================================================
#
ANALYSIS
#------
title           = Analysis 1
region          = 11
max_pedigree    = 18
scan_type       = intervals
allow_loops     = off
ibd_mode        = multipoint, exact
split_pedigrees = no
use_simulation  = no
#
MARKERS
#------
11s1984  0.0
11_2.0   2.0
11_4.0   4.0
11s2362  6.0
11_8.0   8.0
11_10.0 10.0
11_12.0 12.0
11s1999 14.0
.
.
.
#=====================================================================
#Pedigree   Ind 1   Ind 2        11s1984 f0,   11s1984 f1m-f1p,        11s1984 f2,  ...
#--------  ------  ------  -----------------  -----------------  -----------------
     105,      6,      7,  0.000000000000000  1.000000000000000  0.000000000000000  ...
     106,      3,      4,  0.063250723689565  0.936749276310435  0.000000000000000  ...
     106,      3,      5,  0.000000000000000  0.000000000000000  1.000000000000000  ...
     106,      4,      5,  0.063250723689565  0.936749276310435  0.000000000000000  ...
     107,      3,      4,  0.005884636962262  -0.00000000000000  0.000000000000000  ...
     107,      7,      1,  0.500000000000000  ----------------   0.000000000000000  ...
     107,      7,      2,  0.500000000000000  ----------------   0.000000000000000  ...
     107,      7,      3,  1.000000000000000  ----------------   0.000000000000000  ...
     107,      7,      8,  0.000000000000000  1.000000000000000  0.000000000000000  ...
     107,      8,      1,  0.500000000000000  ----------------   0.000000000000000  ...
     107,      8,      2,  0.500000000000000  ----------------   0.000000000000000  ...
     107,      8,      3,  1.000000000000000  ----------------   0.000000000000000  ...
       .       .       .                  .                  .                  .  ...
       .       .       .                  .                  .                  .  ...
       .       .       .                  .                  .                  .  ...
```

Chapter 9

RELTEST

RELTEST helps classify pairs in a (sib pair) linkage study according to their true relationship using genome scan data. It is based on a Markov process model of allele-sharing along chromosomes. The program currently performs analyses to classify putative sib pairs, putative half-sib pairs, putative parent-offspring pairs, and putative marital pairs into five different types of pairs: MZ twin pairs, full sib pairs, half sib pairs, parent-offspring pairs, and unrelated pairs. A summary file is produced that contains the identifiers of the putative full-sib pairs to be reclassified and their sibling classification statistics, parent offspring classification statistics; for each pair, missing data rates over the genome; and histograms of the sibling classification statistic and parent offspring classification statistic. An optional output file contains the same pair-specific statistics, but for all putative pairs other than MZ twins (i.e., including putative half-sib pairs, parent-offspring pairs and unrelated pairs).

9.1 Limitations

The probability of misclassification depends on the total length of the genotyped genome provided and overall marker informativeness. The misclassification rates are minimal when at least half the genome is genotyped using microsatellite markers at most 20 cM apart. Individual pairs may be misclassified if one or both members have a high proportion of missing genotypes, as the classification cut points are based on the length of the genotyped genome and marker informativeness calculated for the entire sample. It should also be noted that the proportion of missing genotypes is calculated using as the denominator the number of markers listed in the genome file.

9.2 Theory

This program is intended primarily for late-onset diseases, for which parents are not typed and the number of typed sibs is often two. In this case, one cannot detect errors in relationship by looking for inconsistencies, and one must use the entire genome (or as much of it as possible) to examine the overall allele-sharing between the sibs. In practice, this program can be used for other types of data sets, and even pairs with late-onset disease will sometimes have typed parents or additional sibs. However, we do not use all the marker information to construct the relationship statistics. For each pair, only the marker information for that pair is used, and none from the other relatives, including other sibs and parents. Pair-wise allele-sharing is computed using multipoint algorithms.

9.2.1 Full Sib Pairs

Let $\hat{f}_{jis}$be the estimated probability that sib pair j shares i marker alleles identical-by-descent (IBD) at location s on a chromosome. We assume throughout that these IBD probabilities are obtained using multi-point methods. Feingold et al. (1993) proposed a Gaussian process model to describe the ideal (i.e., infinitely dense, fully informative) process for the estimated mean number of alleles shared IBD by a sample of N sib pairs at location s:

$$X_s = \sum_{j=1}^{N} (\hat{f}_{j1s} + 2\hat{f}_{j2s}).$$

If the marker is fully informative, X_s is the total number of alleles shared IBD in the sample at location s.

For the ideal process and a large sample of randomly sampled sib pairs, the mean-sharing statistic

$$Z_s = (X_s - N) \Big/ (N/2)^{1/2}$$

has mean equal to 0, variance equal to 1, and approximate Gaussian process covariance function exp(-$\beta|t|$), where t is the distance between markers and β=0.04 for sib pairs (Feingold et al., 1993). The parameter β is a function of the recombination process and assumes that crossovers are independent, i.e., that there is no crossover interference.

Here we consider a single random sib pair j, and let Z_{js} be the mean-sharing statistic for a single pair (N=1). We obtain a measure of the average number of alleles shared by this pair over the entire genome. Let k=1,2,...,22 index the human autosomes and L_k be the length of chromosome k in cM. The statistic

$$Y_{jk} = \frac{1}{L_k} \int_0^{L_k} Z_{js} ds$$

has expectation

$$E(Y_{jk}) = \frac{1}{L_k} \int_0^{L_k} E(Z_{js}) ds \; = \; 0$$

and variance

$$Var(Y_{jk}) = \frac{1}{L_k^2} \int_0^{L_k} \int_0^{L_k} Cov(Z_{js}, Z_{jr}) dr ds = \frac{2}{\beta L_k} - \frac{2}{(\beta L_k)^2}(1 - e^{-\beta L_k}) \qquad (9.1)$$

(Parzen, 1962; Olson, 1999). In the ideal case of fully informative, infinitely dense markers, the statistic Y_{jk} is the difference between the proportions of the chromosomes sharing 2 and 0 alleles IBD. More generally, it is the difference between the absolute areas above and below the null mean (sharing 1 allele IBD), divided by the length of the chromosome.

If putative sib pair j is a true sib pair, then Y_{jk}/[Var(Y_{jk})]$^{1/2}$ has a standard normal distribution as $L_k \to \infty$. In practice, the normal approximation is somewhat inadequate for single chromosomes of modest length. A genome-wide measure, the sibling classification statistic given by

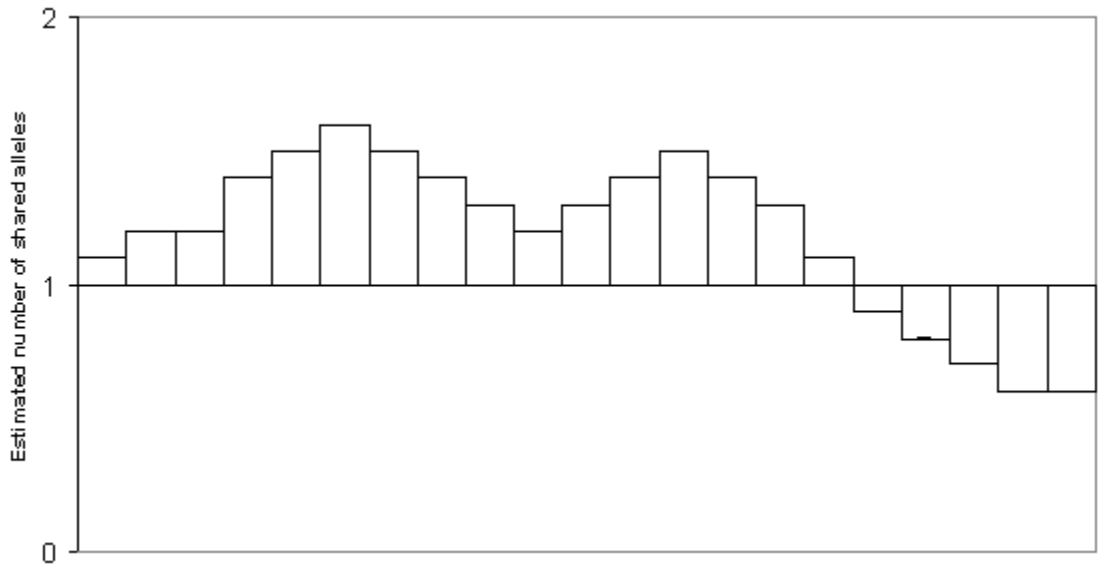


Figure 9.1: Approximate Mean-Corrected Allele Sharing

$$Y_j = (\sum_{k=1}^{22} Y_{jk}) \Big/ [\sum_{k=1}^{22} Var(Y_{jk})]^{1/2},$$

is well approximated by a standard normal distribution in the fully informative, infinitely dense case. Similarly, for any number of chromosomes K,

$$Y_j = (\sum_{k=1}^{K} Y_{jk}) \Big/ [\sum_{k=1}^{K} Var(Y_{jk})]^{1/2}.$$

Relationship estimation for each pair j in the sample is based on estimating genome-wide Y_j for each of the sib pairs. These statistics can be obtained in practice using a standard algorithm to calculate multipoint IBD at equally spaced points throughout the genome. For each chromosome, the absolute areas above and below the estimated mean-corrected allele-sharing curve is approximated using rectangles (see Figure 9.1),

which is equivalent to computing:

$$\hat{Y}_{jk} = [c\sqrt{2} \sum_{s=1}^{P} (X_{sj} - 1)]/P,$$

where P is the number of points at which allele-sharing is computed and c is the distance (cM) between points.

9.2.2 Parent/Offspring Pairs

Parent/offspring pairs are always expected to share exactly one allele IBD, and so $\hat{Y}_j$ cannot be used to discriminate between sib pairs and parent/offspring pairs. Therefore, a second Markov process statistic is used to classify sibs vs. parent/offspring pairs. At location s, the estimated number of alleles shared IBD by a parent/offspring pair is obtained using

$$X_s^* = (\hat{f}_{j2s} + \hat{f}_{j0s} - \hat{f}_{j1s}).$$

For a fully informative location s, the Gaussian process statistic

$$Z_s^* = \sum_{j=1}^{N} \frac{X_s^*}{N^{1/2}}$$

has a standard normal distribution in a large sample of sib pairs, with covariance function exp(-$\beta|t|$), where now β = 0.08. The new statistic Y_j^*, the parent offspring classification statistic, is calculated in the same manner as before, i.e.,

$$Y_j^* = (\sum_{k=1}^{K} Y_{jk}^*) \Big/ [\sum_{k=1}^{K} Var(Y_{jk}^*)]^{1/2},$$

with

$$\hat{Y}_{jk} = [c \sum_{s=1}^{P} X_{sj}^*]/P,$$

and the variance is calculated using equation 9.1 with β= 0.08.

9.2.3 Incomplete Marker Information

When markers are not infinitely dense and fully informative, the variance of the Sibling and Parent-Offspring Classification Statistics are less than one. Classification criteria (cut points) may be determined using the overall marker informativity and the length of the genotyped genome. The *Average Marker Information Content* (AMIC) (Kruglyak and Lander 1995)

is defined as follows,

$$AMIC = \sum_{p=1}^{M} r^2(s)/M,$$

where M is the total number of points over which the genome IBD probabilities are calculated and

$$r^2(s) = 1 - \frac{\sum_{i=1}^{N} \sigma^2_{i,residual}(s)}{\sum_{i=1}^{N} \sigma^2_{i,initial}} = 1 - \frac{2\sum_{i=1}^{N} \sigma^2_{i,residual}(s)}{N},$$

N is total number of sib pairs in the sample and $\sigma^2_{i,residual}(s)$ is the variance of the IBD distribution at point s for sib pair i.

The best-fit regression equations for obtaining classification values, the cut points, are:

- $\log_{10}(-C_u) = 0.421 + 0.506\log_{10}(T) + 1.162\log_{10}(AMIC) + 0.472(\log_{10}(AMIC))^2$,
- $\log_{10}(-C_h) = 0.141 + 0.524\log_{10}(T) + 0.237\log_{10}(AMIC) - 0.861(\log_{10}(AMIC))^2$,
- $\log_{10}(-C_p) = 0.2 + 0.518\log_{10}(T) + 2.220\log_{10}(AMIC)$,
- $C_m = 3.27$,

where T is the total length of the genotyped genome in cM divided by 150, and C_u, C_h, C_m, and C_p are the classification cut points for unrelated pairs, half sib pairs, MZ twins, and parent offspring pairs respectively. C_u, C_h, C_m are used to classify pairs on the basis of the sibling classification statistic into unrelated, half sibs, full sibs, and MZ twins. C_p is used to classify pairs into full sib and parent-offspring pairs on the basis of the parent-offspring classification statistic.

9.2.4 Strategy for Classifying Putative Full-Sib and Non-Full-Sib Pairs

There are two steps to classify each pair:

1. Using Y_j and the cut points defined above, we classify as follows:
   Unrelated $< C_u <$ Half sib $< C_h <$ Sib $< C_m <$ MZtwin
2. If the pair is classified as a sib pair in step 1, we use Y_j^* and the parent_offspring cut point:
   Parent/offspring $< C_p <$ Sib

9.2.5 Nonparametric Estimation Procedure

After calculating the Y_j and Y_j^*, a nonparametric estimation procedure is used to obtain the mean and variance of the sib-pair distributions of these two sets of statistics.

1. Estimating the shift:
   We use the L_2-error procedure (Scott, 2000) to maximize the function
   $$\frac{2}{n}\sum_{j=1}^{n}\phi(Y_j|\mu,\sigma^2) - \frac{1}{2\sqrt{\pi\sigma^2}},$$
   where μ and σ^2 are parameters, n is the total number of sib pairs (all putative full sib pairs), and ϕ(.) is the normal density function
   $$\phi(Y_j|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}(Y_j-\mu)^2}.$$

2. We then adjust the cut points:

   New Cut point = Old Cut point + μ from step 1.

3. We repeat the same step 1 and 2 for the Y_j^* obtained from all putative full sib pairs.

4. We perform the classification as described in 9.2.4 using the new cut points.

   To test the deviation of the sib pair mean from zero, we use the Y_j from putative full sib pairs now classified as true sib pairs to compute the mean

$$\bar{Y} = \frac{\sum_{j=1}^{n} Y_j}{n}$$

   and the standard error of the mean

$$S.E.(\bar{Y}) = \frac{1}{\sqrt{n}} \sqrt{(\sum_{j=1}^{n} Y_j^2 - \frac{(\sum_{j=1}^{n} Y_j)^2}{n})/n} \quad .$$

   Then a confidence interval is constructed as

$$\bar{Y} \pm 2S.E.(\bar{Y}).$$

If zero is not included in this interval, a warning is printed in the output. The user should at this point note that the sib-pair histogram is shifted significantly (in the statistical sense) away from its null hypothesis mean value of zero. If such significant deviation is substantial, there may be large-scale error in the data or specification of parameters. Our previous experience with real data sets has shown that such error may be due to

1. Gross misspecification of marker allele frequencies,
2. Misalignment of marker description information between the parameter file, the data file and/or the genome file, and
3. Large-scale genotype errors.

Examples of large-scale genotype errors that have caused large "shifts" in the sib-pair histogram have included:

1. Errors in programs translating data from the genotyping lab to the pedigree data file and
2. Extensive binning errors in the assignment of genotypes.

The above list includes only errors we have been alerted to by RELTEST; other sources of error detectable by RELTEST are clearly possible. We suggest using RELTEST not only to classify pairs according to relationships, but also as a general test of the overall accuracy of the data and parameter specifications (Olson et al., 2004).

9.3 Program Input

File Type	Description
Parameter file	Specifies the parameters and options with which to perform an analysis.
Pedigree data file	Contains delimited records for each individual including fields for identifiers, sex, parents, and marker data.
Marker locus description file	Lists the alleles, allele frequencies and phenotype to genotype mapping for each marker locus.
Genome description file	Contains a description of the linked marker regions, including distances between markers.

9.3.1 Parameter File Syntax

RELTEST can read multiple pedigree data files in cases where each pedigree file contains the markers for a single chromosome. For each pedigree file, there has to be a corresponding `pedigree` block with `file` name specified. All other fields should be the same, except for the marker fields, for all pedigree files used.

Example:

```
pedigree, file=ped1
{
.
.
.
marker=”ch1m1”
.
.
.
}
pedigree, file=ped2
{
.
.
.
marker=”ch2m1”
.
.
.
}
pedigree, file=ped3
{
.
.
.
```

```
marker="ch3m1"
.
.
.
}
```

The specific syntax for RELTEST parameters, attributes and values is described in the following sections.

9.3.1.1 The `reltest` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main RELTEST parameter.

`parameter [, attribute]`	**Explanation**
`reltest`	Starts a RELTEST parameter block. Value Range: N/A Default Value: N/A Required: Yes Applicable Notes: None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension. Value Range: Character string representing a valid file name. Default Value: reltest Required: No Applicable Notes: None

9.3.1.2 The `reltest` Block

The following syntax table specifies the permissible parameter and attribute settings for the `reltest` block.

`parameter [, attribute]`	**Explanation**	
`pair_type`	Specifies the putative pair to be analyzed.	
	Value Range	sib hsib parent_offspring marital
	Default Value	None
	Required	No
	Applicable Notes	1
`region`	Specifies the genomic regions that will be used in the analysis.	
	Value Range	Character string representing the name of a region in the genome description file. If no region is specified then analysis will take place with respect to all available marker data.
	Default Value	None
	Required	No
	Applicable Notes	None
`cut_points`	Specifies pre-calculated cut points to be used to classify pairs.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	2
`, sib`	Cut point for sibling pairs.	
	Value Range	(-∞, ∞)
	Default Value	None
	Required	No
	Applicable Notes	2
`, hsib`	Cut point for half-sib pairs.	
	Value Range	(-∞, ∞)
	Default Value	None
	Required	No
	Applicable Notes	2
`, MZtwin`	Cut point for MZtwins	
	Value Range	(-∞, ∞)
	Default Value	None
	Required	No
	Applicable Notes	2

, parent_offspring	Cut point for parent-offspring pairs. Value Range: (-∞, ∞) Default Value: None Required: No Applicable Notes: 2
nucfam_file	Specifies option to print out the sibling in the nuclear family file. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 3
detailed_file	Specifies option to print detailed output file. Value Range: {true, true} Default Value: false Required: No Applicable Notes: None

Notes

1. By default, all four pair types will be analyzed.
2. Normally, cut points are automatically generated based on the pedigree data, as given in the theory section. The program will use the generated cut points if the `cut_points` parameter is not specified here.

   Example: All of the following are valid RELTEST analysis statements:

```
reltest

reltest, out=test { # generate summary output file named “test.sum”
                    # and nuclear family output file named “test.fam”
   nucfam = true
   pair_type = sib # do calculations for putative sibs
   pair_type = hsib # do calculations for putative half sibs
}

reltest {
   detailed=true
   region="Chr1"
   region="Chr2"
   region="Chr42" # Do analysis using only Chromosomes 1, 2, and 42
}
```

3. See 9.5.3

9.4 Program Execution

RELTEST is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running RELTEST from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>reltest
S.A.G.E. v5.x -- RELTEST
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: ./reltest <parameters> <pedigree> <locus>
<map>
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
locus - locus description file
map - Genome Map File
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of RELTEST may look like the following:

```
>reltest reltest.par example.ped example.loc example.map
S.A.G.E. v5.x -- RELTEST
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading parameter file.....................done.
Reading locus description file............done.
Reading pedigree file.....................
from example.ped............done.
Reading genome file.......................done.
RELTEST analysis....... 1
Testing sib pairs.........................done
Testing parent_offspring pairs............done
Testing mother_father pairs...............done
Analysis complete!
```

9.5 Program Output

RELTEST produces several output files that contain results and diagnostic information:

Filename	File Type	Description
reltest.inf	Information output file	Contains informational diagnostic messages, warnings and program errors.
reltest.sum	Reclassification summary file	Contains the values of all cut points and pairs to be reclassified, together with related statistics. Contains histograms and classification statistics for all putative pairs.
reltest.fam	Sibling in nuclear family information file	Contains information about all sib pairs of the nuclear families in which at least one sib pair should be reclassified.
reltest.det	Detailed pair information file	Contains statistics for all pairs used in the analysis.

9.5.1 Information Output File

The RELTEST information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables provide information about what the program has read in, and are included with all programs in S.A.G.E. They are very useful for debugging many common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. The program attempts to correct many common errors by making reasonable assumptions that are usually sufficient, but may not be for your data set. It is strongly recommended that you check this file for warning and error messages before examining the results of any run of the program. The file "reltest.inf" should be checked for errors and diagnostic information after each run of the program.

9.5.2 Reclassification Summary File

The reclassification summary file contains the cut point values to classify pairs and the total length of genome used in the analysis. It also provides a separate table for each putative pair type, listing pairs to be reclassified with their individual IDs and pedigree IDs from the original pedigree data file, new class, sibling classification statistic, parent offspring classification statistic and missing genotype rate. Note: misclassification may occur if one or both members of the pair have a high rate of missing genotypes. For each putative pair type, the total number of original pairs and the total number of pairs to be reclassified are also included.

This file also provides text-based histograms of the sibling classification statistic and the parent offspring classification statistic for each putative pair type included in the analyses. The minimum and maximum values of these statistics are also included.

9.5.3 Sibling in Nuclear Family Information File

The sibling-in-nuclear-family information file contains information about all sib pairs in nuclear families in which at least one sib pair should be reclassified. This file provides heuristic information intended to aid understanding the statistical distribution related to pairs that should be reclassified.

9.5.4 Detailed Pair Information File

This file provides a table of the Y_j and Y_j^* values for all pairs used in the analysis for each putative pair type.

9.6 Example Output Files

9.6.1 Reclassification Summary File

Here is a typical example of a RELTEST summary output file:

```
S.A.G.E. v5.x -- RELTEST
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
================================================================================
  RELATIONSHIP TEST PROGRAM SUMMARY OUTPUT
================================================================================
    Analysis Name                      : default_analysis

    Average Marker Information Content : 0.61057
    Total Length of Genome             : 3539 (cM)

    Cut-points
                                    |           |  original       adjusted
      --------------------------------|-----------|-------------------------
      Sibling Classification        |unrelated  | -7.73259       -6.56278
        Statistics(Yj)              |half sib   | -3.07633       -1.90652
                                    |MZtwins    |  3.27000        4.43981
      --------------------------------|-----------|-------------------------
      Parent/Offspring Classification |parent/    |
        Statistics(Yj*)             | offspring | -2.72544       -3.22814
      ------------------------------------------------------------------------

    Sibling Classification Statistics(Yj)
       robust (L2) mean     : 1.16981
       robust (L2) variance : 0.5
    Parent/Offspring Classification Statistics(Yj*)
       robust (L2) mean     : -0.502698
       robust (L2) variance : 0.5

    Average Yj of Pairs
      Reclassified as Full Sibs : 1.18377
    Standard Error              : 0.0477349
    95% Confidence Interval     : 1.0883 to 1.27924

  ! WARNING : THE MEAN OF THE SIB-PAIR DISTRIBUTION DIFFERS SIGNIFICANTLY FROM
              ZERO.  YOU MAY HAVE SUBSTANTIAL DATA ERROR OR MISSPECIFICATION OF
              PARAMETERS SUCH AS ALLELE FREQUENCIES.

================================================================================

PUTATIVE FULL SIB PAIRS TO BE RECLASSIFIED :

                    reclassified
  pid      pair     pair type         Yj          Yj*          missing data
--------------------------------------------------------------------------------
  118      3/4      HSIB            -2.2236     -1.6296          4% /  5%
  159      3/5      HSIB            -4.2079      0.0942          8% /  8%
    4      3/4      HSIB            -2.8302     -1.6862          2% /  1%
   45      3/4      HSIB            -3.0415     -0.6532          3% /  5%
   58      5/6      HSIB            -3.1239     -1.6834          3% /  4%
   60      3/5      HSIB            -2.7622     -1.0231          3% /  4%
   66     30/31     HSIB            -2.7980     -1.1299          5% /  5%
   66     12/16     MZTWINS          8.3388      7.1082         30% /  4%
--------------------------------------------------------------------------------
 total putative  pairs          : 342
 total pairs to be reclassified : 8
.
.
.
```

```
  ============================================================
  ==                                                        ==
  ==   HISTOGRAM OF SIBLING CLASSIFICATION STATISTIC (Yj)   ==
  ==                FOR PUTATIVE PAIRS                      ==
  ==                                                        ==
  ==           putative pair type : FULL SIB                ==
  ==                  maximum Yj  :    8.33881             ==
  ==                  minimum Yj  :   -4.20787             ==
  ==                     bin size :       0.25             ==
  ==                                                        ==
  ============================================================


      Interval              count (one * is equal to 1 or 2 pairs.)
=========================================================================
  -4.22  to  -3.97            1 *
  -3.97  to  -3.72            0
  -3.72  to  -3.47            0
  -3.47  to  -3.22            0
  -3.22  to  -2.97            2 *
  -2.97  to  -2.72            3 **
  -2.72  to  -2.47            0
  -2.47  to  -2.22            1 *
  -2.22  to  -1.97            0
  -1.97  to  -1.72            0
  -1.72  to  -1.47            1 *
  -1.47  to  -1.22            1 *
  -1.22  to  -0.97            1 *
  -0.97  to  -0.72            2 *
  -0.72  to  -0.47            2 *
  -0.47  to  -0.22            8 ****
  -0.22  to   0.03           13 *******
   0.03  to   0.28           22 ***********
   0.28  to   0.53           27 **************
   0.53  to   0.78           26 *************
   0.78  to   1.03           42 *********************
   1.03  to   1.28           43 **********************
   1.28  to   1.53           34 *****************
   1.53  to   1.78           36 ******************
   1.78  to   2.03           21 ***********
   2.03  to   2.28           16 ********
   2.28  to   2.53           18 *********
   2.53  to   2.78           11 ******
   2.78  to   3.03            5 ***
   3.03  to   3.28            3 **
   3.28  to   3.53            0
   3.53  to   3.78            0
   3.78  to   4.03            1 *
   4.03  to   4.28            1 *
   4.28  to   4.53            0
   4.53  to   4.78            0
.
.
.
```

9.6.2 Sibling in Nuclear Family Information File

Here is a typical example of a sibling in nuclear family information file:

```
S.A.G.E. v5.x -- RELTEST
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
================================================================================
  RELATIONSHIP TEST PROGRAM NUCLEAR FAMILY INFORMATION
================================================================================

    Note          : This file contains information about all sib pairs of the
                    nuclear families in which at least one sib pair has been
                    reclassified.

    Analysis Name : analysis_1

================================================================================

                   reclassified
  pid      pair     pair type         Yj          Yj*          missing data
--------------------------------------------------------------------------------
  118      3/4      HSIB           -2.2236     -1.6296          4% /  5%
  118      7/8      SIB             0.4039     -1.0336          4% /  3%
  159      3/5      HSIB           -4.2079      0.0942          8% /  8%
  159      7/8      SIB             1.5395     -0.2638          7% /  7%
    4      3/4      HSIB           -2.8302     -1.6862          2% /  1%
   45      3/4      HSIB           -3.0415     -0.6532          3% /  5%
   58      5/6      HSIB           -3.1239     -1.6834          3% /  4%
   58     14/15     SIB             2.0909     -0.6428          2% /  2%
   60      3/5      HSIB           -2.7622     -1.0231          3% /  4%
================================================================================
```

Chapter 10

SIBPAL

This is a model-free linkage program that models trait data from full-sib pairs as a function of marker allele sharing identity-by-descent (IBD). Available analyses can use both single- and multi-point IBD information, and models allow for both binary and continuous traits due to multiple genetic loci, including epistatic interactions, and covariate effects. Like the original SIBPAL, it uses linear regression and hence is extremely fast.

10.1 Limitations

The Haseman-Elston linkage test in this release only includes support for univariate analysis of sibling pairs for autosomal regions. Full support for multivariate analysis and using all relative pairs will be available in future releases.

Unlike earlier versions of SIBPAL, this program does not generate IBD sharing estimates itself. That must be done using GENIBD, which outputs an IBD sharing file as input for SIBPAL.

10.2 Theory

10.2.1 Basic notation

Let the number of individuals be *N* with trait values: $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N$.

Let the number of covariates be *c* with values for sib i: $\mathrm{z}_{1i}, \mathrm{z}_{2i}, \ldots, \mathrm{z}_{ci}$.

Let $\hat{f}_{11}$, $\hat{f}_{12}$, ..., $\hat{f}_{1j}$, ... be the probability of sharing 1 allele IBD for the jth sib pair.

Let $\hat{f}_{21}$, $\hat{f_{22}}$, ..., $\hat{f}_{2j}$, ... be the probability of sharing 2 alleles IBD for the jth sib pair.

Let $\hat{\pi}_1$, $\hat{\pi}_2$, ..., $\hat{\pi}_j$, ... be the average allele sharing IBD (proportion of alleles shared) for the jth sib pair, where $\widehat{\pi}_j = \hat{f_{2j}} + w_1 \hat{f_{1j}}$ and $w_1 \in [0,\, 0.5]$.[1]

These probabilities are conditional on the marker[2] information available.

[1] The default value of w_1 is 0.5.

[2] In this context, marker $\equiv$ marker location, and need not be a measured marker. This is mainly an issue dealt with in the IBD generation phase.

Let the number of sibships be P.

Let the number of full sibling pairs in the i'th sibship be n_i: n_1, n_2, ..., n_P.

Let j index the sib pair: j=1, 2, ..., $\sum_i n_i = n$.

10.2.2 Test of Mean Allele Sharing

We want estimates of the means of the $\hat{\pi}_j$ and $\hat{f}_{ij}$, which we will denote $\bar{\hat{\pi}}$ and $\bar{\hat{f}}_i$, and test the hypothesis that their values agree with expectation under random sampling. These tests are that $E(\bar{\hat{\pi}}) = \pi$ and $E(\bar{\hat{f}}_i) = f_i$, where $\pi = f_2 + w_1 f_1$ and $(f_0, f_1, f_2) = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4},\right)$ for a random sample of full sib pairs if there is no meiotic drive or selection. The means and their variances are estimated by calculating:

$$\begin{aligned} \bar{\hat{\pi}} &= \tfrac{1}{n}\sum_j \hat{\pi}_j \qquad s^2_{\hat{\pi}} = \tfrac{1}{n(n-1)}\sum_j (\hat{\pi}_j - \bar{\hat{\pi}})^2 \\ \bar{\hat{f}}_i &= \tfrac{1}{n}\sum_j \hat{f}_{ij} \qquad s^2_{\hat{f}_i} = \tfrac{1}{n(n-1)}\sum_j (\hat{f}_{ij} - \bar{\hat{f}}_i)^2. \end{aligned} \tag{10.1}$$

From each mean, a t-statistic is computed and referred to the t-distribution with n-1 d.f. for a two-sided test, i.e., the p-values are

$P\left(t_{n-1} \geq \frac{\left|\bar{\hat{\pi}}-\pi\right|}{s_{\hat{\pi}}}\right)$ and $P\left(t_{n-1} \geq \frac{\left|\bar{\hat{f}}_i - f_i\right|}{s_{\hat{f}_i}}\right)$,

where t_{n-1} is a random variable that is distributed as t with n-1 d.f.

10.2.3 Test of Mean Allele Sharing for Binary Traits in Selected Pairs

The above tests can be performed separately for pairs with 0, 1, and 2 affected members as tests for linkage. However, all tests are then one-sided and the p-values are

$P\left(t_{n-1} \geq \frac{\delta\left[\bar{\hat{\pi}}-\pi\right]}{s_{\hat{\pi}}}\right)$ and $P\left(t_{n-1} \geq \frac{\delta\left[\bar{\hat{f}}_i - f_i\right]}{s_{\hat{f}_i}}\right)$

where δ=1 for affected pairs (2 affected members) if *i* = 2 and for unaffected pairs (0 affected members) if *i* = 0, and δ= -1 for discordant pairs (1 affected member). No tests are performed if *i* = 1.

10.2.4 Generalized Haseman and Elston Linkage Test

10.2.4.1 Dependent variables

Denote the j-th sib-pair with the subscript ii', and let

$$\bar{x} = \frac{1}{2N}\sum_{j=ii'=1}^{N} (x_i + x_{i'}).$$

Then the dependent variable for the ii'-th pair can be

$$y_j = \begin{cases} (x_i - \overline{x})(x_{i'} - \overline{x}) & mean{-}corrected\,cross-product\,(default) \\ -\frac{1}{2}[(x_i - \overline{x}) - (x_{i'} - \overline{x})]^2 \;=\; -\frac{1}{2}(x_i - x_{i'})^2 & -\frac{1}{2}(squared\,pair\,trait\,difference) \\ \frac{1}{2}[(x_i - \overline{x}) + (x_{i'} - \overline{x})]^2 & \frac{1}{2}(squared\,mean-corrected\,trait\,sum) \\ w_i[(x_i - \overline{x}) + (x_{i'} - \overline{x})]^2 - (1 - w_i)(x_i - x_{i'})^2 & \textit{weighted combination of the squared trait difference and squared mean-corrected sum} \end{cases}$$

Any individual who has a missing trait, marker or covariate value is not used, and not included in any pair that requires it - i.e., those pairs are not included in n.

Note that in the case of a binary trait, x_i = 1 for an affected individual and 0 for an unaffected individual, and x_i is then treated the same as for any other quantitative trait to obtain y_j.

10.2.4.2 Regression Model

The basic model we fit is of the form

$$y = \alpha + \sum_h a_h \hat{\pi}_h + \sum_h d_h \hat{f}_{2h} + \sum_k c_k f(z_k) + \varepsilon \tag{10.2}$$

where α is the intercept, and in a random sample a_i is the additive genetic variance due to the h-th marker when $w_1 = 0.5$, $\widehat{\pi} = \hat{f}_2 + w_1 \hat{f}_1$, d_h is the dominant genetic variance due to the h-th marker, c_k is a nuisance parameter accounting for the effect of some function f of the k'th covariate, and ε is the residual error. The variances a_h and d_h are the trait locus-specific variances attenuated by the recombination fraction between the trait and marker loci, when w_1 is 0.5. An iterative method using generalized estimating equations (GEE) is used to fit this model to allow for the non-independence of sibling pairs.

A problem may occur when performing multiple regression using multi-point IBD estimates. The IBD sharing between closely linked markers can be almost totally linearly dependent, resulting in a singular design matrix during Trait Regression. If multiple regression is to be performed using multiple markers with multi-point IBD sharing information, it is recommended that the loci used should have significantly different information content (i.e. be on different chromosomes or have at least one informative marker between them).

10.2.4.3 Correlation Matrices

In a nuclear family, if there are s sibs in the sibship, there are s(s - 1)/2 sib pairs and $s^2(s - 1)^2/4$ entries in the correlation matrix for that sibship, of which

s(s - 1)/2 are 1 (down the main diagonal)

s(s - 1)(s - 2) are r_1(when the pair of pairs has one sib in common)

s(s - 1)(s - 2)(s - 3)/4 are r_o(when the pair of pairs has no sibs in common)

where r_1 and r_o are correlations that differ depending on the dependant variable.

Below are the correlation matrices for sib pairs in sibships of size 3 (top left 3×3 matrix), 4 (top left 6×6 matrix) and 5 (whole matrix).

Sib1,Sib2	**1,2**	**1,3**	**2,3**	**1,4**	**2,4**	**3,4**	**1,5**	**2,5**	**3,5**	**4,5**
1,2	1	r_1	r_1	r_1	r_1	r_0	r_1	r_1	r_0	r_0
1,3	r_1	1	r_1	r_1	r_0	r_1	r_1	r_0	r_1	r_0
2,3	r_1	r_1	1	r_1	r_0	r_0	r_0	r_0	r_0	r_0
1,4	r_1	r_1	r_1	1	r_1	r_0	r_0	r_1	r_0	r_1
2,4	r_1	r_0	r_1	r_1	1	r_0	r_1	r_1	r_0	r_0
3,4	r_0	r_1	r_1	r_1	r_1	1	r_1	r_0	r_1	r_1
1,5	r_1	r_1	r_0	r_1	r_0	r_0	1	r_1	r_1	r_1
2,5	r_1	r_0	r_1	r_0	r_1	r_0	r_1	1	r_1	r_1
3,5	r_0	r_1	r_1	r_0	r_0	r_1	r_1	r_1	1	r_1
4,5	r_0	r_0	r_0	r_1	r_1	r_1	r_1	r_1	r_1	1

Let **r** = (r_1, r_0) be a vector of correlations, where r_i is the correlation between pairs sharing i individuals in common. **r** is either estimated from the data for the chosen dependent variable, with the restriction that the correlations are constrained to be greater than 0 to avoid numerical instability, or set to 0 by the user.

Let R_s be the correlation matrix for a sibship of size s. There are n_i (i = 1, 2,..., *P*) pairs in the i-th family, $n = \sum_{i=1}^{P} n_i$ pairs all told, $n_i = s_i(s_i - 1)/2$. Let W be a block diagonal matrix of the R_i :

$$W = \begin{matrix} R_{n1} & 0 & \cdots & 0 \\ 0 & R_{n2} & & 0 \\ & & \ddots & 0 \\ 0 & \cdots & 0 & R_{nP} \end{matrix}$$

10.2.4.4 Univariate Test of Linkage Using Full Sib Pairs

We want to calculate the vector of m estimates[3]:

$$b = (A^T W^{-1} A)^{-1} A^T W^{-1} y, \tag{10.3}$$

where *y* is an n $\times$ 1 vector of dependent variates with transpose y^T = (y_1, y_2...y_n). A is an n $\times$ m design matrix, where m is the number of parameters estimated - each parameter corresponds to a particular column of A.

Columns of A:

1. The first column is a column of 1's
2. Following this come one or two columns for each marker locus entered in the model. The first of each of these is a column whose elements are $\hat{\pi}_j$ and the second of each (if present) is a column whose elements are $\hat{f}_{2j}$.For each marker, the user can choose whether or not to include dominance (f_2) in the model. If it is not included, a_i is an attenuated measure of the total (additive and dominant) genetic variance when w_1 is 0.5.

[3]The equations are not actually computed by SIBPAL as listed. A significantly more complex method is implemented that is efficient and numerically stable.

3. Following this may come one or more columns each element of which is the product of elements of two (or more) of the previous columns.

4. Following this come one or more columns for each covariate entered in the model. Each of these is a column whose elements are the mean-corrected sum, absolute difference, or mean-corrected cross product between the sib pair covariate values. Additional columns may be entered that are powers of these covariate sums, differences or products.

Let the k'th covariate term included in the model for the ii'-th pair be

$$z_{kii'} = \begin{cases} (z_{ki} - \bar{z}_k) + (z_{ki'} - \bar{z}_k) & \text{mean-corrected covariate sum} \\ \left|(z_{ki} - \bar{z}_k) - (z_{ki'} - \bar{z}_k)\right| & \text{mean-corrected covariate absolute difference} \\ (z_{ki} - \bar{z}_k)(z_{ki'} - \bar{z}_k) & \text{mean-corrected covariate product (default)} \end{cases}$$

for covariate *k* and individuals *i* and $i^{'}$.

For each covariate, the user can choose whether or not to include any combination of the sum, difference and product terms in the model, as well as powers of them. Including too many covariate terms may cause A to be singular due to linear dependencies in the data. Covariate means may be specified in the parameter file or estimated from the set of relative pairs used in the analysis.

1. Following this may come one or more columns, each element of which is the product of elements of two (or more) of the covariate terms in previous columns.

Thus, A will be of the form:

$$\begin{array}{ccccccccccccccc} 1 & \hat{\pi}_{11} & f_{211} & \hat{\pi}_{21} & f_{221} & \cdots & \hat{\pi}_{11}\hat{\pi}_{21} & \cdots & [z_{11}]^p & [z_{21}]^p & \cdots & [z_{11}z_{21}]^p & \cdots \\ 1 & \hat{\pi}_{12} & f_{212} & \hat{\pi}_{22} & f_{222} & \cdots & \hat{\pi}_{12}\hat{\pi}_{22} & \cdots & [z_{12}]^p & [z_{22}]^p & \cdots & [z_{12}z_{22}]^p & \cdots \\ 1 & \hat{\pi}_{13} & f_{223} & \hat{\pi}_{23} & f_{223} & \cdots & \hat{\pi}_{13}\hat{\pi}_{23} & \cdots & [z_{13}]^p & [z_{23}]^p & \cdots & [z_{13}z_{23}]^p & \cdots \\ \vdots & \vdots & & \vdots & & & \vdots & & \vdots & & & \vdots & \end{array}$$

(a)	(b) first marker	(b) second marker	(c) marker interactions	(d) covariate sum, difference and product terms	(e) covariate interactions

The first n_1 rows of A are for family 1,

the next n_2 are for family 2,

.

.

.

the last n_P are for family *P*.

10.2.4.5 Output of estimates and t-statistics

We want the estimates b_i, the three elements of *b* indicated by (10.3), and the test statistics $\frac{b_i}{s_i}$, for i = 2, 3, ..., *m*, where s_i^2 is

$$\frac{y^T W^{-1}(y - Ab)d_{ii}}{n - m} \tag{10.4}$$

in which d_{ii} is the i-th diagonal element of $(A^T W^{-1} A)^{-1}$. For each test statistic we calculate a p-value which is either

$$\mathrm{P}_t = P\left(t_{n-m} \geq \frac{b_i}{s_i}\right) \tag{10.5}$$

or

$$\mathrm{P}_t = P\left(t_{n-m} \geq \frac{|b_i|}{s_i}\right). \tag{10.6}$$

Estimates b_i corresponding to a column of $\hat{\pi}s$ or $\hat{f}_2 s$ and other columns of marker terms (i.e., products of $\hat{\pi}s$ and $\hat{f}_2 s$) use 10.5. A two-sided test (10.6) is used for all remaining columns that contain any covariate terms.

Alternatively, the above tests can be performed using variances estimated using an estimator that is robust to misspecification of the model and the correlation matrices. When this option is specified, the covariance matrix of the parameter estimates is computed using the *sandwich* variance estimator

$$(A^T W^{-1} A)^{-1} \left[y^T W^{-1}(y - Ab_)\right] \left[y^T W^{-1}(y - Ab\,)\right]^T (A^T W^{-1} A)^{-1}. \tag{10.7}$$

These variance estimates can be extremely conservative and caution should be exercised when using this option.

10.2.4.6 Empirical estimates of significance

We can also estimate an empirical p-value of the test statistic using a Monte Carlo permutation procedure with N replicate permutations. For each replicate, we permute the allele sharing among the pairs (both within sibships and across sibships of the same size), recalculate the test statistic, and determine the proportion of the replicates that are equal to or greater than the statistic calculated from the original observations. We choose N, the number of replicates, such that the estimated empirical p-value, $\hat{p}$, is within a proportion w (the width parameter) of its true p-value, p, with predetermined confidence probability γ (the confidence parameter). That is, we want the standard deviation $s_{\hat{p}}$ of $\hat{p}$ to be proportional to $\hat{p}$. This permutation process can be viewed as a set of N independent Bernoulli trials each with success probability p. The sample variance, $s^2_{\hat{p}}$, of $\hat{p}$ is $s^2_{\hat{p}} = \frac{\hat{p}(1-\hat{p})}{N}$. So we choose N such that $Pr(|\hat{p} - p| \leq w\hat{p}) = \gamma$. Using a normal approximation for the distribution of $\hat{p}$, we obtain

$$N = \left(\frac{1-\hat{p}}{w^2 \hat{p}} \left[\Phi^{-1}\left(\frac{\gamma+1}{2}\right)\right]^2\right),$$

where Φ is the standard normal cumulative distribution function. We estimate N by substituting for $\hat{p}$ the p-value obtained on assuming the test statistic follows a t-distribution, and use this number of replicates to obtain an empirical p-value within any prespecified proportion of its true value with a known confidence coefficient. For example, if we wish to estimate an empirical p-value within 20% of its true value with 95% confidence, then N should be approximately $\frac{100(1-\hat{p})}{\hat{p}}$. The number of replicates, *N*, can be limited to avoid excessive computing time.

10.3 Program Input

File Type	Description
Parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual including fields for identifiers, sex, parents, trait and marker data.
IBD sharing file	Stores identity-by-descent (IBD) distributions between pairs of related individuals at one or more marker loci.

10.3.1 The `sibpal` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main SIBPAL parameter.

`parameter [, attribute]`	**Explanation**	
`sibpal`	Starts a SIBPAL parameter block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	traits
	Required	No
	Applicable Notes	None

10.3.2 The `sibpal` Block

The following syntax table specifies the permissible parameter and attribute settings for the `sibpal` block.

parameter [, attribute]	**Explanation**
`mean_test`	Starts a sub-block for specifying a test of mean IBD sharing. Value Range: N/A Default Value: None Required: Yes Applicable Notes: None
`trait_regression`	Starts a sub-block for specifying a regression of traits on one or more markers, covariates, and interactions Value Range: N/A Default Value: None Required: Yes Applicable Notes: 1, 2
`, zero_marker` `, zero`	Specifies that regression will be performed on covariate(s) only, and any listed markers will be disregarded. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, single` `, single_marker`	Selects single regression on one marker at a time. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, multiple` `, multiple_marker`	Selects multiple regression on all chosen markers at once. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`trait_regression_default`	Specifies default type of regression for listed trait regression blocks. Value Range: {zero, single, multiple} Default Value: single Required: Yes Applicable Notes: 1

Notes

1. If a `trait_regression` statement does not have either the `single` or `multiple` attributes, then the `trait_regression_default` statement will determine whether the given marker or interval estimates will be regressed one at a time (`single`) or all at once

(`multiple`).

Each `trait_regression` statement performs a test of linkage of a trait to one or more markers. The analysis may consist of several regression tests each using a single marker, if either the `single` attribute is included or the value of the `trait_regression_default` parameter is set to `single`. Similarly, a single multiple-regression test is performed if either the `multiple` attribute is included or the value of the `trait_regression_default` parameter is set to `multiple`. The traits, covariates, markers and other options to be used may be listed in a sub-block of the `trait_regression` statement. All options changed in a sub-block are local to the analysis being performed, and do not affect further analyses. If no sub-blocks are listed, then analysis will be performed using all traits and all markers. All parameters that may be included in the sub-block are optional and all values are case-insensitive.

2. A single regression is performed by default.

10.3.3 The `mean_test` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `mean_test` sub-block.

`parameter [, attribute]`	**Explanation**
`marker`	Specifies the name of a marker for which to test mean IBD sharing Value Range: Character string representing the name of a marker listed in the pedigree data file. Default Value: None Required: No Applicable Notes: 1
`trait`	Names a trait denoting affection status. Analysis is performed separately on concordantly affected, unaffected and discordant pairs. Value Range: Character string representing the name of a trait listed in the pedigree data file. Default Value: None Required: Yes Applicable Notes: None
`subset`	Specifies a trait used as an indicator variable to select subsets of pairs to analyze. Value Range: Character string representing the name of a trait listed in the pedigree data file. Default Value: None Required: No Applicable Notes: 2
`wide_out`	Prints more verbose output information. This causes some output tables to be more than 80 columns wide. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 3
`export_output`	Specifies option to produce tab-delimited output that can easily be imported to other programs such as Excel, SAS and SPlus. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None

pval_scientific_notation	Specifies option to print p-values using scientific notation as opposed to the default of fixed decimal notation.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	None

Notes

1. The value of a marker parameter should be set to the name of a marker for which IBD sharing information was generated and stored in the IBD sharing file. If no valid marker parameters are listed, then all markers are used. The following are all valid mean_test statements:

```
mean_test # Test each marker

mean_test { # Equivalent to the previous statement.

}

mean_test {
   marker=M1
   marker="region 1 MRK"
   marker=M3
}
```

2. The `subset` parameter specifies a trait to be used as an indicator variable to limit the individuals that may be used in an analysis; individuals for whom this indicator is zero are assumed to have missing trait values. It may be included more than once, in which case the only individuals included in the analysis are those for which all the indicated binary traits are coded 1. The trait being analyzed for linkage should not be used as a subset variable. If the trait specified is a binary trait, it should be coded as `0` for individuals to be excluded from analysis and `1` for individuals to be included. Only those individuals that are affected will be considered. If the trait is continuous, only individuals with trait values greater than 0 will be included. This option does not alter the direction of any of the test statistics as would the `trait` parameter, so it is usually not appropriate to specify subsets based on phenotypes that are useful for testing linkage.

3. If the `wide_out` parameter is set to **true**, then additional columns are added to the output from Trait Regression analyses, including a column of t-values corresponding to each parameter estimate.

10.3.4 The `trait_regression` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `trait_regression` sub-block.

parameter [, attribute]	**Explanation**
`trait`	Specifies a trait to be used as the dependant variable in the current test.
	Value Range: Character string representing the name of a trait listed in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 1
`, mean`	Fixes the trait mean to a value other than the sample mean.
	Value Range: N/A
	Default Value: None
	Required: No
	Applicable Notes: 1
`marker`	Specifies a marker to be included in the current test.
	Value Range: Character string representing the name of a marker listed in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 2
`, dominance` `, dom`	Specifies option to test the additive and dominance variances linked to the marker separately instead of the total variance.
	Value Range: N/A
	Default Value: None
	Required: No
	Applicable Notes: 3
`covariate`	Names a covariate.
	Value Range: Character string representing the name of a covariate listed in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 4
`, prod`	Include the covariate mean-corrected product.
	Value Range: true false
	Default Value: true
	Required: No
	Applicable Notes: None

, sum	Include the covariate mean-corrected sum. Value Range: true, false Default Value: false Required: No Applicable Notes: None
, diff	Include the covariate difference. Value Range: true, false Default Value: false Required: No Applicable Notes: None
, all	Include all covariate terms (sum, difference and product). Value Range: true, false Default Value: false Required: No Applicable Notes: None
, power	Raise covariate terms to specified power. Value Range: (-∞, ∞) Default Value: 1.0 Required: No Applicable Notes: None
interaction	Starts a `parameter` sub-block that contains marker and covariate parameters that represent a multiplicative interaction term to be included in the regression model . Value Range: N/A Default Value: None Required: No Applicable Notes: 5
regression_method	Specifies which of the following dependent variables to use in the current test. Value Range: diff, sum, prod, W2, W3, W4 Default Value: prod Required: No Applicable Notes: 6

`add_sum_covariate`	If this parameter is set to **true** and `regression_method` = **sum**, then SIBPAL will automatically include the trait sum as a regression covariate (thereby increasing power to detect linkage).
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: None
`subset`	Specifies option to use only a subset of the data. The trait specified should be an indicator variable.
	Value Range: Character string representing the name of a trait, phenotype or covariate listed in the pedigree file, or the name of a variable specified within a function block.
	Default Value: None
	Required: No
	Applicable Notes: 7
`weight`	Names a trait to use as a regression weight for pairs. Weights are computed as the product of each individuals trait value.
	Value Range: Character string representing the name of a trait, phenotype or covariate listed in the pedigree file, or the name of a variable specified within a function block.
	Default Value: None
	Required: No
	Applicable Notes: None
`identity_weights`	Specifies option to assume that all sib pairs are independent by using the identity working matrix.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: None
`robust_variance`	Compute the variance of parameter estimates using the robust, or *sandwich*, variance estimator. This can lead to very conservative tests for larger samples containing non-independent sibling pairs.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: None

`sibship_mean`	Use the sibship mean when mean-correcting trait values. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`, threshold`	Use the grand mean for all sibships with fewer than the specified number of siblings. Value Range: {1, 2, 3, ...} Default Value: 3 Required: No Applicable Notes: None
`wide_out`	Prints more verbose output information. This causes some output tables to be > 80 columns wide. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 8
`compute_empirical_pvalues`	Compute empirical p-values by permutation. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`, threshold`	Only compute empirical p-values for asymptotic p-values less than this value. Value Range: [0,1] Default Value: 0.05 Required: No Applicable Notes: None
`, permutations`	Specifies an exact number of permutations that should always be performed if the asymptotic p-value is less than threshold. Use of this option effectively overrides all of the following attributes. Value Range: {0, 1, 2, 3, ...} Default Value: None Required: No Applicable Notes: None
`, max_permutations`	Specifies the maximum number of permutations that should be performed. Value Range: {0, 1, 2, 3, ...} Default Value: 10000 Required: No Applicable Notes: None

, width	Specifies the relative precision of the empirical p-value. E.g., if width=0.2, p-values will be estimated to be within 20% of their true value with a given confidence level. This value is used to choose the number of replicates necessary. Note that the number of replicates required varies quadratically with the inverse of the width. Value Range: [0,1] Default Value: 0.2 Required: No Applicable Notes: None
, confidence	Specifies the confidence with which an empirical p-value is required to be within a relative interval (i.e., the width) of its true value. Value Range: [0,1] Default Value: 0.95 Required: No Applicable Notes: None
skip_uninformative_pairs	Option to skip pairs of individuals whose prior and observed IBD sharing probabilities are numerically identical given the machine precision. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
export_output	Specifies option to produce tab-delimited output that can easily be imported to other programs such as Excel, SAS and SPlus. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
pval_scientific_notation	Specifies option to print p-values using scientific notation as opposed to the default of fixed decimal notation. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
print_design_matrix	Specifies option to print the design matrix *A*. Value Range: Character string representing the name of a marker or location listed in the IBD sharing file. Default Value: None Required: No Applicable Notes: 9

print_correlation_matrix	Specifies option to print the sibship specific correlation matrices for dependent variable.	
	Value Range	Character string representing the name of a marker or location listed in the IBD sharing file.
	Default Value	None
	Required	No
	Applicable Notes	10

Notes

1. The value of a `trait` parameter should be set to the name of a `trait`, `phenotype` or `covariate` field read from the pedigree data file. If no valid `trait` parameters are listed, then all `trait` fields read in are used. If more than one `trait` is specified, then multiple univariate regressions are performed using each trait with all markers and covariates listed. The *population* mean of the trait may be used in computing the mean-corrected trait values. This is specified by including an attribute, mean, with value set to the desired trait mean. Otherwise, the trait mean is estimated from the sample of individuals used in the regression.

2. The value of a `marker` parameter should be set to the name of a marker for which IBD sharing information was generated and stored in the IBD sharing file. If no valid marker parameters are listed then all markers are used.

3. If a `marker` parameter has the `dom` or `dominance` attribute, then the additive and dominance variances due to that marker will be tested separately (i.e. there will be regression on both $\hat{\pi}$ and $\hat{f}_2$); and a marker parameter without this attribute will test total genetic variance due to that marker (i.e. there will be regression on $\hat{\pi}$ only).

4. The value of a `covariate` parameter should be set to the name of a `trait`, `phenotype` or `covariate` field read from the pedigree data file. If no valid `covariate` parameters are listed, then by default no covariates are included.

5. The `interaction` parameter should contain a sub-block of `marker` and `covariate` parameters that specify a multiplicative interaction term in the regression model. e.g., the following interaction sub-block specifies a gene-environment interaction term between the dominance component of D1S344 and the squared BMI difference:

```
interaction {
   marker    = D1S344, dom
   covariate = BMI, diff, power = 2
}
```

6. The values for the regression_method are explained as follows:

Value	Meaning
diff	-$\frac{1}{2}$ squared trait difference ($-\frac{1}{2}$ $\times$ traditional Haseman-Elston).
sum	$\frac{1}{2}$ squared mean-corrected trait sum.
prod	Mean-corrected cross-product
W2	Weighted combination of squared trait difference and squared mean-corrected trait sum. Weights are chosen proportional to the inverses of the residual variances of the squared differences and sums.
W3	Weighted combination of squared trait difference and squared mean-corrected trait sum, as above but further adjusted for the non-independence of sib-pairs.[a]
W4	Weighted combination of squared trait difference and squared mean-corrected trait sum, as above but further adjusted for the non-independence of sib-pairs and the non-independence of squared trait sums and differences.[b]

[a] This method should be more powerful asymptotically (see Shete, et al., 2003)

[b] This method should be the most powerful asymptotically (see Shete, et al., 2003)

7. The `subset` parameter specifies a trait to be used as an indicator variable to limit the individuals that may be used in an analysis; individuals for whom this indicator is zero are assumed to have missing trait values. It may be included more than once, in which case the only individuals included in the analysis are those for which all the indicated binary traits are coded 1. The trait being analyzed for linkage should not be used as a subset variable. If the trait specified is a binary trait, it should be coded as `0` for individuals to be excluded from analysis and `1` for individuals to be included. Only those individuals that are affected will be considered. If the trait is continuous, only individuals with trait values greater than 0 will be included. This option does not alter the direction of any of the test statistics as would the `trait` parameter, so it is usually not appropriate to specify subsets based on phenotypes that are useful for testing linkage.

8. If the `wide_out` parameter is set to **true**, then additional columns are added to the output from Trait Regression analyses, including a column of t-values corresponding to each parameter estimate.

9. If either the `zero_marker` or `multiple_marker` attribute is specified for the `sibpal` parameter, then no value is required to specify the location. If the `single_marker` attribute is specified, then a character string representing the name of a marker or location listed in the IBD sharing file may be used to specify the location to print. If no value is specified for single marker regression, then the first n rows of the design matrix for all locations will be printed.

10. If either the `zero_marker` or `multiple_marker` attribute is specified for the `sibpal` parameter, then no value is required to specify the location. If the `single_marker` attribute is specified, then a character string representing the name of a marker or location listed in the IBD sharing file must be used to specify the location to print.

10.4 Program Execution

SIBPAL is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running SIBPAL from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>sibpal
S.A.G.E. v5.x -- SIBPAL
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: sibpal <parameters> <pedigree> <IBD...>
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
IBD - IBD sharing file(s)
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of SIBPAL may look like the following:

```
>sibpal sibpal.par example.ped example.ibd
S.A.G.E. v5.x -- SIBPAL
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Loading parameters.............done.
Reading pedigree data...........
from example.ped.........done.
Sorting pedigrees..............done.
Reading pairs..................done.
Sorting pairs..................done.
Computing mean test....... 1...done.
Computing mean test....... 2...done.
Computing trait regression 1..done.
Analysis complete!
```

10.5 Program Output

SIBPAL produces several output files that contain results and diagnostic information:

Filename	File Type	Description
sibpal.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
means.out	Mean analysis output file	Contains the results of each test of mean allele sharing IBD.
traits.out	Trait Regression analysis output file	Contains the results of each linkage test.

10.5.1 Information Output File

The SIBPAL Information Output file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read in, are included with all programs in the S.A.G.E. and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not run as expected. The file "`sibpal.inf`" should be checked for errors and diagnostic information after each run of the program.

10.5.2 Mean Analysis Output File

One Mean Analysis output file, named "`means.out`" , is generated per run of SIBPAL. It includes the results of all non-trait specific mean analyses: average allele sharing, as well as 0, 1 and 2 alleles IBD are output in a table with standard errors and p-values for each estimate.

10.5.3 Trait Regression Analysis Output File

One Trait Regression analysis output file, named "`traits.out`", is generated per run of SIBPAL. It contains the results of all Trait Regression linkage tests. Each coefficient estimated is printed in a table with its standard error and p-value.

10.6 Example Output Files

10.6.1 Mean Analysis Output File

```
SIBPAL Output -- 2 Aug 2005 15:41:09 -- [S.A.G.E. v5.0.3; bld 01 Aug 2005]
COPYRIGHT (C) 2005 CASE WESTERN RESERVE UNIVERSITY
======================================================================
  Test of Mean Allele Sharing IBD for Full Sib Pairs
======================================================================
  Estimates:
    pi - Average proportion of alleles shared IBD.
    fi - Estimated proportion of sib pairs sharing i alleles IBD.
======================================================================
Marker            Pairs        Estimate      Std Error   P-value
----------------------------------------------------------------------
D5G1                 28  pi    0.45535714    0.04820449  0.36259122038
                         f0    0.29464286    0.05310089  0.40789006347
                         f1    0.50000000    0.02571722  -------------
                         f2    0.20535714    0.04645782  0.34511298675
----------------------------------------------------------------------
D5G2                 28  pi    0.43750000    0.05696380  0.28224850647
                         f0    0.30357143    0.07426574  0.47689606098
                         f1    0.51785714    0.07919128  -------------
                         f2    0.17857143    0.06410910  0.27502563114
----------------------------------------------------------------------
D5G3                 28  pi    0.54464286    0.05310089  0.40789006347
                         f0    0.17857143    0.05729972  0.22325916300
                         f1    0.55357143    0.06967083  -------------
                         f2    0.26785714    0.06916042  0.79821143299
----------------------------------------------------------------------
D5G4                 28  pi    0.55357143    0.04699492  0.26432288020
                         f0    0.17857143    0.04786642  0.14722754816
                         f1    0.53571429    0.05709323  -------------
                         f2    0.28571429    0.06128351  0.56488301140
----------------------------------------------------------------------
D5G5                 28  pi    0.52678571    0.06212092  0.66975583933
                         f0    0.19642857    0.06967083  0.44861400075
                         f1    0.55357143    0.09033772  -------------
                         f2    0.25000000    0.08333333  1.00000000000
----------------------------------------------------------------------
.
.
.
```

10.6.2 Trait Regression Analysis Output File

```
SIBPAL Output -- 2 Aug 2005 15:43:12 -- [S.A.G.E. v5.0.3; bld 01 Aug 2005]
COPYRIGHT (C) 2005 CASE WESTERN RESERVE UNIVERSITY
================================================================================
  Haseman-Elston Regression Analysis of Full Sibs - single_marker regression
================================================================================
    Binary trait : affection, affected = 'A' , unaffected = 'U'
          Number of full sib pairs    =   28
          Sample mean                 =    0.0370
          Sample variance             =    0.0370
          Sample skewness             =    4.9029
          Sample kurtosis             =   22.0385
          Sibling correlation         =    0.0610
    Dependent variate : Squared trait difference
          Correlation between pairs with no sibs in common =    0.0000
          Correlation between pairs with one sib in common =    0.0000
          Correlation between squared difference
                          and squared mean corrected sum   =   -1.0000
          Intercept                                        =   -0.0121
          Total variance                                   =    0.0086
          Residual variance                                =    0.0093
          Residual skewness                                =   -4.9942
          Residual kurtosis                                =   22.9761
    Other options used :
          Identity weights = no
          Robust variance  = no
          Use sibship mean = no
    Legend :
      Note: kurtosis = coefficient of kurtosis - 3
      *   - significance  .05 level;
      **  - significance  .01 level;
      *** - significance .001 level;
================================================================================
Independent                                                              Nominal
variable                    Pairs   Parameter     Estimate Std Error   P-value
--------------------------- ----- ----------- ------------ --------- ---------
D5G1                           28 (A+D)GenVar      -0.0127    0.0726 0.5688081
D5G2                           28 (A+D)GenVar      -0.0127    0.0614 0.5813646
D5G3                           28 (A+D)GenVar       0.0107    0.0659 0.4361386
D5G4                           28 (A+D)GenVar       0.0163    0.0748 0.4147840
D5G5                           28 (A+D)GenVar       0.0466    0.0557 0.2047478
D5G6                           28 (A+D)GenVar       0.0076    0.0629 0.4525494
D5G7                           28 (A+D)GenVar       0.0667    0.0691 0.1715900
D5G8                           28 (A+D)GenVar       0.0000    0.0755 0.5000000
D5G9                           28 (A+D)GenVar      -0.0512    0.0587 0.8046652
D5G10                          28 (A+D)GenVar       0.0000    0.0681 0.5000000
D5G11                          28 (A+D)GenVar       0.0042    0.0662 0.4748203
D5G12                          28 (A+D)GenVar       0.0085    0.0756 0.4553801
D5G13                          28 (A+D)GenVar      -0.0552    0.0642 0.8009718
.
.
.
D5G24                          28 (A+D)GenVar       0.0141    0.0612 0.4100034
D5G25                          28 (A+D)GenVar       0.0942    0.0554 0.0502490
================================================================================
```

Chapter 11

LODPAL

LODPAL performs a linkage analysis based on the LOD score formulation for affected-sib-pairs (ASPs) (Risch, 1990). The current implementation is of the general conditional logistic model proposed by Olson (1999) modified to give the one-parameter model of Goddard et al. (2001). The model allows for the inclusion of all affected-relative-pairs (ARPs) and covariates or discordant sibling pairs, with the possibility of pooling unaffected relative pairs together with ARPs in the analysis.

11.1 Limitations

The current release only includes support for a single disease locus and assumes all pairs of relatives are independent.

11.2 Theory

11.2.1 Basic notation

Let the number of relative pairs be N.

Let i index the relative pair: i = 1, 2, . . . , N.

Let f_{r0}, f_{r1}, and f_{r2} be the prior probabilities of sharing 0, 1, or 2 alleles IBD given a relative pair of type r.

Let w_i be a weight corresponding to the ith pair.

Let

$\hat{f}_{0i}$ be the probability of sharing 0 alleles IBD at a given marker location, for the ith pair,

$\hat{f}_{1i}$ be the probability of sharing 1 allele IBD at a given marker location, for the ith pair, and

$\hat{f}_{2i}$ be the probability of sharing 2 alleles IBD at a given marker location, for the ith pair.

These three IBD-sharing probabilities are estimated by GENIBD given the available marker data and given the pedigree relationship (i.e., type of relative pair). They may be multipoint or single-marker estimates. Marker is equivalent to marker location, and need not be a measured marker. This is mainly an issue dealt with in the IBD generation phase.

The following table summarizes the various notations that have been used for the probability of sharing *i* alleles IBD between affected sib pairs at a particular locus:

# Alleles Shared IBD	Probabilities		
0	Z_0	$\frac{1}{\lambda_0+2\lambda_1+\lambda_2}$	$\frac{1}{1+2e^{\beta_1}+e^{\beta_2}}$
1	Z_1	$\frac{2\lambda_1}{\lambda_0+2\lambda_1+\lambda_2}$	$\frac{2e^{\beta_1}}{1+2e^{\beta_1}+e^{\beta_2}}$
2	Z_2	$\frac{\lambda_2}{\lambda_0+2\lambda_1+\lambda_2}$	$\frac{e^{\beta_2}}{1+2e^{\beta_1}+e^{\beta_2}}$

The sibling locus-specific relative recurrence risk is given by

$$\lambda_s = \frac{1}{4}\left[\lambda_0 + 2\lambda_1 + \lambda_2\right] = \frac{1}{4}\left[1 + 2\lambda_1 + \lambda_2\right] = \frac{1}{4} + \frac{1}{2}\lambda_1 + \frac{1}{4}\lambda_2$$

11.2.2 Affected Relative Pair Linkage Analysis

11.2.2.1 Two-parameter Model (Olson 1999)

The LOD score for a set of N ARPs is

$$z = \sum_{i=1}^{N} \log_{10}\left\{ w_i \frac{\hat{f}_{0i} + \hat{f}_{1i}e^{\beta_1} + \hat{f}_{2i}e^{\beta_2}}{f_{r0} + f_{r1}e^{\beta_1} + f_{r2}e^{\beta_2}} + (1 - w_i) \right\}$$

$$= \sum_{i=1}^{N} \log_{10}\left\{ w_i \frac{\sum\limits_{k=0,1,2} \hat{f}_{ki}e^{\beta_k}}{\sum\limits_{k=0,1,2} f_{rk}e^{\beta_k}} + (1 - w_i) \right\} = \sum_{i=1}^{N} \log_{10}\left\{ w_i \frac{\sum\limits_{k=0,1,2} \hat{f}_{ki}\lambda_k}{\sum\limits_{k=0,1,2} f_{rk}\lambda_k} + (1 - w_i) \right\},$$

where λ_k is the (locus-specific) relative recurrence risk for an individual sharing k alleles IBD with an affected person and w_i is a user-specified weight to be given to the i-th relative pair. Here, β_0 = 0, and β_1, β_2 are estimated by maximizing the LOD score with the constraints $\beta_1 \geq 0$ and $\beta_2 \geq \log_e (2e^{\beta_1} - 1)$ (i.e., $\lambda_1 > 1$ and $\lambda_2 > 2\lambda_1 - 1$).

For full sibs, $f_{S_0} = \frac{1}{4},\ f_{S_1} = \frac{1}{2},\ f_{S_2} = \frac{1}{4}$, giving for the ith sib pair

$$\log_{10}\left\{ w_i 4 \frac{\hat{f}_{0i} + \hat{f}_{1i}e^{\beta_1} + \hat{f}_{2i}e^{\beta_2}}{1 + 2e^{\beta_1} + e^{\beta_2}} + (1 - w_i) \right\}.$$

For half sibs, $f_{h0} = \frac{1}{2},\ f_{h1} = \frac{1}{2},\ f_{h2} = 0$, giving for the ith half sib pair

$$\log_{10}\left\{ w_i 2 \frac{\hat{f}_{0i} + \hat{f}_{1i}e^{\beta_1}}{1 + e^{\beta_1}} + (1 - w_i) \right\}.$$

In summary,

r	f_{r0}	f_{r1}	f_{r2}
Sibs	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
Half-sibs	$\frac{1}{2}$	$\frac{1}{2}$	0
Grandparents	$\frac{1}{2}$	$\frac{1}{2}$	0
Avuncular	$\frac{1}{2}$	$\frac{1}{2}$	0
Cousins	$\frac{3}{4}$	$\frac{1}{4}$	0

In the next sections, the subscript i indexing the pair and the summation over i will be suppressed.

11.2.2.2 One Parameter Model

Under the optimal one parameter model, the LOD score contribution of a single pair is

$$\log_{10}\left\{w\frac{\hat{f}_0+\hat{f}_1e^{\beta_1}+\hat{f}_2(3.634e^{\beta_1}-2.634)}{f_{r0}+f_{r1}e^{\beta_1}+f_{r2}(3.634e^{\beta_1}-2.634)}+(1-w)\right\}.$$

The constants in the above expression fix the mode of inheritance to a value approximately halfway between a dominant and a recessive model and correspond to the Whittemore and Tu (1998) minmax model mode of inheritance parameter (they defined two parameters, w_1 and a, and the minmax values of these are $w_1 = 0.275$ and $\alpha = (2-3a)/a$. To allow more flexibility, the user may specify a different "mode of inheritance" parameter. Thus, under a generalization of this model, the LOD score contribution of a single pair is

$$\log_{10}\left\{w\frac{\hat{f}_0+\hat{f}_1e^{\beta_1}+\hat{f}_2[(\alpha+1)e^{\beta_1}-\alpha]}{f_{r0}+f_{r1}e^{\beta_1}+f_{r2}[(\alpha+1)e^{\beta_1}-\alpha]}+(1-w)\right\},$$

where $\alpha \geq 1$ is a mode of inheritance parameter: α = 1 (corresponding to Whittemore & Tu's $w_1 = a = 0.5$) gives a dominant model and $\alpha \to \infty$ (corresponding to Whittemore & Tu's $w_1 = a = 0$) gives a recessive model. (In practice, $\alpha \approx 10$ gives a pretty good recessive model.) Compared to the two-parameter model, this model has the constraints λ_2 = (α + 1) $\lambda_1 - \alpha$ in terms of relative recurrence risks and we estimate $\beta_1 \geq 0$ (default).

11.2.2.3 Covariates[1]

Inclusion of a single covariate (z) gives

[1] Covariates are pair-specific and are allowed only in the one-parameter model.

$$\log_{10}\left\{w\frac{\hat{f}_0+\hat{f}_1e^{\beta_1+y\delta_1}+\hat{f}_2[(\alpha+1)e^{\beta_1+z\delta_1}-\alpha]}{f_{r0}+f_{r1}e^{\beta_1+y\delta_1}+f_{r2}[(\alpha+1)e^{\beta_1+z\delta_1}-\alpha]}+(1-w)\right\},$$

where δ_1 is an additional parameter to be estimated and z is the adjusted (see below) value of the covariate for that pair. The model extends easily to include more than one covariate; there is one additional parameter for each covariate.

- Constraints on δ_1:

  Let the original (unadjusted) covariate value be denoted x. Two options are allowed:

  1. Genetic constraint on β_1 holds at the average value, $x = \bar{x}$, but not necessarily for all x. The covariate value is centered to give $z = x - \bar{x}$ before inclusion in the likelihood, so that the mean of the centered covariate = 0. Then δ_1 is unconstrained.
  2. Genetic constraint on β_1 holds at all values of x. The minimum value of a covariate is subtracted (i.e., $z = x-$ min x), so that the smallest value of the covariate equals zero. Then for a set of covariates, indexed by j, the following constraint is applied:

$$\min_{y>0}\sum_j z_j\delta_{1j} \geq -\beta_1.$$

11.2.3 Adding Discordant Sib Pairs (DSP) to an ARP Analysis (one-parameter model only)

This model is the same as the ARP one parameter model with one covariate that indicates nonconcordance status:

$$\lambda_1 = e^{\beta_1+y\delta_1},$$

where the covariate y is set to 0 if the pair is concordant and 1 if the pair is discordant for affection status. A related model sets the covariate y to

- 0 if the pair is concordantly affected and to
- 1 if the pair is discordant for affection status (concordantly unaffected pairs are not used in the analysis).

When either option is chosen, β_1 and δ_1 are estimated subject to the constraints: $\beta_1 \geq 0$, $\delta_1 \leq -\beta_1$.

No additional covariates may be included when discordant sib pairs are included in the analysis this way.

11.2.4 X-linked Models

Models for X-linkage are similar to those for autosomal inheritance. Recall the autosomal model: the LOD score contribution for a particular affected relative pair (ARP) of type r is

$$\log_{10}\left\{ w\frac{\sum\limits_{k=0,1,2}\hat{f}_k\lambda_k}{\sum\limits_{k=0,1,2}f_{rk}\lambda_k}+(1-w)\right\}.$$

For X-linked models, the LOD score is

$$\log_{10}\left\{ w\frac{\sum\limits_{k=0,1,2}\hat{f}_{kuv}\lambda_{kuv}}{\sum\limits_{k=0,1,2}f_{rkuv}\lambda_{kuv}}+(1-w)\right\},$$

where u, v denote the sex (m=male, f=female) of the members of the pair. For male-female ARPs, m and f are interchangeable, i.e. $\lambda_{kmf} = \lambda_{kfm}$.

There are four possible relative risk parameters: λ_{1ff}, λ_{2ff}, λ_{1mm}, and λ_{1mf} (=λ_{1fm}). All others equal 1 (e.g., $\lambda_{0ff} = \lambda_{2mm}$ = 1, etc.). The following table gives the λ parameters for each type of ARP.

		λ **Corresponding to IBD-sharing equal to**		
Type of ARP		0	1	2
Male-Male		1	λ_{1mm}	1
Male - Female		1	λ_{1mf}	1
Female - Female	ASP	1	λ_{1ff}	λ_{2ff}
Female - Female	Other types	1	λ_{1ff}	1

- Constraints on λ_{1ff}, λ_{1mm}, λ_{1mf} :
  - DEFAULT VALUE : all λ_1 constrained to be equal:
    $\lambda_{1ff} = \lambda_{1mm} = \lambda_{1mf}$
  - OPTIONAL : all λ_1 not constrained to be equal:
    λ_{1ff}, λ_{1mm}, and λ_{1mf} are estimated separately.
- Constraints on λ_{2ff}:
  - DEFAULT VALUE : λ_{2ff} = (α+1) $\lambda_{1ff} - \alpha$
    The default value of α is 2.634.
  - OPTIONAL : λ_{2ff} is not constrained to be dependent on λ_{1ff}.
    λ_{1ff} and λ_{2ff} are estimated separately. Since both parameters are not estimable if the data contains only ASPs or no ASPs, unless λ_{1ff} is estimated in part using male-male and/or male-female ASPs, this option will be carried out only if either the data contains at least 15 male-male and male-female sib pairs (ASPs) under the default constraints on λ_1, or if the data set contains at least 15 sister-sister ASPs and at least 15 female-female ARPs other than ASPs under the optional constraints on λ_1.

Under this model, the additional constraint $\beta_{2ff} \geq \log_e (2e^{\beta_{1ff}} - 1)$ is used[2].

11.2.4.1 Covariates

Inclusion of a single covariate (z) gives

$$\log_{10}\left\{ w\frac{\hat{f}_{0uv} + \hat{f}_{1uv}e^{\beta_{1uv}+\delta_{1uv}z} + \hat{f}_{2uv}[(\alpha+1)e^{\beta_{1uv}+\delta_{uv1}z} - \alpha]}{f_{r0uv} + f_{r1uv}e^{\beta_{1uv}+\delta_{1uv}z} + f_{r2uv}[(\alpha+1)e^{\beta_{1uv}+\delta_{1uv}z} - \alpha]} + (1-w)\right\},$$

where δ_{1uv} is an additional parameter to be estimated and y is the adjusted value of the covariate for that pair as with the autosomal models. The model extends easily to include more that one covariate; there is one additional parameter for each covariate.

Under a generalization of this model, the LOD score is

$$\log_{10}\left\{ w\frac{\hat{f}_{0uv} + \hat{f}_{1uv}e^{\beta_{1uv}+\sum_l \delta_{luv}z_l} + \hat{f}_{2uv}[(\alpha+1)e^{\beta_{1uv}+\sum_l \delta_{luv}z_l} - \alpha]}{f_{r0uv} + f_{r1uv}e^{\beta_{1uv}+\sum_l \delta_{luv}z_l} + f_{r2uv}[(\alpha+1)e^{\beta_{1uv}+\sum_l \delta_{luv}z_l} - \alpha]} + (1-w)\right\},$$

where l indexes the covariate. Note that covariates can only be included in the one-parameter model, with the constraint λ_{2ff} = (α+1) $\lambda_{1ff} - \alpha$.

Constraints on the δs are the same as with the autosomal models.

11.2.5 Parent-of-Origin Models

The expression of an allele may depend on the sex of the parent from whom the allele was inherited; this phenomenon is known as a parent-of-origin effect (alternatively, genetic imprinting). For example, individuals affected with the autosomal dominant condition Beckwith-Wiedemann syndrome almost always inherited the defective allele from their mother. Individuals who inherit the defective allele from their father are rarely affected with this disorder.

For the model that includes a parent-of-origin effect, the ARP lod score model fits separate parameters for the maternal and paternal effects. The test of parent-of-origin effect is obtained by comparing the likelihood-ratio statistics (i.e., 4.6 times the lod score) for the models with and without the parent-of-origin effect. The parent-of-origin model can only be applied to autosomal loci.

For the parent-of-origin model, the LOD score for a particular affected relative pair is

$$\log_{10}\left\{ w\frac{\sum\limits_{k=0,1m,1p,2} \hat{f}_k\lambda_k}{\sum\limits_{k=0,1m,1p,2} f_{rk}\lambda_k} + (1-w)\right\},$$

where m denotes maternal and p denotes paternal, so that the sum is over *k = 0, 1m, 1p, 2* rather than $k = 0, 1, 2$. As in previous models, λ_0= 1 and λ_k= exp(β_k), where β_k is the parameter estimated.

[2] As with the autosomal models, β in λ_{kuv} = exp(β_{kuv}) will be estimated instead of estimating λ itself.

11.2.5.1 One Parameter Model

First note that $\lambda_1 = \frac{\lambda_{1m}+\lambda_{1p}}{2}$. The one-parameter model employs the same mode-of-inheritance constraint, i.e., $\lambda_2 = (\alpha + 1)\ \lambda_1 - \alpha$.

11.2.5.2 Covariates

Covariates may be included only in the one-parameter model. Inclusion of a single covariate (z) gives $\lambda_{1m} = e^{\beta_{1m}+\delta_{1m}z}$ and $\lambda_{1p} = e^{\beta_{1p}+\delta_{1p}z}$ where δ_{1m} and δ_{1p} are the additional parameters to be estimated and z is the adjusted value of the covariate for that pair, as with the autosomal models. The model extends easily to include more that one covariate; there are two additional parameters for each covariate, so that $\lambda_{1m} = e^{\beta_{1m}+\sum_l \delta_{lm}z_l}$ and $\lambda_{1p} = e^{\beta_{1p}+\sum_l \delta_{lp}z_l}$, where l indexes the covariate under the generalization of this model. We include an option that fixes either λ_{1m} or λ_{2m} to be equal to 1. In such situations, only one covariate parameter is fitted for each covariate.

Constraints on the δs are the same as with the other autosomal models.

These models only apply to ASPs and affected half-sib pairs because the problem of computing the right IBD probabilities for other types of ARPs is daunting. By default, other types of ARPs are excluded from the analysis, but an option to include other types into the analysis is provided. (It should be recognized that, for other types of ARPs, parent-of-origin effects may be highly confounded with ascertainment.) When other types of ARPs are included in an analysis with a parent-of-origin effect λ_1 is replaced with $\{(\lambda_{1m} + \lambda_{1p})/2\}$ in other ARPs to avoid fitting an extra parameter. The only information about parent-of-origin effect in the models comes from the ASPs, and so parent-of-origin models will not be allowed if the number of ASPs in which $\hat{f}_{1m} \neq \hat{f}_{1p}$ is less than 10, even if many other ARPs are available.

11.3 Program Input

File Type	Description
LODPAL parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual including fields for identifiers, sex, parents, trait (including individual-specific covariate values) and marker data.
IBD sharing file	Stores identity-by-descent (IBD) distributions between pairs of related individuals at one or more marker loci.
Pair information file (Optional)	Contains character delimited records for each known relative pair including fields for identifiers, weights, and pair-specific covariate values.

Notes

To use the X-linked model in LODPAL, an IBD sharing file has to include “x_linked” after the name of the marker in the file header.

11.3.1 Parameter File Syntax

11.3.1.1 The `lodpal` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main LODPAL parameter.

`parameter [, attribute]`	**Explanation**	
`lodpal`	Initiates a particular LOD score analysis for affected relative pairs.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	lodpal
	Required	No
	Applicable Notes	None

11.3.1.2 The `lodpal` Block

The following syntax table specifies the permissible parameter and attribute settings for a LODPAL block.

`parameter [, attribute]`	**Explanation**
`trait`	Specifies a binary trait to be used in the current analysis. Value Range: Character string representing the name of a binary trait, phenotype or covariate listed in the pedigree data file. Default Value: None Required: Yes Applicable Notes: 1
`, cutpoint`	Traits that are not binary are dichotomized at this value. Value Range: (-∞, ∞) Default Value: 0 Required: No Applicable Notes: 1
`, conaff`	Specifies option to analyze affected relative pairs only. Value Range: N/A Default Value: None Required: No Applicable Notes: 1
`, condisc`	Specifies option to pool concordantly affected relative pairs with concordantly unaffected sib pairs, and include discordant sib pairs in the analysis. Value Range: N/A Default Value: None Required: No Applicable Notes: 1
`, noconunaff`	Specifies option to analyze concordantly affected relative pairs and discordant sib pairs. Value Range: N/A Default Value: None Required: No Applicable Notes: 1

`subset`	Specifies option to use only a subset of the data. The value given should be a binary trait used as an indicator variable.	
	Value Range	Character string representing the name of a binary trait listed in the pedigree data file.
	Default Value	None
	Required	No
	Applicable Notes	2
`marker`	Specifies a marker to be included in the current analysis.	
	Value Range	Character string representing the name of a marker listed in the pedigree data file.
	Default Value	None
	Required	No
	Applicable Notes	3
`covariate`	Specifies a covariate for the one-parameter model only.	
	Value Range	Character string representing the name of a covariate listed in the pedigree data file.
	Default Value	None
	Required	No
	Applicable Notes	4
`, power`	Covariate terms are taken to the power specified.	
	Value Range	(-∞, ∞)
	Default Value	1
	Required	No
	Applicable Notes	None
`, sum`	Specifies option to include the covariate sum.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	4
`, diff`	Specifies option to include the covariate difference.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	4
`, both`	Specifies option to include covariate terms: sum & difference.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	4

, prod	Specifies option to include the covariate product. Value Range: N/A Default Value: None Required: No Applicable Notes: 4
, avg	Specifies option to include the covariate average. Value Range: N/A Default Value: None Required: No Applicable Notes: 4
, single	Specifies option to include the covariate value for only the first member of a given pair. Value Range: N/A Default Value: None Required: No Applicable Notes: 4
, mean	Specifies option to center covariates around the observed mean or the user-supplied value. Value Range: (-∞, ∞) Default Value: observed mean Required: No Applicable Notes: 4
, minimum	Specifies option to set covariate values as offsets from the smallest observed value. Value Range: N/A Default Value: None Required: No Applicable Notes: 4
diagnostic	Specifies option to print diagnostic information to a separate file. Value Range: Character string representing the name of a marker or location listed in the IBD sharing file. Default Value: None Required: No Applicable Notes: 5
turn_off_default	Specifies option to disable the default maximization process. Value Range: N/A Default Value: None Required: No Applicable Notes: 6

`sib_pairs_only` `sib`	Specifies option to use only full sib-pairs in the analysis.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	None
`wide_out`	Specifies option to print more verbose output information. This causes some output tables to be more than 80 columns wide.	
	Value Range	true false
	Default Value	false
	Required	No
	Applicable Notes	7
`weight`	Specifies a variable name to be used as weight in the current analysis.	
	Value Range	Character string representing the name of a trait, phenotype or covariate listed in the pedigree data file.
	Default Value	None
	Required	No
	Applicable Notes	8
`pair_info_file`	Starts a sub-block for specification of pair-specific covariate(s) and/or weight values to be used in the current analysis.	
	Value Range	Character string representing a valid file name.
	Default Value	None
	Required	No
	Applicable Notes	9
`autosomal` `autosomal_model`	Starts a sub-block for specification of an autosomal model on an existing autosomal marker.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	10
`x_linkage` `x_linkage_model`	Starts a sub-block for specification of an X-linked model on existing X-linked markers.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	11

Notes

1. The value of a `trait` parameter should be set to the name of a binary `trait`, `phenotype` or `covariate` field read from the pedigree data file.

(a) If no valid `trait` parameters are listed, then all `trait` fields read in from the pedigree data file are used.

(b) If more than one `trait` is specified, then each will be used in a separate analysis.

(c) If a `trait` is not a binary trait, then it will be dichotomized at 0 (trait values $<= 0$ will be treated as unaffected and values > 0 will be treated as affected) or at the value of the `cutpoint` attribute. When dichotomizing a trait using a cutpoint, all values less than or equal to the cutpoint are considered unaffected and all values strictly greater than the cutpoint are considered to be affected.

(d) If a `trait` parameter has the `conaff` attribute, then the program selects only concordantly affected relative pairs (ARPs) and performs an analysis on these pairs.

(e) If a `trait` parameter has the `condisc` attribute, then the program pools the concordantly affected relative pairs with the concordantly unaffected sib pairs and performs a one-parameter model analysis in which these are analyzed together with the discordant sib pairs (DSP) by creating a covariate to indicate concordance status of the pairs.

(f) If a `trait` parameter has the `noconunaff` attribute, the program performs the same analysis as with the `condisc` attribute, but without including the concordantly unaffected sib pairs.

(g) When either the `condisc` attribute or the `noconunaff` attribute is used, no covariates can be included. If the user specifies any covariates, they are ignored by the program.

(h) If no attributes are listed, then by default an ARP analysis is performed.

2. The trait specified by a `subset` parameter should be a binary trait coded as 0 for individuals to be excluded from, and 1 for individuals to be included in, the analysis. The `subset` parameter may be included more than once, in which case the only individuals included in the analysis are those for which all the indicated binary traits are coded 1.

3. The value of a `marker` parameter should be set to the name of a marker (or marker location) for which IBD sharing information was generated and stored in the IBD sharing file. The `marker` parameter may be included more than once. If no valid marker parameters are listed then all markers are used.

4. The value of a `covariate` parameter should be set to the name of a `trait`, `phenotype` or `covariate` field read from the pedigree data file. The `covariate` parameter may be included more than once. A `covariate` parameter may have two attributes. One is to specify the function to compute a pair-specific value from two individual-specific values (`sum`, `diff`, `single`, `avg` or `prod`), and the other is to adjust the covariate value to impose genetic constraints on them (`mean` or `minimum`).

   (a) If `sum`, `diff`, `avg` or `prod` attributes are specified, then a single covariate sum, difference, average or product term of two individual-specific values is included as a pair-specific covariate value.

   (b) If the `both` attribute is specified, then both sum and difference terms are included.

   (c) If the `single` attribute is specified, then the covariate value for the first member of the pair is included as a pair-specific value.

   (d) If no attribute of this kind is specified, then the sum is included by default.

(e) If the `mean` attribute is specified (the default), then the program automatically centers each pair-specific covariate value before inclusion in the likelihood, using the sample mean or a user-supplied value (for example, `mean` = 0.5).

(f) If the `minimum` attribute is included, then the program automatically puts the offset from the smallest observed covariate value as the pair-specific covariate value into the likelihood, so that the smallest value of the pair-specific covariate equals zero.

5. The value of a `diagnostic` parameter should be set to the name of a marker or a location (in centiMorgans) for which IBD sharing information was generated and stored in the IBD sharing file. If the `diagnostic` parameter has a valid value, then an additional output file, "`LODPAL.lod`", will be generated that contains the individual pair LOD score contributions for the final model at the particular location specified by the `diagnostic` parameter value.

6. If the program finds the `turn_off_default` parameter, then the program maximizes the LOD score in a somewhat simpler way than the default way. By default, the program uses a method that avoids, as much as possible, spuriously high LOD scores. However, because there may be multiple true maxima, the result obtained using the `turn_off_default` parameter may also be of interest.

7. If the `wide_out` parameter is set to **true**, then additional columns are added to the output of the LOD score Analysis of Affected Relative Pairs. The information contained in the additional columns are :

   - detailed info on the number of pairs
   - first derivatives of parameter estimates
   - the number of iterations it took for maximization
   - the return flag from the MAXFUN library function

8. The value of a `weight` parameter should be set to the name of a `trait`, `phenotype` or `covariate` field read from the pedigree data file. The weight value for the first member of the pair is included as a pair-specific value. Weights must be between 0 and 1, inclusive (i.e., in [0,1]). Pairs with weights outside this interval will not be included in the analysis. If a pair-specific weight from the pair information file is to be specified, it must instead be specified within the `pair_info_file` sub-block using the `pair_weight` parameter (see note 9). Only one weight, specified either by the `weight` parameter or by the `pair_weight` parameter in `pair_info_file` sub-block (see note 9) may be included; and weights are given values of 1 for all pairs in the analysis by default, if neither a `weight` parameter nor a `pair_weight` parameter is found.

9. If the program finds the `pair_info_file` parameter with a valid file name, then the program uses the pre-constructed pair-specific covariate and/or weight values from the file name specified. The `pair_info_file` parameter may have its own sub-block to specify the name of the pair-specific covariate(s) and/or weight to be used in the current analysis.

10. If the program finds the `autosomal` or `autosomal_model` parameter, then the program uses the specified autosomal model for the autosomal locations. The `autosomal` parameter may have its own sub-block to specify the model to be used in the current analysis. If no sub-block is found, the default autosomal model will be used, i.e., the one-parameter model with the default `alpha` value, and without parent-of-origin effect.

11. If the program finds the `x_linkage` or `x_linkage_model` parameter, then the program uses an X-linked model for the X-linked markers. The `x_linkage` parameter may have its own sub-block to specify the model to be used in the current analysis. If no sub-block is found, the default X-linked model will be used, in which all three λ_1 parameters are constrained to be equal, and λ_{2ff} is fixed.

11.3.1.3 The `pair_info_file` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for a `pair_info_file` sub-block.

`parameter [, attribute]`	**Explanation**
`pair_covariate`	Specifies a variable name to be used as a covariate in the current test.
	Value Range: Character string representing the name of a trait, phenotype or covariate listed in the pair information file.
	Default Value: None
	Required: No
	Applicable Notes: 1
`, mean`	Specifies option to center covariates around the observed mean or the user-supplied value.
	Value Range: (-∞, ∞)
	Default Value: observed mean
	Required: No
	Applicable Notes: 3
`, minimum`	Specifies option to set covariate values as offsets from the smallest observed value.
	Value Range: N/A
	Default Value: None
	Required: No
	Applicable Notes: 3
`pair_weight`	Specifies a variable name to be used as a weight in the current test.
	Value Range: Character string representing the name of a trait, phenotype or covariate listed in the pair information file.
	Default Value: None
	Required: No
	Applicable Notes: 2

Notes

1. The value of a `pair_covariate` parameter should be set to the name of a covariate field read from the Pair Information File. The `pair_covariate` parameter may be included more than once.

2. The value of a `pair_weight` parameter should be set to the name of a `weight` field read from the Pair Information File. Weights must be in the interval [0, 1]. Pairs with weights outside this interval will not be included in the analysis. Only one `pair_weight` parameter may be included, and by default weights are given values of 1 for all pairs if no

`pair_weight` parameter is specified. If the `pair_info_file` parameter does not have a sub-block, then the program uses every field listed in the `pedigree` block, with the exception of ID fields, as a mean-centered covariate in the analysis. The `pair_info_file` parameter is ignored for the two-parameter model unless `pair_weight` is found in this sub-block.

3. The value of a `pair_covariate` parameter should be set to the name of a `trait`, `phenotype` or `covariate` field read from the pedigree data file. The `pair_covariate` parameter may be included more than once. A `pair_covariate` parameter may have an attribute to adjust the covariate value to impose genetic constraints on them (`mean` or `minimum`).

   (a) If the `mean` attribute is specified (the default), then the program automatically centers each pair-specific covariate value before inclusion in the likelihood, using the sample mean or a user-supplied value (for example, `mean` = 0.5).

   (b) If the `minimum` attribute is included, then the program automatically puts the offset from the smallest observed covariate value as the pair-specific covariate value into the likelihood, so that the smallest value of the pair-specific covariate equals zero.

11.3.1.4 The `autosomal` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for a `autosomal` sub-block.

`parameter [, attribute]`	**Explanation**
`model`	Specifies the type of model to use in the analysis. Value Range: {one_parameter, two_parameter} Default Value: one_parameter Required: No Applicable Notes: 1
`, uncon` `, unconstrained`	Specifies that no genetic constraints on the parameters are to be estimated. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, alpha`	Specifies the Whittemore and Tu one-parameter models alpha value. An `alpha` value of = 1 specifies a model with no dominant genetic variance. Value Range: $(1, \infty)$ Default Value: 2.634 Required: No Applicable Notes: None
`parent_of_origin`	Specifies the option to test for the parent-of-origin effect. By default, only sib-pairs are used in the analysis. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`, fixed`	Specifies the option to fix either λ_{1m} or λ_{1p} to 1. Value Range: {maternal, paternal} Default Value: None Required: No Applicable Notes: 2
`, all_pairs`	Specifies the option to include non-sibs in the analysis Value Range: N/A Default Value: None Required: No Applicable Notes: 3

Notes

1. The value **one_parameter** specifies the Whittemore and Tu one-parameter model. The value **two_parameter** specifies the two-parameter model. The two-parameter model does not allow

the inclusion of covariate data. If **two_parameter** is specified, any covariate parameter is ignored and no covariates are included in the analysis.

2. A `fixed` value of **maternal** sets λ_{1m} equal to 1, and a `fixed` value of **paternal** sets λ_{1p} equal to 1.

3. By default, other types of affected relative pairs are excluded from the analysis because the problem of computing the right IBD sharing probabilities for other types of affected relative pairs is daunting. If the `all_pairs` attribute is specified, then other types of affected relative pairs are included in an analysis with parent-of-origin effect test for affected sib pairs. When other types of affected relative pairs are included, λ_1 will be replaced with $\{(\lambda_{1m} + \lambda_{1p})/2\}$ in other affected relative pairs to avoid fitting an extra parameter.

11.3.1.5 The `x_linkage` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for a `x_linkage` sub-block.

`parameter [, attribute]`	**Explanation**
`pair_type`	Specifies the sex-specific pair types to be included in the analysis. Multiple pair_type statements can be specified to include more than one pair type. Value Range: {M-M, M-F, F-F, all} Default Value: all Required: No Applicable Notes: None
`lambda1_equal`	Specifies λ_{1mm} = λ_{1mf} = λ_{1ff} regardless of sexes of the pair in the current test. When set to **false**, all three λ_1s are estimated separately. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
`lambda2_fixed`	Specifies λ_{2ff} to be dependent on λ_{1ff}. When set to **false**, λ_{1ff} and λ_{2ff} are estimated separately. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
`, alpha`	The alpha value to compute λ_{2ff} when it is dependent on λ_{1ff}. This is ignored when `lambda2_fixed` is set to **false**. Value Range: (-∞, ∞) Default Value: 2.634 Required: No Applicable Notes: None

The following are all valid LODPAL statements:

```
lodpal, multipoint      # Test all markers
lodpal, multipoint      # Equivalent to the previous statement.
{
}

lodpal, singlepoint {
   marker    = M1
   covariate = ageexam, minimum, diff
}

lodpal, multipoint {
   trait = T1
   autosomal_model {
      model = two_parameter
   }

   diagnostic = "20 44.0" # The additional output file for location
                          # "20 44.0" is generated.
}

lodpal, multipoint,out="t1condisc.out" {
   trait = T1,condisc
   turn_off_default        # Turn off default maximization process
}

lodpal, multipoint {            # The covariate is ignored.
   trait = T1, noconunaff       # Analysis is done with the one-
   covariate = ageexam          # parameter model using default alpha.
}

lodpal,multipoint {
   trait = T1
   pair_info_file = "cov.in" {
      pair_weight    = probability
      pair_covariate = covariate1, mean
   }
}

lodpal,multipoint {
   trait = T1
   x_linkage {
      pair_type     = "M-M"
      lambda1_equal = false  # All three lambdas are estimated separately.
      lambda2_fixed = false  # The data set has to have at least 15
                             # sister-sister pairs and at least 15
                             # female-female pairs other then sister
                             # -sister pairs to use this model
   }
}

lodpal,multipoint {
   trait = T1
   x_linkage {
```

```
      lambda2_fixed = true, alpha = 3.5  # The same as default model,
                                         # but the different alpha value
                                         # is used.
   }
}

lodpal, multipoint {
   trait = T1
   autosomal {
      model            = one_parameter, alpha=2.634
      parent_of_origin = true, fixed=maternal
   }
}

lodpal,multipoint {
   trait=T1
   autosomal {
      model            = one_parameter, alpha=2.634
      parent_of_origin =true, all_pairs
   }
}

lodpal,multipoint {
   trait = T1
   autosomal {
      model = one_parameter, uncon
   }
}

lodpal,multipoint {
   trait = T1
   autosomal {
      model = two_parameter, unconstrained
   }
}
```

11.3.2 Pair Information File

The pair information file is a character delimited file that stores the pre-constructed pair-specific covariate and/or weight values for the pairs to be used in the analysis. The first line of the file is the header that contains the name of each field, and the rest of the file contains the line for each pair with the required IDs, covariate(s) and/or weight fields. The pedigree ID (PEDID in the example below), first individual ID (ID1 in the example), and second individual ID (ID2 in example) fields are required in that order, and the weight and covariate fields can be in any order. Each individual is expected to be found in the pedigree data file, and the pairs are expected to be found in the IBD sharing file, for the analysis to proceed. Any individual or pair that is not in both of these files will be ignored. The weight and covariate values should be numerics, and no missing values are allowed.

A pair information file may look like the following:

```
PEDID ID1 ID2 probability covariate1
1       3   4   0.0033619  0.0033619
102     3   6   0.0114638  0.0000000
```

```
102     6   7   0.0022620  0.3283151
102     3   7   0.0162358  0.0000000
104     5   6   0.9802018  0.0000000
105     6   7   0.0135131  0.9079691
106     3   4   0.8125513  0.0334500
107     7   8   0.9497964  0.0006405
.
.
.
```

Another Pair Information File may look like:

```
PEDID, ID1, ID2, probability, covariate1
1, 3, 4, 0.0033619, 0.0033619
102, 3, 6, 0.0114638, 0.0000000
102, 6, 7, 0.0022620, 0.3283151
102, 3, 7, 0.0162358, 0.0000000
104, 5, 6, 0.9802018, 0.0000000
105, 6, 7, 0.0135131, 0.9079691
106, 3, 4, 0.8125513, 0.0334500
107, 7, 8, 0.9497964, 0.0006405
.
.
.
```

11.4 Program Execution

LODPAL is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running LODPAL from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>LODPAL
S.A.G.E. v5.x -- LODPAL
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: ./LODPAL <parameters> <pedigree> <IBD>
Command line parameters:
  parameters   - parameter file
  pedigree     - pedigree data file
  IBD          - IBD sharing file
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of LODPAL may look like the following:

```
>LODPAL par example.ped example.ibd
S.A.G.E. v5.x -- LODPAL
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading Parameter File...................done.
Reading pedigree data.....................
              from example.ped............done.
Sorting pedigrees........................done.
Reading pairs............................done.
Sorting pairs............................done.
  No analyses specified.
Performing default LODPAL analysis........done.
Analysis complete!
```

11.5 Program Output

LODPAL produces several output files that contain results and diagnostic information:

Filename	File Type	Description
lodpal.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
lodpal.out lodpal.xln	Pair analysis output file	Contains tables of LOD scores and parameter estimates. For autosomal markers the extension is 'out', for X-linked markers it is 'xln'.
lodpal.lod	Diagnostic output file	Contains tables of individual LOD score contributions and other variables at a particular location.

11.5.1 Information Output File

The LODPAL information output file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read in, are included with all programs in the S.A.G.E. Release and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not run as expected. The file "lodpal.inf" should be checked for errors and diagnostic information after each run of the program.

11.5.2 Pair Analysis Output File

One pair analysis output file, named either "LODPAL.out" or "LODPAL.xln", is generated per run of LODPAL. It contains tables of LOD scores and parameters estimates for each marker location tested.

11.5.3 Diagnostic Output File

One diagnostic output file, named "LODPAL.lod" by default, is generated per run of LODPAL when a valid diagnostic location (i.e., marker) has been specified by the user. It contains a table of individual LOD score contributions, covariates, and allele-sharing probabilities at the specified location, along with a variance-covariance matrix of the parameter estimates (assuming independence of all pairs) and a histogram of the individual LOD score contributions.

11.6 Example Output Files

11.6.1 Pair Analysis Output File

Here is a typical example of a LODPAL output table.

```
S.A.G.E. v5.x -- LODPAL
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Wed Oct 10 10:40:33 2001
================================================================
  Conditional Logistic Analysis of
    Affected Relative Pairs - multipoint
================================================================
    Trait    : affection
               concordantly affected relative pairs
    Covariate: cov1
               sum of two individual covariate values
               mean centered
               mean before adjusting = 0.232673
               mean after adjusting  = 0.000000
               std. deviation        = 0.423585
    Method   : default analysis method
    Model    : one-parameter model, constrained, alpha = 2.634
================================================================
                            Full           Parameter Estimates
                  LOD       Sib    All     --------------------
MARKER      cM    SCORE     Pairs  Pairs   Beta1     cov1
----------------- --------- ------ ------  --------- ---------
 20s103    ------  0.080913    117    202   0.053795 -0.053795
 20_2.0       2.0  0.064902    117    202   0.051271 -0.051271
 20_4.0       4.0  0.048328    117    202   0.045973 -0.045973
 20_6.0       6.0  0.033579    117    202   0.038439 -0.038439
 20_8.0       8.0  0.022750    117    202   0.030557 -0.030557
 20s482    ------  0.019024    117    202   0.027139 -0.027139
 20_10.0     10.0  0.020959    117    202   0.029545 -0.029545
 20_12.0     12.0  0.025838    117    202   0.035060 -0.035060
 20_14.0     14.0  0.031819    117    202   0.041004 -0.041004
 20_16.0     16.0  0.037990    117    202   0.046230 -0.046230
 20_18.0     18.0  0.042800    117    202   0.049278 -0.049278
 20_20.0     20.0  0.044706    117    202   0.049128 -0.049128
 20_22.0     22.0  0.043029    117    202   0.045823 -0.045823
 20_24.0     24.0  0.038242    117    202   0.040305 -0.040305
 20s851    ------  0.035044    117    202   0.037096 -0.037096
 20_26.0     26.0  0.061769    117    202   0.052506 -0.052506
 20_28.0     28.0  0.130234    117    202   0.081308 -0.081308
.
.
.
================================================================
```

11.6.2 Diagnostic Output File

Here is a typical example of a LODPAL diagnostic output table.

```
S.A.G.E. v5.x -- LODPAL
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Wed Aug 29 13:30:52 2001
================================================================================
  Conditional Logistic Analysis of Affected Relative Pairs - multipoint
================================================================================
    Trait    : affection
               concordantly affected relative pairs
    Covariate: cov1
               sum of two individual covariate values
               mean centered
               mean before adjusting = 0.232673
               mean after adjusting  = 0.000000
               std. deviation        = 0.423585
    Method   : default analysis method
    Model    : one-parameter model, constrained, alpha = 2.634
    Location : 20_36.0
================================================================================
# Final Result Summary
    Parameter Estimates:
      1. beta 1 =  0.121141
      2. cov1(delta 1) =  0.132373
    Variance-Covariance Matrix(assuming independent pairs):
      ------------------------------------
      |    \     |     1     |     2     |
      ------------------------------------
      |     1    |  0.017046 |  0.003377 |
      ------------------------------------
      |     2    |  0.003377 |  0.001245 |
      ------------------------------------
# Histogram of Individual LOD Score Contributions
    Maximum LOD Score =  0.5936
    Minimum LOD Score = -0.4504
    Bin Size          =  0.1044
    Interval            Count (one * is equal up to 2 pair(s).)
--------------------------------------------------------------------------------
-0.4505  to  -0.3461       2 *
-0.3461  to  -0.2417       4 **
-0.2417  to  -0.1372       7 ****
-0.1372  to  -0.0328      47 ************************
-0.0328  to   0.0716      89 *********************************************
 0.0716  to   0.1760      35 ******************
 0.1760  to   0.2805      13 *******
 0.2805  to   0.3849       4 **
 0.3849  to   0.4893       0
 0.4893  to   0.5937       1 *
--------------------------------------------------------------------------------
               Total :  202
# Individual LOD Score Contribution
FAMID    IDSIB1   IDSIB2   F0         F2          Cov. prin2       LOD SCORE CONTRIBUTION
-------- -------- -------- ---------- ----------  ---------------  ---------------
1        3        4         0.0033619  0.0033619     3.5349345       -0.0715001694
102      3        6         0.0114638  0.0000000     0.9441527        0.0490763128
102      6        7         0.0022620  0.3283151     1.1912290        0.0415850027
102      3        7         0.0162358  0.0000000     1.7190322        0.0670693326

...
108      4        5         0.0051893  0.0271982    -4.2598469        0.0957374556
109      3        4         0.0028969  0.1407563    -1.0821854        0.0003981387
-------- -------- -------- ---------- ----------  ---------------  ---------------
Total Pair Count = 202                              Total LOD Score =  3.3944467415
================================================================================
```

Chapter 12

LODLINK

LODLINK performs model-based lod score calculations for two-point linkage between a main trait and each of the other markers in the pedigree file. The main trait may be a marker or a trait that follows Hardy Weinberg proportions, Mendelian transmission and has either two or three types. In the latter case, output from SEGREG can be used as input. LODLINK uses the genotype/phase elimination algorithms proposed by Lange and Boehnke (1983) and Lange and Goradia (1987), together with other enhancements, to perform fast linkage calculations

12.1 Limitations

Pedigrees may not contain loops or marriage rings.

12.2 Theory

12.2.1 Computation of the Likelihood and Lod Scores

Let $T_1, ..., T_k$ be the alleles at the trait locus, $q_{T_1}, ..., q_{T_k}$ be the corresponding frequencies, $M_1, ..., M_m$ be the alleles at the marker locus, and $q_M, ..., q_{M_m}$ be the corresponding allele frequencies.

Define

1. a phased joint genotype to be: $\frac{T_b M_c}{T_d M_e}$ where b, d = 1, ...k; c, e = 1, ..., m;
2. the probability of a joint genotype in the population to be:

$$\psi\left(\frac{T_b M_c}{T_d M_e}\right) = C\psi(T_b T_d)\psi(M_c M_e),$$

where

$$C = \begin{cases} 1 & , if\, T_b = T_d \, or \, M_c = M_e \\ \frac{1}{2} & , otherwise \end{cases}$$

and

$\psi(\mathrm{T}_b\mathrm{T}_d)$ = probability of trait genotype in the population,

$\psi(\mathrm{M}_c\mathrm{M}_e)$ = probability of marker genotype in the population;

3. the transmission probability to be:

$$\tau_S\left(\frac{T_bM_c}{T_dM_e} \rightarrow T_fM_g\right) = Pr\left(\begin{array}{c} parent\ of\ sex\ s\ and\ genotype \frac{T_bM_c}{T_dM_e} \\ transmits\ haplotype\ T_fM_g\ to\ child \end{array}\right)$$

$$= \frac{(1-\theta_S)(\delta_{T_bT_f}\delta_{M_cM_g} + \delta_{T_dT_f}\delta_{M_eM_g})}{2} + \frac{\theta_S(\delta_{T_bT_f}\delta_{M_eM_g} + \delta_{T_dT_f}\delta_{M_cM_g})}{2},$$

where θ_S is the sex-dependent recombination fraction between the trait and marker loci ($\theta_S = \theta_{male}$ or θ_{female}) and

$$\delta_{xy} = \begin{cases} 1 & , if\ x = y \\ 0 & , if\ x \neq y \end{cases}$$

4. the transition probability to be:

$$\Pr\left(\begin{array}{c} Mother\ with\ genotype\ \frac{T_bM_c}{T_dM_e}\ and\ Father\ with\ genotype\ \frac{T_rM_s}{T_uM_v} \\ have\ a\ child\ with\ genotype\ \frac{T_fM_g}{T_hM_j} \end{array}\right)$$

$= \tau_{female}\left(\frac{T_bM_c}{T_dM_e} \rightarrow T_fM_g\right)\tau_{male}\left(\frac{T_rM_s}{T_uM_v} \rightarrow T_hM_j\right), if\ T_f = T_h\ and\ M_g = M_j$
or
$= \tau_{female}\left(\frac{T_bM_c}{T_dM_e} \rightarrow T_fM_g\right)\tau_{male}\left(\frac{T_rM_s}{T_uM_v} \rightarrow T_hM_j\right) +$
$\tau_{female}\left(\frac{T_bM_c}{T_dM_e} \rightarrow T_hM_j\right)\tau_{male}\left(\frac{T_rM_s}{T_uM_v} \rightarrow T_fM_g\right), otherwise.$

For a joint genotype $\mathrm{T}_b\mathrm{M}_c/\mathrm{T}_d\mathrm{M}_e$ of pedigree member i, let the separate one-locus genotypes be denoted $\mathrm{u}_i = \mathrm{T}_b\mathrm{T}_d$ for the trait and $\mathrm{v}_i = \mathrm{M}_c\mathrm{M}_e$ for the marker. Let y_i be the trait phenotype and m_i be the marker phenotype (discrete). Let w_{male_i}, w_{female_i} and w_i be the joint genotypes of the father of individual i, mother of individual i, and individual i respectively. The likelihood for a pedigree of n persons is

$$L(\theta) = \sum_{w_1} ... \sum_{w_n} \prod_{i=1}^{n} H_i,$$

where

$$H_i = \begin{cases} p_i(w_{female_i}, w_{male_i}, w_i) & , if\ i\ is\ missing \\ p_i(w_{female_i}, w_{male_i}, w_i)g_{u_i}(y_i)g_{v_i}(m_i) & , otherwise \end{cases}$$

in which

$$p_i(w_{female_i}, w_{male_i}, w_i) = \begin{cases} Tr(w_{female_i}, w_{male_i}, w_i), & if\ the\ parents\ of\ i\ are\ in\ the\ pedigree \\ \psi(w_i), & otherwise, \end{cases}$$

$g_{v_i}(m_i)$ = probability of marker phenotype m_i given marker genotype v_i (assumed to be always 0 or 1).

$g_{u_i}(y_i)$ = probability (density) of trait phenotype y_i conditional on genotypes u_i and possibly other factors. These can be obtained as output from SEGREG by specifying `type_prob` = **true** in the SEGREG `output_options` sub-block.

Lod scores are defined as

Z(θ) = Log_{10}L(θ) - Log_{10}L(0.5).

12.2.2 Estimation of Parameters

When estimating the recombination fraction θ, maximum likelihood estimates of θ are obtained as the values that make the likelihood largest in the parameter space [0, 0.5]. If a larger likelihood exists for θ in the parameter space [0, 1], the corresponding estimate(s) are also given.

When estimating both the recombination fraction θ and the proportion of linked families, α, maximum likelihood estimates are obtained over the range of parameter values indicated in the output.

12.2.3 Hypothesis Tests

12.2.3.1 Maximum Lod Score Test for Linkage

If we are estimating recombination fractions with $\theta_{male} = \theta_{female}$, then the asymptotic chi-square statistic calculated is

$$\chi_1^2 = 2[\log_e L(\hat{\theta}) - \log_e L(0.5)]$$

and the corresponding p-value quoted is

$$1 - \Phi\left(\sqrt{\chi_1^2}\right),$$

where Φ is the standard cumulative normal distribution. The upper bound of the p-value is calculated as $\frac{1}{10^{z(\hat{\theta})}}$. The p-value and upper bound are quoted only if $0 \leq \hat{\theta} < 0.5$.

If we are calculating the recombination fractions for males and females separately, the chi-square statistic calculated is

$$\chi_2^2 = 2[\log_e L(\hat{\theta}_{male}, \hat{\theta}_{female}) - \log_e L(0.5, 0.5)]$$

The corresponding p-value quoted as corresponding to this lod score is calculated on the assumption that the estimates $\hat{\theta}_{male}$ and $\hat{\theta}_{female}$ are independent, i.e. assuming that, under the null hypothesis $\hat{\theta}_{male} = \hat{\theta}_{female} = 0.5$, $2\log_e 10 \times$ (maximum lod) is distributed as $\frac{1}{4} + \frac{1}{2}\chi_1^2 + \frac{1}{4}\chi_2^2$. The upper bound of the p-value is calculated as

$$\frac{1}{10^{z(\hat{\theta}_{male}, \hat{\theta}_{female})}}.$$

The p-value and upper bound are quoted only if $0 \leq \hat{\theta}_{male}, \hat{\theta}_{female} < 0.5$.

12.2.3.2 Cleves and Elston's (1997) Likelihood Ratio Test for Linkage

Let $L(\hat{\theta}_{male}, \hat{\theta}_{female})$ be the likelihood evaluated at the maximum likelihood estimates $\hat{\theta}_{male}, \hat{\theta}_{female}$ and L($\tilde{\theta}_{male}, \tilde{\theta}_{female}$) be the likelihood estimated at the values $\tilde{\theta}_{male}, \tilde{\theta}_{female}$ that maximize the likelihood under the constraint $\tilde{\theta}_{male} + \tilde{\theta}_{female} = 1$. Then the asymptotic chi-square statistic calculated is

$$\chi_1^2 = 2[\log_e L(\hat{\theta}_{male}, \hat{\theta}_{female}) - \log_e L(\tilde{\theta}_{male}, \tilde{\theta}_{female})]$$

and the corresponding p-value quoted is

$$1 - \Phi\left(\sqrt{\chi_1^2}\right),$$

where Φ is the standard cumulative normal distribution. If both $\hat{\theta}_{male}\, and\, \hat{\theta}_{female}\, are > 0.5$, no p-value is calculated.

12.2.3.3 Morton's (1956) Likelihood Ratio Test for Homogeneity of the Recombination Fraction

Let $\sum_{i=1}^{n} \log_e L_i(\hat{\theta}_i)$ be the maximum log likelihood over n groups of pedigrees with $\hat{\theta}_i$ estimated separately for each group, and let $\sum_{i=1}^{n} \log_e L_i(\hat{\theta})$ be the maximum log likelihood over the n groups with a common $\hat{\theta}$ estimated; then the asymptotic chi-square statistic is

$$2[\sum_{i=1}^{n} \log_e L_i(\hat{\theta}_i) - \sum_{i=1}^{n} \log_e L_i(\hat{\theta})], \quad \begin{array}{r} with\, n-1\, degrees\, of\, freedom\, if\, \theta_{male} = \theta_{female} \\ 2(n-1)\, degrees\, of\, freedom\, if\, \theta_{male} \neq \theta_{female} \end{array},$$

The "asymptotic p-value" is the p-value based on the statistic following a chi-square distribution.

12.2.3.4 Smith's (1963) Test for Homogeneity of the Recombination Fraction

Let $\theta < 0.5$ be the recombination fraction in a proportion α of the families, and suppose there is no linkage in the remaining 1 - α of the families. Define the log likelihood of the i-th family as $\log_e L_i(\alpha,\theta) = \log_e[\alpha\, L_i(\theta) + (1 - \alpha) L_i(0.5)]$. Under the model $0 \leq \alpha \leq 1$, and $0 \leq \theta \leq 0.5$, we test the null hypothesis $\alpha = 1$.

Let $\sum_{i=1}^{n} \log_e L_i(\hat{\alpha}, \hat{\theta})$ be the maximum log likelihood over *n* constituent pedigrees with α and θ estimated, and $\sum_{i=1}^{n} \log_e L_i(1, \hat{\theta})$ be the maximum log likelihood over n constituent pedigrees with α = 1 and θ estimated.

If $\hat{\theta}$ is scalar (i.e., we assume $\theta_{male} = \theta_{female}$) then the asymptotic chi-square statistic for heterogeneity versus homogeneity is

$$\chi_1^2 = 2[\sum_{i=1}^{n} \log_e L_i(\hat{\alpha}, \hat{\theta}) - \sum_{i=1}^{n} \log_e L_i(1, \hat{\theta})], \; and \; the \, one \, sided \, p - value \, is \, 1 - \Phi(\sqrt{\chi_1^2}),$$

where Φ is the standard cumulative normal distribution.

If $\theta_{male} = \theta_{female}$ is not assumed, so that both $\hat{\theta}_{male} \, and \, \hat{\theta}_{female}$ are estimated, the chi-square statistic is compared to the chi-square distribution with 2 degrees of freedom and the asymptotic p-value is "two-sided".

12.2.3.5 Faraway's (1993) Test for Linkage Under Smith's (1963) Heterogeneity Model.

The asymptotic "chi-square" for linkage in the presence of heterogeneity is

$$2[\sum_{i=1}^{n} \log_e L_i(\hat{\alpha}, \hat{\theta}) - \sum_{i=1}^{n} \log_e L_i(0.5)],$$

for which the p-value is obtained on the assumption that this statistic is distributed as the maximum of two independent chi-square variables, each with one degree of freedom.

If $\theta_{male} = \theta_{female}$ is not assumed, the "chi-square" statistic is assumed to be distributed as the maximum of two independent chi-square variables, each with 2 degrees of freedom, and the asymptotic p-value quoted is "two-sided".

The posterior probability that the i-th family belongs to the linked type, given the observations, is computed as

$$w_i(\hat{\alpha}, \hat{\theta}) = \frac{\hat{\alpha} L_i^*(\hat{\theta})}{\hat{\alpha} L_i^*(\hat{\theta}) + 1 - \hat{\alpha}},$$

where

$$L_i^* = \frac{L_i}{L_i(0.5)}.$$

$w_i(\widehat{\alpha}, \widehat{\theta}) > \widehat{\alpha}$ indicates that the i-th family contains evidence for linkage.

12.2.4 Conditional Trait Genotype Probabilities

The table in the detail file headed "Individual Genotype Probabilities" gives, for each pedigree member, the probabilities of having genotypes *bd* conditional on that member's output marker phenotype and assuming maximum likelihood estimates of the recombination fraction (or fractions, sex specific), assuming homogeneity across pedigrees, i.e., expressing $L(\theta)$ as a function of the two locus genotypes *bc/de* (*bd* for the trait and *de* for the marker), *L(bc/de)*,

$$P_{bd} = \frac{\sum_{all\ ce} L(bc/de)}{\sum_{all\ bd} \sum_{all\ ce} L(bc/de)}$$

where, by default, *bd* = *AA*, *AB* or *BB* as in SEGREG.

12.3 Program Input

File Type	Description
LODLINK Parameter File	Specifies the parameters and options with which to perform a particular analysis.
Pedigree Data File	Contains delimited records for each individual including fields for identifiers, sex, parents, and trait data.
Marker Locus Description File[a]	Lists the alleles, allele frequencies and phenotype to genotype mapping for each marker locus.
Trait Locus Description File[b]	Lists the alleles, allele frequencies and phenotype to genotype mapping for each trait marker.
Trait Genotype Probability File	Produced by SEGREG and has a ".typ" extension. Lists the trait-marker penetrance functions for each individual.

[a] If an allele frequency for a particular individual is zero, then the likelihood for that individual's pedigree will be zero, and the pedigree will effectively be skipped during analysis.

[b] Both the Trait Locus Description File and the Type File are optional. One, but not both, may be used for LODLINK input.

12.3.1 Parameter File Syntax

12.3.2 The `lodlink` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main LODLINK parameter.

parameter [, attribute]	**Explanation**	
lodlink	Starts a LODLINK parameter block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
, out	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	lodlink_analysis
	Required	No
	Applicable Notes	None

12.3.2.1 The `lodlink` Block

The following syntax table specifies the permissible parameter and attribute settings for the `lodlink` block.

parameter [, attribute]	**Explanation**
`title`	Specifies a title for the analysis defined within the parameter block. Value Range: Quoted character string. Default Value: "LODLINK Analysis" Required: No Applicable Notes: None
`model`	Specifies model for linkage calculations. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, trait`	Model name in ".typ" file generated by SEGREG when `type_prob` = **true** in the SEGREG `output_options` sub-block, or the name of a valid trait marker. Linkage is calculated between each marker and this locus. Value Range: Character string representing a valid model name, or the name of a valid trait marker. Default Value: None Required: No Applicable Notes: 1
`, marker`	Marker against which all others are tested for linkage. Value Range: Character string representing a valid marker name. Default Value: None Required: No Applicable Notes: 1
`linkage_tests`	Specifies option to perform linkage tests. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 2
`, sex_specific`	Specifies option to use sex-specific recombination fractions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2

, homog	Specifies option to assume linkage homogeneity. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 2
homog_tests	Starts a sub-block that specifies tests for linkage homogeneity. Value Range: N/A Default Value: None Required: No Applicable Notes: 3
lods	Starts a sub-block that specifies lod score calculations. Value Range: N/A Default Value: None Required: No Applicable Notes: 4
genotypes	Specifies option to calculate genotype probabilities. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
, sex_specific	Specifies option to use sex-specific recombination fractions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None

Notes:

1. A value for either a `trait` or `marker` must be specified, but both should not be specified.

2. Linkage tests are performed according to the following table, depending on the values assigned to the `sex_specific` and `homog` attributes of the `linkage_tests` parameter.

	Sex-specific Recombination Fractions	
Homogeneity Assumed	true	false
true	Lod Score test Cleves-Elston test	Lod Score test
false	Faraway's test	Faraway's test

3. The default is to perform no linkage homogeneity tests. Otherwise a `homog_tests` sub-block must be included.

4. The default is to calculate lod scores for the following non-sex-specific recombination fractions: 0, .01, .05, 0.1, 0.2, 0.3 and 0.4. Otherwise a `lods` sub-block must be included.

12.3.2.2 The `homog_tests` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `homog_tests` sub-block.

`parameter [, attribute]`	**Explanation**
`smiths_test`	Specifies option to perform Smith's test for linkage homogeneity. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`, sex_specific`	Use sex-specific recombination fractions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`mortons_test`	Starts a sub-block that specifies Morton's test for linkage homogeneity. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`, sex_specific`	Use sex-specific recombination fractions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None

12.3.2.3 The `mortons_test` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `mortons_test` sub-block.

`parameter [, attribute]`	**Explanation**	
`group`	This sub-block specifies groups of pedigree IDs to be used for Morton's test. The value of the group parameter is the name of the group. This parameter may be specified as many times as necessary.	
	Value Range	Character string that uniquely names a pedigree group.
	Default Value	None
	Required	No
	Applicable Notes	1, 2

Notes

1. If no groups are specified, each pedigree is its own group.
2. Each pedigree must be listed in one, and only one group in the `group` sub-block described below.

12.3.2.4 The `group` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `group` sub-block.

`parameter [, attribute]`	**Explanation**	
`pedigree_id`	This sub-block specifies groups of pedigree IDs to be used for Morton's test. This parameter may be specified as many times as necessary to describe the group.	
	Value Range	Character string representing a valid pedigree ID.
	Default Value	None
	Required	No
	Applicable Notes	1, 2

Notes:

1. Required if `group` parameter is specified.
2. Example:

```
lodlink {
   model, trait  = T1
   linkage_tests = false

   homog_tests {
      smiths_test  = false #explicitly set to the default value
      mortons_test = true, sex_specific = false {
         group = 1 {
            pedigree_id = 1
            pedigree_id = 2
            pedigree_id = 3
            pedigree_id = 4
            pedigree_id = 5
         }

         group = 2 {
            pedigree_id = 6
            pedigree_id = 7
            pedigree_id = 8
         }
      }
   }

   lods {
      option = none
   }
}
```

12.3.2.5 The `lods` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `lods` sub-block.

`parameter [, attribute]`	**Explanation**
`option`	Specifies calculation option. Value Range: {none, standard, specified} Default Value: standard Required: No Applicable Notes: 1
`sex_specific`	Specifies option to use sex-specific recombination fractions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`male_female`	Starts a sub-block for specifying sex-specific recombination fractions at which lods will be calculated. Value Range: N/A Default Value: None Required: No Applicable Notes: 2
`average`	Starts a sub-block for specifying sex-averaged recombination fractions at which lods will be calculated. Value Range: N/A Default Value: None Required: No Applicable Notes: 3

Notes

1. If **none** is specified, no lod scores will be calculated. If **standard** is specified, lod scores will be calculated for the following non-sex-specific recombination fractions: 0, .01, .05, 0.1, 0.2, 0.3 and 0.4. If **specified** is specified, the desired recombination fractions must be specified using `male_female` or `average` sub-blocks for sex-specific or sex-averaged recombination fractions, respectively.

2. Required if the `option` parameter is set to **specified** and the `sex_specific` parameter is set to **true**.

3. Required if the `option` parameter is set to **specified** and the `sex_specific` parameter is set to **false**.

12.3.2.6 The `male_female` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `male_female` sub-block.

`parameter [, attribute]`	**Explanation**
`theta`	Specifies sex-specific recombination fractions for which a lod score is to be calculated. This parameter may be repeated as often as desired. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, male`	Specifies the male recombination fraction. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 1
`, female`	Specifies the female recombination fraction. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 1

Notes:

1. Required if the `theta` parameter is specified.

12.3.2.7 The `average` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `average` sub-block.

`parameter [, attribute]`	**Explanation**
`theta`	Specifies a sex-averaged recombination fraction for which a lod score is to be calculated. This parameter may be repeated as often as desired. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 1

Notes:

1. Required if the `average` parameter is specified.

Example 1

Do Morton's test for linkage homogeneity between model T1 (produced by SEGREG) and each marker in the pedigree file, estimating non-sex-specific recombination fractions. For these tests the group designated "1" consists of pedigrees 1-5 and group "2" consists of pedigrees 6-7.

```
lodlink {
   model, trait  = T1
   linkage_tests = false

   homog_tests {
      smiths_test  = false #explicitly set to the default value
      mortons_test = true, sex_specific = false {
         group = 1 {
            pedigree_id = 1
            pedigree_id = 2
            pedigree_id = 3
            pedigree_id = 4
            pedigree_id = 5
         }

         group = 2 {
            pedigree_id = 6
            pedigree_id = 7
            pedigree_id = 8
         }
      }
   }

   lods {
      option = none
   }
}
```

Example 2

Test for linkage between marker "Mfd154" and each of the other markers in the pedigree file estimating sex-specific recombination fractions assuming linkage homogeneity. Also calculate lod scores for the following pairs of recombination fractions: male .4, female 0; male .4, female .1; male .3, female .2.

Use the title "linkage test" in the output files. Name the summary and detail output files "example2.sum" and "example2.det", respectively.

```
lodlink, out = "example2" {
   title = "linkage test"
   model, marker = Mfd154
   linkage_tests = true, sex_specific = true, homog = true

   homog_tests {
      smiths_test  = false #explicitly set to the default value
      mortons_test = false #explicitly set to the default value
   }

   lods {
      option       = specified
      sex_specific = true

      male_female {
         theta, male = .4, female = 0
         theta, male = .4, female = .1
         theta, male = .3, female = .2
      }
   }
}
```

12.4 Program Execution

LODLINK is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running LODLINK from the command prompt with no arguments, or the wrong number of arguments, will result in the program displaying its usage statement, which lists the input files names required on the command line.

```
S.A.G.E. v4.4 -- LODLINK
COPYRIGHT (C) 2003 CASE WESTERN RESERVE UNIVERSITY
usage: ./lodlink <parameters> <pedigree> <locus> [trait/type]
Command line parameters:
  parameters   - Parameter File
  pedigree     - Pedigree Data File
  locus        - Marker Locus Description File
  trait/type   - Trait Locus Description File or SEGREG Generated Type File (optional)
```

A typical run of LODLINK may look like the following:

```
>lodlink par ped mld
S.A.G.E. v5.x -- LODLINK
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY
Reading Parameter File.....................done.
Reading Marker Locus Description File.....done.
Reading Pedigree File.....................
from ped.....................done.
Sorting pedigrees.........................done.
Generating statistics.....................
Parsing LODLINK analyses ...
Parsing new analysis block ...
Parsing of LODLINK Analysis 1 complete. Analysis valid.
-------------------------------------------------------------------
Performing lod score calculations ...
Performing lod ratio test for linkage ...
Parsing new analysis block ...
Parsing of LODLINK Analysis 2 complete. Analysis valid.
-------------------------------------------------------------------
Performing test for linkage under Smith's model ...
Performing Smith's homogeneity test ...
done.
Analysis complete!
```

12.5 Program Output

LODLINK produces four types of output files that contain results and diagnostic information:

File Name	File Type	Description
lodlink.inf	Diagnostic information output file	Contains informational diagnostic messages, warnings and program errors. No calculation results are stored in this file.
genome.inf	Genome information output file	Contains marker allele and genotype frequencies.
lodlink_analysis1.sum	Summary output	Contains lod scores and results of linkage and linkage homogeneity tests. Results in this file are based on calculations done on the pedigree data file as a whole.
lodlink_analysis1.det	Detailed output	Contains lod scores by family, the posterior probability that each family belongs to the linked pedigree, recombination fraction estimates by group for Morton's test, and individual trait genotype probabilities, as appropriate

12.5.1 Information Output File

The Information Output File contains a variety of useful information, including:

- Information on fields read from the Pedigree Data File. These tables, which provide information about what the program has read from the Pedigree Data File, are included with all programs in S.A.G.E. and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.

- Information, warning and error messages generated throughout the program. It is recommended that this file be checked for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "lodlink.inf" should be checked for errors and diagnostic information after each run of the program.

12.5.2 Genome Information Output File

Contains marker allele and genotype frequencies.

12.5.3 Summary Output File

Contains results pertaining to whole data set. See section 12.2 for details regarding interpretation of these results.

12.5.4 Detailed Output File

Contains results on a per individual, per family or per group basis. See section 12.2 for details regarding interpretation of these results.

12.6 Example Output Files

12.6.1 Summary Output File

```
================================
LODLINK Analysis 1 SUMMARY FILE
================================
Options Selected
================
Main locus type                          trait
Main locus name                          T1
Lod scores                               yes
Linkage tests                            yes
  Sex-specific recombination fractions   no
  Assume homogeneity                     yes
Smith's test for homogeneity             no
  Sex-specific recombination fractions   no
Morton's test for homogeneity            no
  Sex-specific recombination fractions   no
Genotype probabilities                   no
  Sex-specific recombination fractions   no
Recombination Fractions Selected
================================
Sex averaged      0.0000      0.0100      0.0500      0.1000      0.2000      0.3000

Results
=======
                  Lod Scores
                  Non-Sex-Specific Recombination Fractions

Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                -INFINITY       -73.3       -24.2       -7.23        3.15        4.31
M2                -INFINITY         -51        6.76        23.5        27.5          20
                  MLE                       Lod Score and Linkage Test
                                             Using Recom. Fract. in [0, .5]

Locus             Recom       Recom       Lod         Chi         P-Value     P-Value
                   Fract in    Fract in    Score       Square                  Upper
                   [0, .5]     [0, 1]                  Stat                    Bound
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                   0.2737         ---        4.42        20.4   3.196e-06   3.779e-05
M2                   0.1669         ---        28.2         130   < 1.0e-20   < 1.0e-20
```

12.6.2 Detailed Output File

```
==============================
LODLINK Analysis 1 DETAIL FILE
==============================
Results
=======
                  Lod Scores By Family
                  Non-Sex-Specific Recombination Fractions

                  Family in Pedigree 1 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                       0           0           0           0           0           0
M2                       0           0           0           0           0           0
                  Family in Pedigree 2 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                   0.301       0.292       0.258       0.215       0.134      0.0645
M2                       0           0           0           0           0           0
                  Family in Pedigree 3 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                -INFINITY       -1.11      -0.442      -0.188      0.0103      0.0704
M2                -INFINITY       -1.11      -0.442      -0.188      0.0103      0.0704
                  Family in Pedigree 4 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                       0           0           0           0           0           0
M2                     1.2        1.19        1.12        1.02       0.816       0.585
                  Family in Pedigree 5 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                -INFINITY      -0.809      -0.164      0.0668       0.214       0.217
M2                -INFINITY      -0.809      -0.164      0.0668       0.214       0.217
                  Family in Pedigree 6 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                       0           0           0           0           0           0
M2                     1.2        1.16       0.986        0.77       0.369      0.0984
.
.
.
                  Family in Pedigree 195 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                   0.602       0.593       0.558       0.511       0.408       0.292
M2                -INFINITY        -1.4      -0.721      -0.444      -0.194     -0.0757
                  Family in Pedigree 196 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                -INFINITY        -1.4      -0.721      -0.444      -0.194     -0.0757
M2                -INFINITY       -2.52        -1.2      -0.708      -0.303      -0.124
                  Family in Pedigree 197 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                   0.949        0.93       0.851        0.75       0.534       0.306
M2                  -0.176      -0.168      -0.137      -0.104     -0.0555     -0.0238
                  Family in Pedigree 198 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                       0           0           0           0           0           0
M2                -INFINITY        -2.8       -1.44      -0.887      -0.388      -0.151
                  Family in Pedigree 199 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                   0.347       0.336       0.295       0.245       0.151      0.0721
M2                -INFINITY       -1.33      -0.662      -0.396      -0.167     -0.0634
                  Family in Pedigree 200 Containing Member 1
Locus             0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
---------------   ---------   ---------   ---------   ---------   ---------   ---------
M1                   0.602       0.589       0.535       0.465       0.318        0.17
M2                   0.301       0.288       0.238        0.18      0.0849      0.0269
                  Lod Score Linkage Test
                  Variance-Covariance Matrices
                  Parameter Order (Avg Recomb)
Locus             Estimates in [0, .5]          Estimates in [0, 1]
---------------   --------------------          -------------------
M1                   0.0012                         ---
M2                  0.00029                         ---
```

Chapter 13

MLOD

MLOD Performs multi-point model-based LOD-score linkage analysis on small constituent pedigrees. Analysis is optimized for examining one-locus trait models and will, in future versions, allow for meiosis specific recombination fractions.

13.1 Limitations

MLOD calculates the likelihood of each possible inheritance pattern (i.e., ancestral origin of each allele) at each marker location for each constituent pedigree, using all marker data and assuming no crossover interference. It is restricted to small pedigrees due to the exponential nature of the algorithm related to the number of individuals in the pedigree. Only discrete traits may be analyzed, but there is no limit on the number of discrete categories allowed (this effectively allows the analysis of continuous traits). The time and space complexity of the algorithm is largely characterized by the exponent $2n - f$, the number of bits in an inheritance vector, where n is the number of non-founders and f is the number of founders in a constituent pedigree. During parameter specification the maximum value of $2n - f$ may be set, so that any constituent pedigree that has a value larger than this maximum will be skipped.

13.2 Theory

Given trait data, T, marker data, M, for a chromosomal region, and a point of interest in that region, p, MLOD computes a multi-point LOD score, defined as:

$$Z(p) = \log_{10}\left(\frac{P(M \mid T\,at\,p)}{P(M)P(T)}\right),$$

where P(T) can be a probability mass or density function.

Given a chromosomal region, a trait, and several pedigrees, MLOD calculates multi-point LOD scores for each point of interest along the chromosome by first generating exact multi-point likelihoods at each marker using a modified Lander-Green approach (Idury and Elston, 1996), and then computing the likelihood for the trait of each inheritance pattern (which is proportional to the probability of the trait for each inheritance pattern). These likelihoods are combined to generate the final LOD score at each location specified by the user.

13.2.1 The Exact Multi-point Algorithm

The general algorithm used by MLOD to generate multi-point likelihoods and other related statistics is called the exact multi-point algorithm. This algorithm takes a chromosomal region and generates likelihoods of all the possible inheritance patterns at each marker location in the region. These likelihoods are then combined at each marker location to generate multi-point LOD scores.

Given a pedigree with f founders and n non-founders and a pattern of segregation at a particular locus for this pedigree, we may represent this segregation as a vector of binary (0 or 1) digits of length $2n$ where each element represents one of the $2n$ meioses in the pedigree. The value of each binary element is determined by that meiosis having either a grandpaternal or grandmaternal allele from the parent. This "inheritance vector" is the basis for the Lander-Green multi-point algorithm (Lander and Green, 1987).

Because each meiosis is a separate event at a given locus, there are 2^{2n} possible patterns of locus segregation in the pedigree for each marker. However, because founder phase is unknown, it is impossible to determine the true state of the meioses from the founders. This means that, for the founder meioses, we do not know the binary values to be used in the inheritance vectors for a given inheritance pattern. Each inheritance pattern can therefore be represented by 2^f different inheritance vectors that represent the same inheritance pattern and share the same likelihood. These "equivalence classes" of inheritance vectors reduce the number of vectors that we must consider to 2^{2n-f}.

For a given set M of i markers $m_1 \ldots m_i$ (including a trait-marker, i.e. a trait considered in the same manner as any other marker but with more general penetrance functions), we calculate the joint probability of each inheritance vector and the pedigree data at each marker. The set of 2^{2n-f} joint probabilities at a particular marker is called the *likelihood vector* for that marker. The sum of these 2^{2n-f} joint probabilities is proportional to the likelihood for the pedigree data.

13.2.2 Combining Likelihood Vector Elements to Obtain a Multi-point Likelihood

Given two likelihood vectors, v_1 and v_2 at markers m_1 and m_2, and a recombination fraction θ_1 between them, we wish to calculate the joint likelihood.

To do this, we form a transition matrix T_1. This is a $2^{2n-f} \times 2^{2n-f}$ matrix with elements $t_{\alpha\beta} = \theta_1^q (1-\theta_1)^{2n-f-q}$ where α, β are inheritance vectors of the two markers and q is the Hamming Distance between them (the number of elements of α, β that differ). Then,

$$L(v_1, v_2) = v_1' T_1 v_2.$$

To add a third likelihood vector v_3 at marker m_3, with recombination fraction θ_2 between m_2 and m_3, we form a transition matrix T_2 analogous to T_1. Then

$L(v_1, v_2, v_3) = v_1' T_1 V_2 T_2 v_3$, where V_2 is a $2^{2n-f} \times 2^{2n-f}$ diagonal matrix containing the elements of v_2.

In general,

$$L(v_1, v_2, \ldots v_{i-1}, v_i) = v_1' T_1 V_2 T_2 \ldots V_{i-1} T_{i-1} v_i.$$

Idury and Elston (1996) suggested methods of calculating these likelihoods that are efficient, given the underlying structure of the transition matrices. S.A.G.E. extends these methods to include additional optimizations that use the genetic information at the markers to reduce the time complexity

of these algorithms. Even so, the algorithm takes time and space that increases exponentially with the size of the pedigree. It is for this reason that these algorithms are restricted to small-to-medium sized pedigrees.

13.2.3 Using Genetic Information to Improve Algorithm Performance

There are 2^{2n-f} inheritance vectors that we must consider at each marker. However, when most individuals are typed, the joint probability of the data and many of these inheritance patterns will be zero, because the inheritance pattern indicated by the vector is not consistent with the observed phenotypes at the marker in question.

A ***fixed point*** is any meiosis where the transmission is known with certainty. Given a fixed point in our likelihood vector, all inheritance vectors that do not match the transmission of the fixed point have a joint probability of 0. This information is used to speed up the computation. For each fixed point, we can reduce the time required for calculation by a factor of 2. These reductions are cumulative, so that for n fixed points, the time is reduced by a factor of 2^n.

13.2.4 Calculating Multi-point Likelihood Vectors

It is often necessary to calculate the multi-point likelihood vector at a specific location p along a chromosome. Assume we have a chromosome containing markers $m_1, \ldots, m_i$ with distances $d_1, \ldots, d_{i-1}$ between them. We have two adjacent markers, m_j and m_{j+1} between which is a point p for which we wish to calculate a multi-point likelihood vector *v*, with p some known distances d_{j1} and d_{j2} (where $d_{j1} + d_{j2} = d_j$) from m_j and m_{j+1}, respectively. Distances are expressed as recombination fractions and may be computed from genetic distance using either the Kosambi or Haldane map function.

First, we calculate $v_1, \ldots, v_i$, the single-point likelihood vectors for each marker. Then we calculate the following:

$$P_{j1} = v_1' T_1 V_2 T_2 \ldots V_j \, and \, P_{j2} = v_i' T_{i-1} V_{i-1} \ldots V_{j+1}.$$

P_{j1} is the multi-point information contributed to point p by all markers before point p, while P_{j2} is the multi-point information contributed by all markers after p. Each is a $1 \times 2^{2n-f}$ vector representing the combined multi-point information contributing to *v*. Calculating *v* is now trivial:

$$v = P_{j1} T_{j1} T_{j2} P_{j2}'$$

where $P_{j2}'$is a diagonal matrix consisting of elements of P_{j2}.

13.2.5 Computing LOD Scores

For a given point p on a chromosome, we calculate the multi-point LOD score given that a trait locus T (the trait-marker), is at that location by first calculating P($M|T$ at p), the multi-point likelihood for the chromosome given that T is present at that location and follows the model specified. We then calculate, P(M), the multi-point likelihood for the chromosomal region without T, and P(T), the probability of the trait given the underlying model. Then the LOD score for T being at point p is

$$Z(p) = \log_{10} \left(\frac{P(M \mid T \, at \, p)}{P(M) P(T)} \right).$$

At each location *p* we generate a LOD score for each pedigree. The combined LOD score at *p* is the sum of each constituent pedigree's individual LOD score at *p*.

13.2.6 Computing Information Content

Information content at a location is determined based on the probabilities of each inheritance pattern within the likelihood vector at that location. If we have n possible inheritance patterns, $i_1 \ldots i_n$, each with b bits and probability p_i such that

$$\sum_i p_i = 1,$$

then, the Information I is defined by [Kruglyak and Lander, 1995b]

$$I = 1 + \frac{\sum_i p_i \frac{\log(p_i)}{\log(2)}}{b}.$$

13.3 Program Input

MLOD requires the following input files in order to run:

File Type	Description
Parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual, including fields for identifiers, sex, parents, trait, and marker data.
Marker locus Description File	Lists the alleles, allele frequencies and phenotype to genotype mapping for each marker locus.
Trait locus description file	Lists the genetic model for each trait being analyzed.
Genome description file	Contains a description of the linked marker regions, including distances between markers.

13.3.1 The `mlod` Parameter

The following table shows the main MLOD syntax:

`parameter [, attribute]`	**Explanation**	
`mlod`	Starts a MLOD analysis block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the “root” name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	An alphanumeric constant representing a valid filename
	Default Value	”mlod”
	Required	No
	Applicable Notes	1

Notes

1. An analysis output file is generated for each analysis performed. The name of this file may be provided in the out attribute of the `mlod` parameter. If no filename is provided, the filename defaults to the name of the region with the extension ".lod" appended to it.

13.3.2 The `mlod` Parameter Block

The following lists all parameters that may occur in a MLOD block.

parameter [, attribute]	**Explanation**
`scan_type`	Specifies option to computes LOD scores at the observed markers or at the markers and intervals between them. Value Range: marker, interval Default Value: marker Required: No Applicable Notes: 3
`title`	Specifies title of the run. Value Range: Character string Default Value: None Required: No Applicable Notes: None
`region`	Specifies the name of the region to be analyzed. Must be a name listed in the genome description file. Value Range: Character string Default Value: None Required: No Applicable Notes: 5
`trait_marker`	Character string representing the name of a trait-marker to be analyzed. MLOD requires at least one trait-marker to be specified, but the user may list as many as desired. Value Range: Character string Default Value: None Required: Yes Applicable Notes: None
`max_size`	Maximum size (2n - f) of pedigree to analyze. Value Range: {0, 1, 2, ...} Default Value: 18 Required: Yes Applicable Notes: None
`distance`	Sets the interval used to compute LOD scores between observed markers in centiMorgans. Value Range: $(0, \infty)$ Default Value: 2.0 Required: No Applicable Notes: 1

`pedigree_lod_out`	Print verbose results of LOD score results for each pedigree.	
	Value Range	true false
	Default Value	false
	Required	No
	Applicable Notes	2
`output_pedigrees`	Controls the amount of output generated on a per pedigree basis.	
	Value Range	{**none**, **markers**, **all**}
	Default Value	**none**
	Required	No
	Applicable Notes	3
`sample_detail`	Controls the amount of detail provided about the useable pedigree data sample.	
	Value Range	{**none**, **removed**, **all**}
	Default Value	removed
	Required	No
	Applicable Notes	4

Notes

1. The `scan_type` parameter defines the locations where LOD scores are to be computed. If the value of `scan_type` is set to **markers**, then LOD scores are computed only at observed marker loci. If set to **intervals**, then LOD scores are computed both at the marker locations and at intervals between markers defined by the `distance` parameter.

2. The value of the `pedigree_lod_out` parameter affects the verbosity of the results in the LOD analysis output file. If `pedigree_lod_out` is set to **true**, LOD scores will be printed at marker loci, and each of the locations computed in between markers for each individual pedigree. Otherwise LOD scores are printed only at marker loci. This option does not affect how the overall LOD score results are output in the LOD analysis summary output file.

3. If `output_pedigrees` is set equal to **markers**, pedigree tables are printed, but only for the markers. If set equal to **all**, all points in the region are produced.

4. If `sample_detail` is set equal to **removed**, the table only includes those individuals removed from analysis (with reasons for removal), if set equal to **all**, all individuals are included in the table with reason for removal or being kept.

5. This causes the region to be analyzed using the current parameter settings and the corresponding output to be generated. If the value of the `region` parameter is not the name of a valid region, then the analysis is skipped. If no `region` parameters are specified in the parameter file, then multi-point LOD scores will be computed for each region in the genome description file. If no genome description file is given then single-point LOD scores will be generated at each marker in the marker locus description file.

13.4 Program Execution

MLOD is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running MLOD from the command prompt with no arguments, or with the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>mlod
S.A.G.E. v5.x -- MLOD
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: mlod <params> <pedigree> <traits> <locus>
<map>
Command line parameters:
params - Parameters File
pedigree - pedigree data file
traits - trait-marker description file
locus - locus description file
map - Genome Map File
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of MLOD may look like the following:

```
>mlod data.par data.ped data.trt data.loc data.map
S.A.G.E. v5.x -- MLOD
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Loading Map file...
Validating Analysis...
ANALYSES
================================================================================
LOD Score Analysis : LOD score analysis on region 'chr5'. Maximum connected
pedigree size is set to 18.
Processing Analyses....
================================================================================
LOD Score Analysis : LOD score analysis on region 'chr5'. Maximum connected
pedigree size is set to 18.
LOD Score Analysis: Pedigree 1
Generating Marker Likelihoods ...................Done.
Generating Multipoint Combined Info...................Done.
Generating Marker Likelihoods ...................Done.
LOD Score Analysis: Pedigree 2
Generating Marker Likelihoods ...................Done.
Generating Multipoint Combined Info...................Done.
Generating Marker Likelihoods ...................Done.
LOD Score Analysis: Pedigree 3
Generating Marker Likelihoods ...................Done.
Generating Multipoint Combined Info...................Done.
Generating Marker Likelihoods ...................Done.
.
.
.
```

13.5 Program Output

MLOD produces several output files that contain results and diagnostic information:

Filename	File Type	Description
mlod.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
genome.inf	Genome Information File	Contains diagnostic information on the genetic map data and the marker loci that were provided for analysis. No analysis results are stored in this file.
region.lod	LOD analysis output file(s)	There is one LOD analysis output file for each analysis performed by MLOD. This file contains a table for each pedigree analyzed, listing LOD scores and information content for each trait analyzed at each marker location.
summary.out	LOD Analysis summary output file	Contains a table for each analysis performed by MLOD. This table sums LOD scores and pools information content over all pedigrees for each point considered in the analysis.

13.5.1 Information Output File

The MLOD Information file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read in, are included with all programs in S.A.G.E. and are very useful for debugging many common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be carefully checked to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not run as expected. The file "mlod.inf" should be checked for errors and diagnostic information after each run of the program.

13.5.2 Genome Output File

This file includes warnings and errors produced while parsing the locus description files, as well as a table for each marker listing allele and genotype population frequencies, assuming Hardy-

Weinberg equilibrium. If allele frequencies do not sum to 1.0, they are standardized to 1.0, so these frequencies may not be as described in the locus description files. Note that a table is produced only for markers and not for trait-markers (traits with models.)

13.5.3 LOD Analysis Output File

A separate LOD analysis output file is created for each analysis performed by MLOD. This file contains a table for each pedigree analyzed, listing LOD scores and information content at each marker for each trait analyzed. Points between markers are also be listed if the `pedigree_lod_out` parameter has been set to **true**.

13.5.4 LOD Summary Output File

The LOD summary output file contains a table for each analysis performed by MLOD. These tables summarize LOD scores and information content for each point considered in the analysis by summing LOD scores from all pedigrees in the data set into a single LOD statistic. Information content is similarly summarized.

13.6 Example Output Files

```
LOD Table File
================================================================================
Analysis 'Analysis 1' on marker region 'chr5'
================================================================================
Pedigree 1   1 LOD scores
================================================================================
                                Dominant                Recessive
          Marker       LOD score   Information   LOD score   Information
         --------     -----------  -----------  -----------  -----------
   0.0  D5G1             -0.00001   0.94751443     -0.00001   0.94746010
   0.7  D5G2             -0.00000   1.00000000     -0.00001   1.00000000
   3.0  D5G3             -0.00000   0.97313461     -0.00001   0.97312913
   5.4  D5G4             -0.00000   0.98934838     -0.00001   0.98933651
   6.5  D5G5             -0.00000   1.00000000     -0.00001   1.00000000
   9.3  D5G6             -0.00000   0.96847998     -0.00001   0.96847998
  11.6  D5G7             -0.00056   0.96911857     -0.00001   0.96915157
  14.8  D5G8             -0.00056   1.00000000     -0.00001   1.00000000

# Summary LOD score Output file
Analysis: Analysis 1 (chr5
=================================================
 Pos      Trait       Marker      LOD score     Information   # Ped.
------  ----------  ----------  -------------  -------------  -------
   0.0  Dominant    D5G1         -5.519175122  0.96972173829      239
   0.7  Dominant    D5G2         -5.555971669  0.98266560809      239
   3.0  Dominant    D5G3         -5.979345816  0.98893249560      239
   5.4  Dominant    D5G4         -5.724044828  0.98947251776      239
   6.5  Dominant    D5G5         -5.771942142  0.98631478376      239
   9.3  Dominant    D5G6         -6.028611158  0.97542619041      239
  11.6  Dominant    D5G7         -6.498739837  0.99294067379      239
  14.8  Dominant    D5G8         -6.815240677  0.96763531437      239
  17.2  Dominant    D5G9         -6.298380200  0.98134489280      239
  19.6  Dominant    D5G10        -5.971049479  0.98499045128      239
  22.6  Dominant    D5G11        -5.115293921  0.97504538569      239
  23.5  Dominant    D5G12        -5.219382005  0.98451863326      239
  26.1  Dominant    D5G13        -6.053329330  0.99199116082      239
  27.9  Dominant    D5G14        -5.906801214  0.99470950295      239
  30.3  Dominant    D5G15        -6.435072155  0.99356584224      239
```

Chapter 14

ASSOC

ASSOC analyzes the association between a continuous trait and one or more covariates from pedigree data in the presence of familial correlations, simultaneously estimating familial variance components (and hence familial correlations and heritability). Covariates may include marker phenotypes that have been transformed into quantitative covariates by using a function block in the parameter file. Given data on one or more independent pedigrees sampled at random, this program estimates (by maximum likelihood assuming a generalization of multivariate normality) the parameters of the model with and without inclusion of a specified set of covariates. It provides the corresponding values of the ln(likelihood), as well as twice the difference between these values, so that the significance of the set of covariates can be determined. It also calculates numerically the standard errors of the estimates of all individual parameters in the model and performs an appropriate Wald test on each.

14.1 Limitations

ASSOC does not support pedigrees with loops, mating clusters of more than three total individuals (one individual with two spouses), or mating chains. Any constituent pedigree data that contains these structures will be excluded from the analysis. Also, ASSOC does not infer marker genotypes from their relatives' genotypes.

Further, if the sample size is small relative to the number of parameters being estimated, the likelihood may have multiple maxima. There is no guarantee that in such a situation the maximum found and reported by the program is also the global maximum. Also, situations can occur in which it is not numerically possible to calculate the variance-covariance matrix of the estimates.

14.2 Theory

14.2.1 Description of the Model

To incorporate familial correlations and arbitrary covariates into a likelihood, we assume the correlation structure described in Elston, George and Severtson (1992) and the regression model described in George and Elston (1987). For individual *i*, let:

Y_i	=	a continuous trait
x_i	=	a vector of covariates
G_i	=	a polygenic effect
F_i	=	family effect
$F_i^{'}$	=	family effect
M_i	=	a marital effect
S_i	=	a sibship effect
E_i	=	a random environmental effect

Then for a continuous trait the model is of the form

$$h(Y_i) = h\left(\beta^T x_i\right) + G_i + F_i + F_i^{'} + M_i + S_i + E_i,$$

where h is a transformation[1], and the polygenic effect (G_i) and all the environmental effects $(F_i,\ M_i,\ S_i,\ E_i)$ are assumed to be normally distributed random effects with zero means. All covariates are centered prior to inclusion in the likelihood (see below). Because the transformation *h* is applied to both sides of the equation, the estimates of the parameter values in β are on the original scale on which Y_i is measured. F_i is a random effect that a person shares with his/her spouse and children; F_i' is an effect that person shares with his/her parents, full sibs, half sibs and half sibs' parents; M_i is an effect that spouses share with each other; S_i is an effect that full sibs share with each other; and E_i is a person-specific random effect. These random effects are assumed to have variances $\sigma_G^2,\ \sigma_F^2 = \sigma_{F'}^2,\ \sigma_M^2,\ \sigma_S^2\, and\, \sigma_E^2$, respectively, and these variances are on the transformed scale.

Thus,

$$V\left[h\left(Y\right)\right] = \sigma_G^2 + 2\sigma_F^2 + \sigma_M^2 + \sigma_S^2 + \sigma_E^2, \tag{14.1}$$

where any of the variances other than σ_E^2 can be set to zero. However, if only 2-generation families are present in the sample,

$$V\left[h\left(Y\right)\right] = \sigma_G^2 + \sigma_F^2 + \sigma_M^2 + \sigma_S^2 + \sigma_E^2.$$

For σ_F^2 to be estimable, it is often necessary to have large pedigrees or large numbers of pedigrees, or both, and therefore σ_F^2 is set equal to zero by default. Variance components divided by the total variance can be interpreted as intraclass correlations (interclass in case of the marital correlation); it is not possible to estimate any variances to be less than zero.

14.2.2 Likelihood for a Randomly Sampled Pedigree

The likelihood formulation is based on the assumption of normality of the residuals and on the assumed correlational structure of the y_j. An algorithm for computing this likelihood is described in Elston et al. (1992).

It should be noted that singletons (unrelated individuals) may be included in the data. Although ASSOC counts and treats them separately for convenience, they are in fact simply one-person pedigrees with parent information missing and, as such, require no special treatment in the model.

[1]The dependent variable y may be transformed by one of two transformations: Box-Cox or George-Elston. See section 5.3.2.2 for details on the transformation theory implemented in this program.

14.2.3 Estimation of Parameters

Estimation is performed by maximizing the likelihood numerically. For computational reasons, however, the likelihood is not considered directly. Instead of the likelihood L based on the above model description, we maximize its natural logarithm ln(L). If several independent pedigrees are analyzed jointly, the logarithms of the likelihoods are summed over all pedigrees.

The program itself determines initial estimates for the maximizing process. The user, however, may override the initial estimate of any of the variance components or covariate coefficients.

14.2.4 Tests

ASSOC can be executed in one of two ways. If the user does not specify any test covariates, the likelihood maximization will be performed *once*, and there will be no joint-test output.

If, however, the user does specify test covariates, the maximum ln(likelihood) of the model is determined under two hypotheses: H_1 assumes the general model, including all test covariates specified by the user; H_0 , the null hypothesis, excludes from the regression model the test covariates specified by the user. If L_1 and L_0 are the maximum likelihoods under H_1 and H_0 respectively, then the likelihood ratio statistic is $2[ln(L_1) - ln(L_0)]$. Under the assumption of normality of the transformed variable and the null hypothesis that the test covariates have no effect, this statistic is asymptotically distributed as chi-square with the number of degrees of freedom being equal to the number of test covariates. In addition, p-values are calculated for each individual parameter in the model using its standard error obtained by double differentiation of the ln likelihood. These p-values are two-sided for the covariate coefficients β, λ_1, and λ_2; they are one-sided for all variances. In each case the test is for the null hypothesis that the parameter is 0, except for λ_1, where the null hypothesis is $\lambda_1 = 1$.

14.3 Program Input

File Type	Description
Parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data File	Contains delimited records for each individual including fields for identifiers, parents, trait and covariates.
Marker locus description file	Lists the alleles at each marker locus. This file will be used by ASSOC only if a marker, transformed to be quantitative, is used as a covariate[a].

[a] ASSOC does not use any information on allele frequencies or phenotype to genotype mapping that may be in the Marker Locus Description File.

14.3.1 Parameter File Syntax

The specific syntax for ASSOC parameters, attributes and values is described in the following sections.

14.3.1.1 The `assoc` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main ASSOC parameter.

`parameter [, attribute]`	**Explanation**	
`assoc assoc_analysis,`	Starts an ASSOC analysis block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Valid file name in the form of a quoted character string.
	Default Value	analysis_n, where n = 1, 2, ..., k for a given set of k specified ASSOC analyses.
	Required	No
	Applicable Notes	None

14.3.1.2 The `assoc` Block

The following syntax table specifies the permissible parameter and attribute settings for the `assoc` block.

`parameter [, attribute]`	**Explanation**	
`title`	Specifies the title of the run.	
	Value Range	Quoted character string.
	Default Value	analysis_n, where n = 1, 2, ..., k for a given set of k specified ASSOC analyses.
	Required	No
	Applicable Notes	1
`trait` `primary_trait`	Specifies a dependent variable as the trait in the regression model.	
	Value Range	Character string representing the name of a trait, phenotype or covariate from the pedigree data file.
	Default Value	None
	Required	Yes
	Applicable Notes	None
`covariate` `cov`	Specifies a variable in the regression model. It can be a trait name, a covariate name, or a phenotype name.	
	Value Range	The name of a trait, phenotype, covariate.
	Default Value	None
	Required	No
	Applicable Notes	2, 3
`, test`	Specifies a covariate to be included in the general model H_1, but excluded from the null hypothesis H_0.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	3
`, val`	Specifies the initial estimate for the covariate coefficient.	
	Value Range	(-∞, ∞)
	Default Value	None
	Required	No
	Applicable Notes	None
`, fixed`	Specifies that the coefficient for this covariate is fixed.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	4

`polygenic_effect pe`	Specifies the inclusion of a polygenic variance component in the model. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 5
`, val`	Specifies the initial estimate for this variance component. Value Range: $[0, \infty)$ Default Value: None Required: No Applicable Notes: 5
`, fixed`	Specifies that the effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
`family_effect fe`	Specifies the inclusion of a nuclear family variance component in the model. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 5
`, val`	Specifies the initial estimate for this variance component. Value Range: $[0, \infty)$ Default Value: None Required: No Applicable Notes: None
`, fixed`	Specifies that this effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
`marital_effect me`	Specifies the inclusion of a marital (i.e., spousal) variance component in the model. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 5
`, val`	Specifies the initial estimate for this variance component. Value Range: $[0, \infty)$ Default Value: None Required: No Applicable Notes: None

, fixed	Specifies that the effect is fixed. Value Range: true, false Default Value: false Required: No Applicable Notes: 4
sibship_effect se	Specifies the inclusion of a sibling variance component in the model. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 5
, val	Specifies the initial estimate for this variance component. Value Range: [0, ∞) Default Value: None Required: No Applicable Notes: None
, fixed	Specifies that the effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
transformation transform trans	Starts a transformation sub-block. Value Range: N/A Default Value: None Required: No Applicable Notes: 7
maxfun	Starts a **maxfun** sub-block to specify diagnostics options for the likelihood maximization process. Value Range: N/A Default Value: None Required: No Applicable Notes: 8
allow_averaging aa	This option allows the user to substitute covariates' respective means for missing covariate data. Value Range: {mean, none} Default Value: none Required: No Applicable Notes: 6

Notes

1. The `title` parameter also specifies the naming convention for ASSOC output files. However, the value of the `out` attribute (of the `assoc` parameter) will override the `title` parameter as the name of output files.

2. If a `sex_code` covariate is specified, the estimated effect will be for that of a female (i.e.

males are coded 0, females are coded 1). This requires that the `sex_code` has been specified as available to be used as a trait.

3. The trait analyzed can be a linear function of the primary trait (with coefficient 1) and other covariates whose coefficients are fixed or estimated. This linear function is called a composite trait. Without this sub-block a composite trait is not formed. All covariates are centered, the centering (average) value being included as part of the output. The covariates can be any covariate, phenotype or trait (other than the primary trait) listed in the pedigree data file. Note: This sub-block is not applicable to binary traits.

4. If the `fixed` attribute is set to **true**, the attribute `val` must be included. If set to **false** and the attribute `val` is included, this determines the initial value of the variable to be used in the maximization process. If set to **false** and the attribute `val` is not included, then the program supplies various initial values for the maximization process.

5. If `val` is set to 0 and `fixed` is set to **true**, the relevant effect (polygenic, family, marital, etc.) will be excluded from the model. This is equivalent to setting the effect to **false** (i.e., `pe = false`, `fe = false, or me = false`).

6. If the value **none** is specified, and any individual's value is missing for any particular covariate, then that individual will be treated as uninformative for the purpose of the analysis. If **mean** is specified, the individual's missing covariate value will be replaced with the sample mean of that covariate (calculated on the basis of all fully informative individuals).

7. By default, ASSOC will estimate λ_1 (and fix λ_2 at 0), using the George-Elston transformation. See section 5.3.2.2 for details on the transformation theory implemented in this program.

8. MAXFUN (***fun****ction* ***max****imization*) is the name of the S.A.G.E. library component designed to find the global maximum of a given function with respect to a vector of parameters. When a S.A.G.E. program reports a failure of convergence to a global maximum, the `maxfun` sub-block provides the ability to generate diagnostic information about the maximization process. The diagnostic information is generated to a text file with a ".max" extension on the filename.

14.3.1.3 The `transformation` sub-block

The following syntax table specifies the permissible parameter and attribute settings for the `transformation` sub-block.

parameter [, attribute]	**Explanation**
`option`	Specifies a particular transformation option. Value Range: {none, box_cox, george_elston} Default Value: george_elston Required: No Applicable Notes: 1
`lambda1`	Specifies the power parameter Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, val`	Value of the parameter Value Range: (-∞, ∞) Default Value: 1 Required: No Applicable Notes: None
`, fixed`	Specifies option to fix the value. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
`, lower_bound`	Specifies inclusive lower bound for power parameter. Value Range: (-∞, ∞) Default Value: -1 Required: No Applicable Notes: None
`, upper_bound`	Specifies inclusive upper bound for power parameter. Value Range: (-∞, ∞) Default Value: +∞ Required: No Applicable Notes: None
`lambda2`	Specifies the shift parameter Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, val`	Specifies the value of the parameter. Value Range: (-∞, ∞) Default Value: 0 Required: No Applicable Notes: None

`, fixed`	Option to fix this value.	
	Value Range	{true, false}
	Default Value	None
	Required	No
	Applicable Notes	None

Notes

1. An `option` value of **none** disables transformation calculations for the analysis, and an `option` value of either **george_elston** or **box_cox** means that both the λ_1 and λ_2 transformation parameters are to be estimated.

14.3.1.4 The `maxfun` sub-block

The following syntax table specifies the permissible parameter and attribute settings for the `maxfun` sub-block.

`parameter [, attribute]`	**Explanation**	
`level`	Specifies the quantity and type of diagnostic information desired from the MAXFUN component.	
	Value Range	no_debug_info basic per_run complete
	Default Value	None
	Required	No
	Applicable Notes	1

Notes

The following are all valid `assoc` statements:

```
assoc_analysis {
   trait = TRAIT1
   cov   = TRAIT2, test
}

assoc_analysis, out = my_test {
   trait = trait1
   cov   = a_covariate, test
}

assoc_analysis {
   title = “Analysis, Oct. 8, 2001“
   trait = TRAIT3
   cov   = COV1, test
}

assoc_analysis, out=Assoc_res {
   trait = TRAIT3
   cov   = COV1, test
   cov   = cov2, test
   cov   = Cov3
   cov   = TRAIT1
}

assoc_analysis, out=test_analysis {
   title         = ”Test Ignore”
   cov           = a_trait1
   cov           = a_trait2
   cov           = a_test_cov_1, test
   cov           = a_test_cov_2, test
   primary_trait = the_trait
}
```

14.3.2 Exclusion Criteria for Individuals and Pedigrees

Under some conditions ASSOC will exclude individuals and/or pedigrees from the analysis, and the user is advised to take note of program outputs that indicate the numbers of valid and invalid individuals being counted. The exclusion criteria are:

1. Any structurally invalid families will be excluded from the analysis (see 16.1).
2. Any individual whose primary trait is missing or who is missing at least one covariate value (if the `allow_averaging` option is disabled – the default behavior) will be retained to provide relationship information for the analysis, but all of the individual’s trait information will be treated as missing.

14.4 Program Execution

ASSOC is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running ASSOC from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>Assoc
S.A.G.E. v5.x -- ASSOC
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: Assoc <parameters> <pedigree> [locus]
Command line parameters:
parameters - Parameter File
pedigree - Pedigree Data File
locus - Locus Description File (optional)
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of ASSOC may look like the following:

```
>ASSOC data.par data.ped
S.A.G.E. v5.x -- ASSOC
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading Parameter File...................done.
Reading Pedigree File.....................
from data.ped..............done.
Sorting pedigrees.........................done.
Verifying Analyses:
Analysis 1.......................done.
Analysis 2.......................done.
```

14.5 Program Output

ASSOC produces three output files for each analysis, an Information Output File, a Summary Output File and a Detailed Output File:

Filename	Filetype	Description
`analysis.inf`	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
`analysis.sum`	ASSOC calculation summary output	Contains the final estimates and standard errors of the parameters used in the model
`analysis.det`	ASSOC calculation detailed output	This file contains the variance-covariance matrix of the estimates and the partial derivatives of the log likelihood with respect to the parameters.

14.5.1 Information Output File

The ASSOC Information file contains a variety of useful information, including:

- Information on fields read from the Pedigree Data File. These tables, which provide information about what the program has read from the Pedigree Data File, are included with all programs in S.A.G.E., and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that this file be checked before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not performed as expected. The file "assoc.inf" should be checked for errors and diagnostic information after each run of the program.
- A table indicating the structural validity of each constituent subpedigree in the pedigree data file used for the analysis. For each constituent subpedigree, the table lists whether or not it has loops, mating chains, or mating clusters larger than 3 total individuals.

14.5.2 Summary Output File

The Summary Output File contains the final estimates, standard error, and p-value of the parameters used in the model:

1. Transformation parameters
   - λ_1-Lambda 1
   - λ_2-Lambda 2
2. Variance components on the transformed scale.
   - σ^2_G-Polygenic variance
   - σ^2_E-Random variance
   - σ^2_F-Familial variance
   - σ^2_M-Marital variance
   - σ^2_S-Sibship variance
3. Coefficients
   - β_0-Intercept
   - β_j-Covariate coefficients, $j > 0$
4. Total variance: $V\left[h\left(Y_i\right)\right]$
5. "Heritability": $\sigma^2_G/V\left[h\left(Y_i\right)\right]$

6. Environmental intraclass correlations

   - Nuclear Family: $\sigma_F^2 / \left\{ V\left[h(Y_i)\right] - \sigma_G^2 \right\}$
   - Marital: $(\sigma_F^2 + \sigma_F^2) / \left\{ V\left[h(Y_i)\right] - \sigma_G^2 \right\}$
   - Sibship: $(\sigma_F^2 + \sigma_S^2) / \left\{ V\left[h(Y_i)\right] - \sigma_G^2 \right\}$

7. Residual familial correlations

   - Full Sib $(\sigma_F^2 + \sigma_S^2 + \frac{1}{2}\sigma_G^2) / \left\{ V\left[h(Y_i)\right] \right\}$
   - Half Sib $(\sigma_F^2 + \frac{1}{4}\sigma_G^2) / \left\{ V\left[h(Y_i)\right] \right\}$
   - Parent-offspring $(\sigma_F^2 + \frac{1}{2}\sigma_G^2) / \left\{ V\left[h(Y_i)\right] \right\}$
   - Step-parent-step-offspring $\sigma_F^2 / \left\{ V\left[h(Y_i)\right] \right\}$
   - Spouses: $(\sigma_F^2 + \sigma_M^2) / \left\{ V\left[h(Y_i)\right] \right\}$

In addition, several p-values are quoted based on the asymptotic distribution of the test statistics (likelihood ratio, Wald). p-values quoted for σ_G^2, σ_E^2, σ_F^2, σ_M^2 and σ_S^2 use a 1-sided test. All other p-values use 2-sided tests.

14.5.3 Detailed Output File

The detailed output includes:

1. The estimated variance-covariance matrix of all the estimated parameters.
2. The partial first derivative of the natural logarithm of the likelihood with respect to each of the parameters estimated.

14.6 Example Output Files

14.6.1 ASSOC Summary Output File

The following is an example of an ASSOC Summary Output File:

```
=================
      Results
=================
============================
      Sample description
============================
   Number of pedigrees in dataset               7
   Number of analyzable pedigrees               4

   Number of individuals in dataset           942
   Number of analyzable individuals           923
   Number of analyzable invalid individuals   424
   Number of analyzable valid individuals     499
==========================
      Model description
==========================
   Title                   pe_se_fe_me
   Primary Trait           sqrtdbh
   Test covariate:         apres        Mean =   0.512146  Std. dev. =   0.499852  Min. =   0.000000  Max. =   1.000000
-------------------------------------------------------------------------------------------------------------------------
Note: No transformation is applied.
=====================================================
      MAXIMIZATION RESULTS without test covariates
=====================================================
-------------------------------------------------------------------------
   Parameter                          Estimate   S.E.         P-value
-------------------------------------------------------------------------
   Variance components
                           Polygenic   2.251733   0.335338    < 1e-07
                              Family   0.000000  Ind. func. fixed @ bnd
                             Marital   0.000000  Ind. func. fixed @ bnd
                             Sibling   0.179865   0.098181     0.033478
                              Random   0.300521   0.191533     0.058320
   Correlations
                       Nuclear family   0.000000   0.000000     0.004495
                              Marital   0.000000   0.000000     0.004495
                              Sibling   0.374417   0.237980     0.115646
                           Full sibs   0.477919   0.047367    < 1e-07
                           Half sibs   0.206043   0.017349    < 1e-07
                     Parent-offspring   0.412085   0.034699    < 1e-07
                Step parent-offspring   0.000000   0.000000    < 1e-07
   Spouses or spouses of common spouse   0.000000   0.000000    < 1e-07
                           Intercept   4.941494   0.137505    < 1e-07
   Other parameters
                       Total variance   2.732120   0.225741    < 1e-07
                         Heritability   0.824171   0.069397    < 1e-07
-------------------------------------------------------------------------
Final ln likelihood: -881.373404
=================================================
      MAXIMIZATION RESULTS with test covariates
=================================================
-------------------------------------------------------------------------
   Parameter                          Estimate   S.E.         P-value
-------------------------------------------------------------------------
   Variance components
                           Polygenic   2.223050   0.335194    < 1e-07
                              Family   0.000000  Ind. func. fixed @ bnd
                             Marital   0.000000  Ind. func. fixed @ bnd
                             Sibling   0.185202   0.099337     0.031135
                              Random   0.311680   0.191454     0.051766
   Correlations
                       Nuclear family   0.000000   0.000000     0.003702
                              Marital   0.000000   0.000000     0.003702
                              Sibling   0.372727   0.229468     0.104309
                           Full sibs   0.476750   0.047508    < 1e-07
                           Half sibs   0.204330   0.017667    < 1e-07
                     Parent-offspring   0.408659   0.035334    < 1e-07
                Step parent-offspring   0.000000   0.000000    < 1e-07
   Spouses or spouses of common spouse   0.000000   0.000000    < 1e-07
                           Intercept   4.948390   0.137428    < 1e-07
   Test covariates
                               apres   0.094199   0.142815     0.509517
   Other parameters
                       Total variance   2.719933   0.224600    < 1e-07
                         Heritability   0.817318   0.070668    < 1e-07
-------------------------------------------------------------------------
Final ln likelihood: -881.155989
====================
      JOINT TEST
====================
   H0 ln likelihood without test covariates  -881.373404
   H1 ln likelihood with test covariates     -881.155989

   2 * |H0 - H1|                               0.434830
   Degrees of freedom                                 1
   P-value                                     0.509628
```

14.6.2 ASSOC Detailed Output File

The following is an example of an ASSOC Detailed Output File:

```
================
     Results
================
===========================
     Sample description
===========================
   Number of pedigrees in dataset              7
   Number of analyzable pedigrees              4

   Number of individuals in dataset          942
   Number of analyzable individuals          923
   Number of analyzable invalid individuals  424
   Number of analyzable valid individuals    499
==========================
     Model description
==========================
   Title                   pe_se_fe_me
   Primary Trait           sqrtdbh
   Test covariate:         apres        Mean =   0.512146  Std. dev. =   0.499852  Min. =   0.000000  Max. =   1.000000
-------------------------------------------------------------------------------------------------------------------------
Note: No transformation is applied.
=====================================================
     MAXIMIZATION RESULTS without test covariates
=====================================================
-----------------------------------------------------------------------------------
   Parameter                          Estimate   S.E.       P-value    Deriv
-----------------------------------------------------------------------------------
   Variance components
                           Polygenic   2.251733   0.335338  < 1e-07     0.0000003594
                              Family   0.000000        Ind. func. fixed @ bnd
                             Marital   0.000000        Ind. func. fixed @ bnd
                             Sibling   0.179865   0.098181   0.033478   0.0000002697
                              Random   0.300521   0.191533   0.058320   0.0000004046
   Correlations
                       Nuclear family   0.000000   0.000000   0.004495   0.0000000000
                              Marital   0.000000   0.000000   0.004495   0.0000000000
                              Sibling   0.374417   0.237980   0.115646   0.0000000000
                           Full sibs   0.477919   0.047367  < 1e-07     0.0000000000
                           Half sibs   0.206043   0.017349  < 1e-07     0.0000000000
                     Parent-offspring   0.412085   0.034699  < 1e-07     0.0000000000
                Step parent-offspring   0.000000   0.000000  < 1e-07     0.0000000000
   Spouses or spouses of common spouse   0.000000   0.000000  < 1e-07     0.0000000000
                           Intercept   4.941494   0.137505  < 1e-07     0.0000000000
   Other parameters
                        Total variance   2.732120   0.225741  < 1e-07     0.0000000000
                          Heritability   0.824171   0.069397  < 1e-07     0.0000000000
-----------------------------------------------------------------------------------
Final ln likelihood: -881.373404
=====================================================
     MAXIMIZATION RESULTS with test covariates
=====================================================
-----------------------------------------------------------------------------------
   Parameter                          Estimate   S.E.       P-value    Deriv
-----------------------------------------------------------------------------------
   Variance components
                           Polygenic   2.223050   0.335194  < 1e-07     0.0000001517
                              Family   0.000000        Ind. func. fixed @ bnd
                             Marital   0.000000        Ind. func. fixed @ bnd
                             Sibling   0.185202   0.099337   0.031135  -0.0000001349
                              Random   0.311680   0.191454   0.051766   0.0000001349
   Correlations
                       Nuclear family   0.000000   0.000000   0.003702   0.0000000000
                              Marital   0.000000   0.000000   0.003702   0.0000000000
                              Sibling   0.372727   0.229468   0.104309   0.0000000000
                           Full sibs   0.476750   0.047508  < 1e-07     0.0000000000
                           Half sibs   0.204330   0.017667  < 1e-07     0.0000000000
                     Parent-offspring   0.408659   0.035334  < 1e-07     0.0000000000
                Step parent-offspring   0.000000   0.000000  < 1e-07     0.0000000000
   Spouses or spouses of common spouse   0.000000   0.000000  < 1e-07     0.0000000000
                           Intercept   4.948390   0.137428  < 1e-07     0.0000000000
   Test covariates
                               apres   0.094199   0.142815   0.509517   0.0000000000
   Other parameters
                        Total variance   2.719933   0.224600  < 1e-07     0.0000000000
                          Heritability   0.817318   0.070668  < 1e-07     0.0000000000
-----------------------------------------------------------------------------------
Final ln likelihood: -881.155989
====================
     JOINT TEST
====================
   HO ln likelihood without test covariates  -881.373404
   H1 ln likelihood with test covariates     -881.155989

   2 * |H0 - H1|                               0.434830
   Degrees of freedom                                 1
   P-value                                     0.509628
=========================================================
     VARIANCE-COVARIANCE MATRIX without test covariates
=========================================================
-------------------------------------------------------------------------------------------------------------------------
                                     Polygenic  Family     Marital    Sibling    Random     Nuclear family  Marital    ...
-------------------------------------------------------------------------------------------------------------------------
```

```
   Polygenic                          0.112451   0.000000   0.000000   0.007029  -0.052069   0.000000        0.000000  ...
   Family                             0.000000   0.000000   0.000000   0.000000   0.000000   0.000000        0.000000  ...
   Marital                            0.000000   0.000000   0.000000   0.000000   0.000000   0.000000        0.000000  ...
   Sibling                            0.007029   0.000000   0.000000   0.009640  -0.008868  -0.000000       -0.000000  ...
   Random                            -0.052069   0.000000   0.000000  -0.008868   0.036685  -0.000000       -0.000000  ...
   Nuclear family                     0.000000   0.000000   0.000000  -0.000000  -0.000000   0.000000        0.000000  ...
   Marital                            0.000000   0.000000   0.000000  -0.000000  -0.000000   0.000000        0.000000  ...
   Sibling                            0.049737   0.000000   0.000000   0.019465  -0.040141   0.000000        0.000000  ...
   Full sibs                          0.011360   0.000000   0.000000   0.003450  -0.008533   0.000000        0.000000  ...
   Half sibs                          0.005206   0.000000   0.000000   0.000055  -0.002936   0.000000        0.000000  ...
   Parent-offspring                   0.010412   0.000000   0.000000   0.000110  -0.005871   0.000000        0.000000  ...
   Step parent-offspring             -0.000000   0.000000   0.000000  -0.000000   0.000000  -0.000000       -0.000000  ...
   Spouses or spouses of common spouse -0.000000  0.000000   0.000000  -0.000000   0.000000  -0.000000       -0.000000  ...
   Intercept                         -0.003622   0.000000   0.000000  -0.000332   0.002230  -0.000000       -0.000000  ...
   Total variance                     0.067411   0.000000   0.000000   0.007801  -0.024253   0.000000        0.000000  ...
   Heritability                       0.020824   0.000000   0.000000   0.000220  -0.011742   0.000000        0.000000  ...
=========================================================
      VARIANCE-COVARIANCE MATRIX with test covariates
=========================================================
-------------------------------------------------------------------------------------------------------------------
                                   Polygenic  Family     Marital    Sibling    Random     Nuclear family  Marital    ...
-------------------------------------------------------------------------------------------------------------------
   Polygenic                          0.112355   0.000000   0.000000   0.006410  -0.052018   0.000000        0.000000  ...
   Family                             0.000000   0.000000   0.000000   0.000000   0.000000   0.000000        0.000000  ...
   Marital                            0.000000   0.000000   0.000000   0.000000   0.000000   0.000000        0.000000  ...
   Sibling                            0.006410   0.000000   0.000000   0.009868  -0.008608  -0.000000       -0.000000  ...
   Random                            -0.052018   0.000000   0.000000  -0.008608   0.036654  -0.000000       -0.000000  ...
   Nuclear family                     0.000000   0.000000   0.000000  -0.000000  -0.000000   0.000000        0.000000  ...
   Marital                            0.000000   0.000000   0.000000  -0.000000  -0.000000   0.000000        0.000000  ...
   Sibling                            0.047113   0.000000   0.000000   0.018915  -0.038363   0.000000        0.000000  ...
   Full sibs                          0.011311   0.000000   0.000000   0.003462  -0.008525   0.000000        0.000000  ...
   Half sibs                          0.005313   0.000000   0.000000   0.000013  -0.002980   0.000000        0.000000  ...
   Parent-offspring                   0.010626   0.000000   0.000000   0.000026  -0.005961   0.000000        0.000000  ...
   Step parent-offspring             -0.000000   0.000000   0.000000  -0.000000   0.000000  -0.000000       -0.000000  ...
   Spouses or spouses of common spouse -0.000000  0.000000   0.000000  -0.000000   0.000000  -0.000000       -0.000000  ...
   Intercept                         -0.003931   0.000000   0.000000  -0.000244   0.002354  -0.000000       -0.000000  ...
   Total variance                     0.066747   0.000000   0.000000   0.007670  -0.023972   0.000000        0.000000  ...
   Heritability                       0.021251   0.000000   0.000000   0.000052  -0.011921   0.000000        0.000000  ...
   apres                             -0.005830   0.000000   0.000000   0.001160   0.002525  -0.000000       -0.000000  ...
```

Chapter 15

TDTEX

The transmission-disequilibrium test (TDT) introduced by Spielman et al. (1993) is a method for detecting linkage between a marker locus and a disease susceptibility locus when linkage disequilibrium or any other type of allelic association is present. The basic TDT test for binary traits has been generalized by Bickeböller and Clerget-Darpoux (1995), Rice et al. (1995), Curtis and Sham (1995), Olson et al. (1997). TDTEX is a computer program based on this work, and implements a very general system for detecting linkage in the presence of linkage disequilibrium between a marker locus and a disease locus with a binary phenotype.

15.1 Limitations

The TDTEX program makes the following assumptions:

1. Each marker has a known genotype-phenotype relation.
2. Only autosomal loci are considered.
3. Only binary phenotypes are considered.

This program is limited by the program execution time of the computer on which it runs. As the transmission table size and number of marker alleles increase, processing time becomes slower. The major computational limitation is the exact permutation algorithm. This becomes prohibitively slow for transmission tables with greater than around 300 observations, or with more than about 8 alleles. In such cases, the asymptotic or Monte Carlo test statistics are recommended instead.

15.2 Theory

TDTEX consists of four main components:

1. A scoring algorithm to identify which alleles or genotypes are transmitted to affected offspring.

2. Transmission tables (i.e., contingency tables) to summarize the number of transmitted vs. non-transmitted alleles or genotypes.

3. A pedigree sampler to identify and collect informative transmissions from pedigree data. The sampler collects transmission information in transmission tables, conditional on the types of relatives to be sampled (individual affected offspring or affected sibling pairs), the availability of marker data, and optionally on parental traits such as sex or affection status.

4. A suite of statistical tests to evaluate significance of the computed transmission tables under the null hypothesis of complete symmetry or marginal homogeneity. These tests include the standard asymptotic TDT tests which rely on large sample theory for validity. Exact tests that do not rely on asymptotic approximations are also provided at the expense of greater computational requirements.

15.2.1 Allele and Genotype Transmissions

Consider a sample of affected individuals and their parents typed for a genetic marker. The basis of the transmission-disequilibrium test is a case/control study, matching alleles found in an affected individual with internal family-based control alleles. The "case" alleles are those that were transmitted to an affected individual, and "control" alleles are the alleles not transmitted from the parents of the individual. By scoring these transmitted and non-transmitted alleles from pedigree data, it is possible to estimate the distribution of these transmissions. If the marker and trait loci are unlinked or are unassociated (in equilibrium), then the distribution of parental alleles transmitted to affected offspring will not differ in expectation from that of alleles that were not transmitted to the affected offspring. Otherwise, if *both linkage and disequilibrium* (or, more generally, linkage and allelic association, whatever the cause of that association) are present between marker and trait loci, then the distribution of alleles transmitted to the affected offspring will differ from that of the non-transmitted alleles. This scheme has the advantage of being robust to the presence of population stratification, a situation caused by admixture of populations with distinct marker allele and disease frequencies. For more details see Spielman et al. (1993).

We define an *allele transmission* from a single parent to a child to be an ordered pair of alleles, where the first allele is transmitted from the parent to the child and the second allele is the other parental allele, i.e., the one that is not transmitted to the child. In other words, an allele transmission is the ordered pair (A_1, A_2) where A_1 is the transmitted allele, and A_2 is the non-transmitted allele.

It is possible to combine the information from the allele transmissions from each of the two parents to a child. Since two allele transmissions involve two transmitted alleles (and two non-transmitted alleles), we can group the transmitted (and non-transmitted) alleles together to form a genotype. Thus a *genotype transmission* is defined as an ordered pair of genotypes, where the first genotype is formed by the two alleles transmitted from the parents to the child, i.e., the genotype of the child. Similarly, the second genotype includes the two alleles not-transmitted from the parents to the child. Consider a pair of allele transmissions from the two parents, (A_1, A_2) and (A_3, A_4). We denote a genotype transmission from these parents as $(A_1/A_3, A_2/A_4)$, where A_1/A_3 is the transmitted genotype and A_2/A_4 is the non-transmitted genotype (see Figure 15.1).

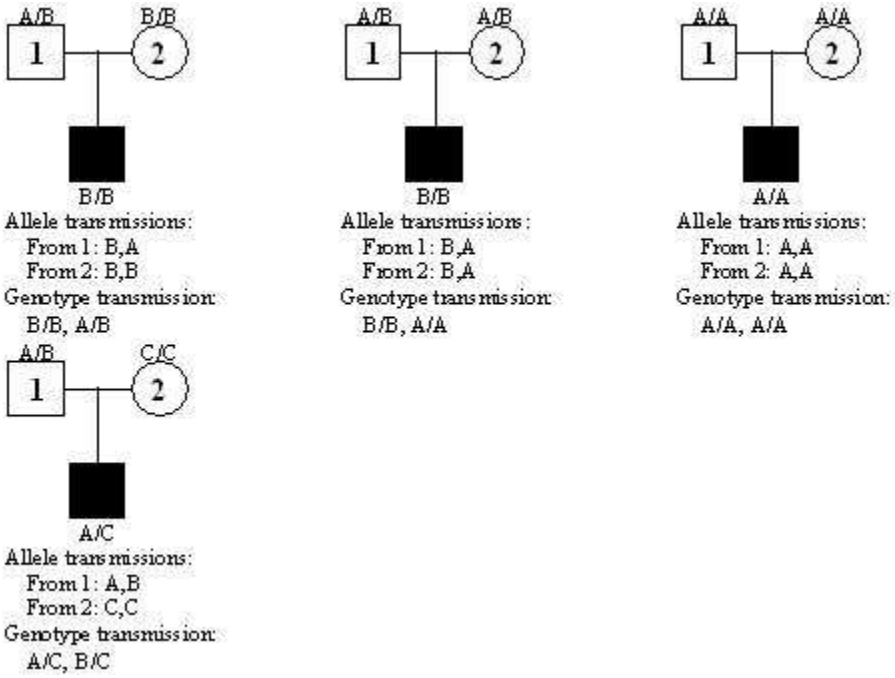


Figure 15.1: Allele and Genotype Transmission Examples

15.2.2 Scoring affected offspring

Scoring affected offspring requires computing the allele or genotype transmissions from the parents of an affected individual. However, not all such transmissions are informative and, in the presence of missing parental data, some transmissions cannot be used due to potential bias introduced by population stratification (Curtis and Sham, 1995). Table 1 represents the scoring function used by TDTEX for affected offspring. The basic distinct patterns of allele configurations for parents and children are shown, together with the resulting allele and genotype transmissions. All possible configurations can be obtained from these by relabeling alleles, permuting the two parents, or permuting the alleles within individuals.

Empty cells in the table represent uninformative or unusable transmissions. Notice that some information on allele transmission can be obtained from affected individuals with only one typed parent.

15.2.3 Scoring affected sibling pairs

In some situations it is advantageous to test for linkage disequilibrium in data sets consisting of pairs of affected offspring and their parents (Spielman et al., 1993; Olson et al., 1997). This variant of the TDT scores only the same allele transmissions to both affected offspring. This is a narrower sampling scheme than the standard affected offspring version, because transmissions from heterozygous parents that transmit a different allele to each offspring are ignored. In some situations, sampling affected sib pairs rather than affected individuals greatly improves the power of the TDT (see Figure 15.2).

Parent 1		**Parent 2**		**Child**	**Parent 1 transmission**	**Parent 2 transmission**	**Genotype transmission**
A/A	x	A/A	→	A/A			
A/A	x	A/B	→	A/A	A,A	A,B	A/A, A/B
A/A	x	A/B	→	A/B	A,A	B,A	A/B, A/A
A/A	x	B/B	→	A/B	A,A	B,B	A/B, A/B
A/A	x	B/C	→	A/B	A,A	B,C	A/B, A/C
A/B	x	A/B	→	A/A	A,B	A,B	A/A, B/B
A/B	x	A/B	→	A/B			
A/B	x	A/C	→	A/A	A,B	A,C	A/A, B/C
A/B	x	A/C	→	A/B	B,A	A,C	A/B, A/C
A/B	x	A/C	→	B/C	B,A	C,A	B/C, A/A
A/B	x	B/C	→	A/B	A,B	B,C	A/B, B/C
A/B	x	C/C	→	A/C	A,B	C,C	A/C, B/C
?/?	x	A/A	→	A/A			
?/?	x	A/A	→	A/B		A,A	
?/?	x	A/B	→	A/A			
?/?	x	A/B	→	A/B			
?/?	x	A/B	→	A/C		A,B	
?/?	x	?/?	→	A/A			
?/?	x	?/?	→	A/B			

Table 15.1: Transmission scores for all possible distinct configurations of parents and offspring.

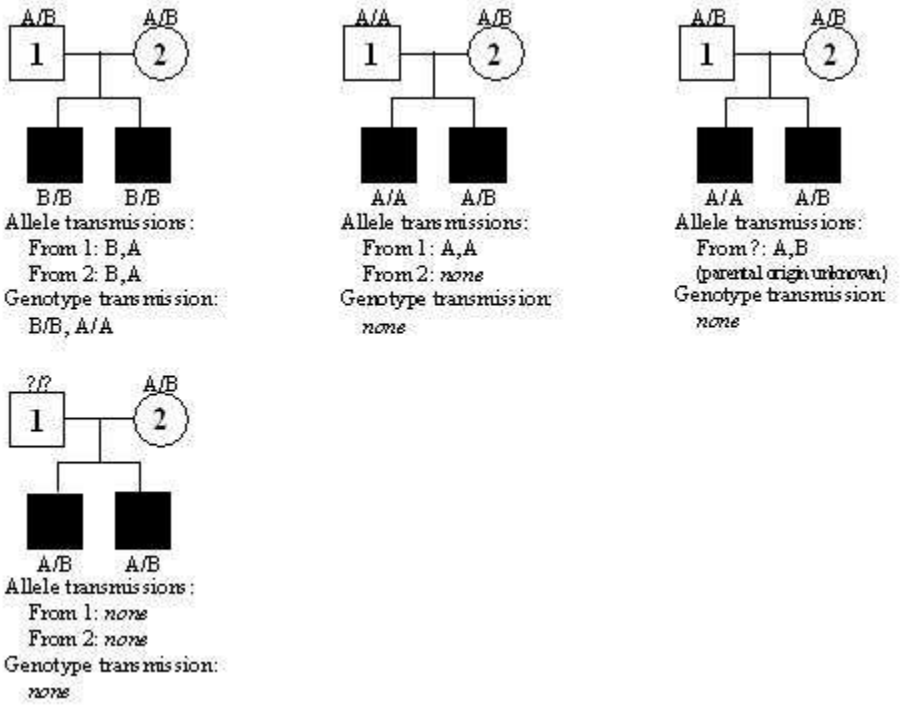


Figure 15.2: Allele and Genotype Transmission to Individual Offspring

Table 2 represents the scoring function used by TDTEX for affected sibling pairs. The basic possible allele configurations for parents and children are shown, together with the resulting allele and genotype transmissions. Empty cells in the table represent uninformative or unusable transmissions. Notice that some information on allele transmission can be obtained from affected pairs with only one typed parent.

Parent 1		Parent 2		Child 1	Child 2	Parent 1 allele transmission	Parent 2 allele transmission	Genotype transmission
A/A	x	A/A	→	A/A	A/A			
A/A	x	A/B	→	A/A	A/A	A,A	A,B	A/A, A/B
A/A	x	A/B	→	A/A	A/B	A,A		
A/A	x	A/B	→	A/B	A/B	A,A	B,A	A/B, A/A
A/A	x	B/B	→	A/B	A/B	A,A	B,B	A/B, B/B
A/A	x	B/C	→	A/B	A/B	A,A	B,C	A/B, A/C
A/A	x	B/C	→	A/B	A/C	A,A		
A/A	x	B/C	→	A/C	A/C	A,A	C,B	A/C, A/B
A/B	x	A/B	→	A/A	A/A	A,B	A,B	A/A, B/B
A/B	x	A/B	→	A/A	A/B	A,B*		
A/B	x	A/B	→	A/A	B/B			
A/B	x	A/B	→	A/B	A/B			
A/B	x	A/C	→	A/A	A/A	A,B	A,C	A/A, B/C
A/B	x	A/C	→	A/A	A/B		A,C	
A/B	x	A/C	→	A/A	B/C			
A/B	x	A/C	→	A/B	A/B	B,A	A,C	A/B, A/C
A/B	x	A/C	→	A/B	A/C			
A/B	x	A/C	→	A/B	B/C	B,A		
A/B	x	A/C	→	A/C	A/C	A,B	C,A	A/C, A/B
A/B	x	A/C	→	A/C	B/C		C,A	
A/B	x	A/C	→	B/C	B/C	B,A	C,A	B/C, A/A
A/B	x	C/D	→	A/C	A/C	A,B	C,D	A/C, B/D
A/B	x	C/D	→	A/C	A/D	A,B		
A/B	x	C/D	→	A/C	B/C		C,D	
A/B	x	C/D	→	A/C	B/D			
?/?	x	A/A	→	A/A	A/A			
?/?	x	A/A	→	A/B	A/B		A,A	
?/?	x	A/B	→	A/A	A/A			
?/?	x	A/B	→	A/A	A/B			
?/?	x	A/B	→	A/B	A/B			
?/?	x	A/B	→	A/B	B/B			
?/?	x	A/B	→	A/C	A/C		A,B	
?/?	x	A/B	→	A/C	B/C			

Table 15.2: Transmission scores for all possible distinct configurations of parents and two offspring receiving the same transmission

* - parental origin is unknown

15.2.4 Transmission Tables

To test for differences between the distribution of transmitted alleles and genotypes and non-transmitted alleles and genotypes, TDTEX tabulates all the pairs of transmissions and non-transmissions into contingency tables, henceforth called "transmission tables".

Let $M_1..M_K$ represent the K alleles or genotypes at a given marker locus. Transmission tables are defined to be K x K tables of counts, where the rows represent transmitted alleles or genotypes, and columns are the non-transmitted alleles or genotypes (Table 15.3). The entries n_{ij} are the number of times M_i was transmitted and M_j was not transmitted to an affected individual/pair.

The diagonal elements of the table, (when scoring allele transmissions, those from homozygous parents) contain no information and are ignored in the analysis.

	Non-transmitted				
Transmitted	$\mathbf{M_1}$	$\mathbf{M_2}$	...	$\mathbf{M_K}$	*Total*
$\mathbf{M_1}$	n_{12}	n_{12}	...	n_{1K}	$n_{1\bullet}$
$\mathbf{M_2}$	n_{21}	n_{22}	...	n_{2K}	$n_{2\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{M_K}$	n_{K1}	n_{K2}	...	n_{KK}	$n_{K\bullet}$
Total	$n_{\bullet 1}$	$n_{\bullet 2}$	...	$n_{\bullet k}$	$n_{\bullet\bullet}$

Table 15.3: The structure of a transmission table

15.2.5 Pedigree sampler

The pedigree sampler is the component of TDTEX that controls the construction of transmission tables. It traverses the pedigree data, identifies potentially informative individuals and pairs based on trait and marker data, scores them, and tabulates the results into a transmission table. For each nuclear family considered, the sampler first attempts to find any *informative* affected sibling pairs, up to a user-specified maximum number. This maximum can be set to zero to disable the sampling of affected sibling pairs, or to an unlimited value to select as many as possible. The sampler will only allow each child to participate in at most one transmission, so there is no problem with overlapping affected sibling pairs. The remaining offspring not already used in a sibling pair are then scored, up to a separate user-specified maximum number. This maximum can also be set to zero to disable the sampling of affected sibling pairs, or to an unlimited value to select as many as possible.

The traditional TDT test corresponds to setting the maximum number of affected children per nuclear family to 1 and the maximum number of affected sibling pairs to none. The sampler will then score the first informative allele or genotype transmission to an affected offspring, and then move on to score the next nuclear family. This will result in a valid test of allelic association in the presence of linkage.

Some other implementations of the TDT test work by setting the maximum number of affected children per nuclear family to unlimited, and the number of affected sibling pairs to none. This allows the sampler to score all informative affected offspring in each nuclear family. Similarly, basic TDT tests utilizing only sibling pairs are possible by setting the maximum number of affected

offspring to none, and the maximum number of affected sibling pairs to 1 or unlimited. This will result in valid tests of linkage in the presence of allelic association.

An interesting option exists to enable the sampling of both affected sibling offspring *and* affected sibling pairs. This very general variation gives preference to informative affected sibling pairs over affected offspring. Overall, this configuration provides a way to take advantage of more information from data sets that include a mixture of family types, not all of which have two affected offspring. Equal weight is given to all transmissions, so power may not be optimal in spite of the larger sample size.

15.2.6 Testing significance of transmission tables

Two null hypotheses have been proposed to test transmission tables for deviations from the expected pattern of allele and genotype transmissions. The first hypothesis is that of complete symmetry between the transmitted and non-transmitted alleles. This states that the expected number of any transmission type is equal to the expected number of observed transmission of the opposite pattern, i.e., $E(n_{ij}) = E(n_{ji})$. The second hypothesis is the hypothesis of marginal homogeneity: in this case, the number of alleles or genotypes transmitted is compared to the number not transmitted, i.e., $E(n_{i\bullet}) = E(n_{\bullet j})$. Which null hypothesis is optimal depends on the sample size, number and distribution of alleles, and the structure of the disequilibrium present in the sample. TDTEX provides tests based on both hypotheses for maximum flexibility.

TDTEX also includes both exact and asymptotic tests. Exact tests, as the name suggests, provide exact significance levels at the expense of being computationally intensive. Asymptotic tests are based on distributional theory and approximations that are only precise for very large sample sizes. They tend to be very quick to compute, but there are situations when asymptotic tests are significantly less powerful than exact versions. Typically, this occurs when sample sizes are small, transmission tables are sparse, and cells have less than 5 observations.

Statistics based on both the hypotheses of complete symmetry and marginal homogeneity may be applied to tables of allele transmissions as well as genotype transmissions. Genotype transmission tables may be preferred because the transmission patterns of the two parents, which include transmission from the homozygous parents, are not independent in the multiallelic case, except when linkage is complete (Bickeböller and Clerget-Darpoux, 1995). However, because of the larger size and increased sparseness of genotype transmission tables for markers with multiple alleles, the marginal homogeneity test is less prone than the complete symmetry test to problems arising from table sparseness.

15.2.6.1 Asymptotic Tests

Under the hypothesis of complete symmetry, the NcNemar test statistic

$$T_{mc} = \sum_{i<j} \frac{(n_{ij} - n_{ji})^2}{n_{ij} + n_{ji}} \sim \chi^2_{K(K-1)/2}$$

has an asymptotically χ^2 distribution with K(K - 1)/2 degrees of freedom (Bickeböller and Clerget-Darpoux, 1995). In practice, the number of degrees of freedom equals the number of types of

parental heterozygotes in the sample. A continuity corrected version of the NcNemar test statistic

$$T_{mcc} = \sum_{i<j} \frac{(|n_{ij} - n_{ji}| - 1)^2}{n_{ij} + n_{ji}} \sim \chi^2_{K(K-1)/2}$$

is also provided, since it tends to be more robust to small sample sizes.

Under the hypothesis of marginal homogeneity, the test statistic

$$T_{mh} = \frac{K-1}{K} \sum_i \frac{(n_{i.} - n_{.i})^2}{n_{i.} + n_{.i} - 2n_{ii}} \sim \chi^2_{K-1}$$

has an asymptotically χ^2 distribution with K-1 degrees of freedom, provided the table margins are independent of each other (Spielman and Ewens, 1996).

15.2.6.2 Exact tests

The exact test of complete symmetry or marginal homogeneity is generally a more powerful test than the asymptotic tests in the presence of table sparseness and/or a small sample size. To obtain the null permutation distribution for the exact test, we write the distribution of the n_{ij}, conditional on the sums of complementary off-diagonal cells, as the product of K(K-1)/2 binomial random variables with equal probability of transmission vs. non-transmission:

$$Pr(n) = \prod_{i<j} \left(\begin{array}{c} n_{ij} + n_{ji} \\ n_{ij} \end{array} \right) \left(\frac{1}{2} \right)^{n_{ij}+n_{ji}}$$

An exact significance level is determined by calculating the probability of finding a permutation of the observed data, conditional on the sums of complementary off-diagonal cells, which is as extreme as, or more extreme than, the observed transmission table. Let $N = \{n' : n'_{ij} + n'_{ji} = n_{ij} + n_{ji}\}$ be the set of all permutations of the observed data, conditional on the sums of complementary off-diagonal cells. Let $N' = \{n' : Pr(n') \leq Pr(n),\ n' \in N\}$, be the set of all permutations with probability less than or equal to that of the observed data. Then the significance level, or p-value, is $P_{cs} = \sum_{n \in N'} Pr(n')$.

Since enumerating all possible permutations of the observed transmission table is infeasible for larger tables, the exact permutation algorithm relies upon methods of ordering permutations of the observed table, and by avoiding the evaluation of many equivalent tables. The algorithm uses the fact that the probability after permuting a pair of symmetric odd-diagonal cells in a transmission table does not involve the remaining cells. The null probability distribution is also independent of the direction of asymmetry. For example, a configuration in which $n_{12} = 4$ and $n_{21} = 0$ has the same probability as that of $n_{12} = 0$ and $n_{22} = 4$.

15.2.6.3 Monte Carlo Approximations

As the transmission table size and number of marker alleles increases, program execution time of the exact permutation test becomes prohibitively slow. For transmission tables with greater than about 300 observations, or with more than about 8 alleles, the Monte Carlo approximation is recommended. Instead of considering every possible permutation, a random sample from the set of all possible permutations, conditional on the observed transmission table, is taken.

The proportion of permutations with significance equal to or greater than the observed table is computed. This proportion is an estimate of the exact pvalue of the observed table. The standard error of the estimated p-value is obtained by computing the variance among several batches of permutations. The total number of permutations considered is chosen to estimate the resulting p-value within 20% of its true value with 95% confidence.

15.3 Program Input

File Type	Description
Parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual including fields for identifiers, sex, parents, trait and marker data.

15.3.1 The `tdtex` Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main TDTEX parameter.

`parameter [, attribute]`	**Explanation**	
`tdtex`	Starts a TDTEX specification block.	
	Value Range	N/A
	Default Value	None
	Required	Yes
	Applicable Notes	None
`, out`	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension.	
	Value Range	Character string representing a valid file name.
	Default Value	tdtex
	Required	No
	Applicable Notes	None

15.3.2 The `tdtex` Block

The following syntax table lists the permissible parameter and attribute settings for the `tdtex` block.

`parameter [, attribute]`	**Explanation**
`marker`	Specifies a marker for which transmissions are scored.
	Value Range: Character string representing the name of a valid marker listed in the pedigree data file.
	Default Value: None
	Required: Yes
	Applicable Notes: 4
`trait`	Specifies a trait denoting affection status for offspring and sibling pairs.
	Value Range: Character string representing the name of a valid trait, phenotype or covariate listed in the pedigree data file.
	Default Value: None
	Required: Yes
	Applicable Notes: None
`parental_trait`	Specifies a trait used as an indicator variable to select subsets of pairs to analyze.
	Value Range: Character string representing the name of a valid trait, phenotype or covariate listed in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: None
`sample`	Specifies which type of transmission is to be scored.
	Value Range: {alleles, genotypes}
	Default Value: alleles
	Required: No
	Applicable Notes: None
`max_children`	Specifies the maximum number of informative affected offspring transmissions per nuclear family that the sampler may use.
	Value Range: none unlimited {0, 1, 2, 3, ...}
	Default Value: None
	Required: No
	Applicable Notes: 1, 3

`max_sib_pairs`	Specifies the maximum number of informative affected sibling pair transmissions per nuclear family that the sampler may use. Value Range: none unlimited {0, 1, 2, 3, ...} Default Value: None Required: No Applicable Notes: 2, 3
`sex_differential`	Causes three tests to be performed: 1. one scoring transmissions from all parents, 2. one that scores only paternal transmissions, and 3. one that scores only maternal transmissions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`skip_exact_tests`	Specifies that no exact tests are to be performed. This option is shorthand for setting all three of the parameters: `skip_permutation_test`, `skip_mc_test` and `skip_mcmh_test`. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
`skip_permutation_test`	Specifies that the exact permutation McNemar test should not be performed. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
`skip_mc_test`	Specifies that the exact Monte Carlo McNemar test should not be performed. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
`skip_mcmh_test`	Specifies that the exact Monte Carlo marginal homogeneity test should not be performed. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None

1. A classic TDT can be performed by setting `max_children` to 1 and `max_sib_pairs` to **none**. All affected children in a pedigree can be used as if they are independent by setting `max_children` to **unlimited** and `max_sib_pairs` to **none**.

2. A sibling TDT using only one sib-pair per pedigree can be performed by setting `max_children` to **none** and `max_sib_pairs` to 1. A sibling TDT using all sibling-pairs in a pedigree as if they are independent can be performed by setting `max_children` to **none** and `max_sib_pairs` to **unlimited**.

3. Regardless of the values of `max_children` and `max_sib_pairs`, pedigrees must have at least one typed parent.

4. The user may list as many different `marker` parameters as desired. If no `marker` parameters are specified, then the default TDTEX behavior is to score transmissions for all markers found in the pedigree data file.

15.4 Program Execution

TDTEX is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running TDTEX from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line.

```
>tdtex
S.A.G.E. v5.x -- TDTEX
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY.
usage: tdtex <parameters>  <pedigree>
Command line parameters:
parameters - parameter file
pedigree - pedigree data file
```

As indicated in the program usage statement, input files are listed on the command line. A typical run of TDTEX may look like the following:

```
>tdtex tdtex.par example.ped
S.A.G.E. v5.x -- TDTEX
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY
Reading Parameter File.....................done.
Reading Pedigree File.....................
from example.ped............done.
Sorting pedigrees.........................done.
Performing TDT analysis ..................done.
```

15.5 Program Output

TDTEX produces several output files that contain results and diagnostic information:

Filename	File Type	Description
tdtex.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
tdtex.out	TDTEX analysis output file	Contains the results of each TDT analysis.

15.5.1 Information Output File

The TDTEX Information Output file contains a variety of useful information, including:

- Information on fields read from the pedigree data file. These tables, which provide information about what the program has read in, are included with all programs in the S.A.G.E. and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that you check this file for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not run as expected. The file "tdtex.inf" should be checked for errors and diagnostic information after each run of the program.

15.5.2 TDTEX Analysis Output File

One analysis output file, named "tdtex.out", is generated per run of TDTEX. It contains the results of all tests.

15.6 Example Output Files

```
S.A.G.E. v5.x -- TDTEX
COPYRIGHT (C) 2002 CASE WESTERN RESERVE UNIVERSITY
  Allele transmissions to affected children:
  ------------------------------------------
  NOTICE: 15 error(s) were found in your sample!  See the Information Output
          File for a detailed description of each problem.
  Marker: mrk1
  Trait:  Trait1
  Max. children/family:  1
  Max. sib pairs/family: none
  [13 empty rows/columns not shown]
  T\NT|  A|  C|  F|  G|
  ----|---------------
     A|  0   1   1   1
     C|  0   0   0   0
     F|  0   0   0   0
     G|  0   0   0   0
               Informative / Total = % Informative
  Pedigrees          :   1 /   2 =  50.00%
  Families           :   3 /  10 =  30.00%
  Affected children  :   3 /   3 = 100.00%
  Affected sib pairs :   0 /   0 =   0.00%
  Sample size        :   3 /   3 = 100.00%
  Exact test statistics               p-value       (std. error)
  --------------------------------    ----------    ------------
  Exact McNemar test                  1.00000000
  Monte Carlo McNemar test            1.00000000    (0.00000000)
  Monte Carlo Marginal Homogeneity    0.25012750    (0.00111201)
  Asymptotic test statistics          p-value
  --------------------------------    ----------
  McNemar test                        0.39162518
  Continuity corrected McNemar test   1.00000000
  Marginal homogeneity test           0.21229020
```

Chapter 16

AGEON

This program produces maximum likelihood estimates of the parameters of a mixed power-normal distribution for a binary trait (affect versus unaffected) with variable age of onset. These estimates can be used in a special function to produce either of two new variables, for a binary trait with variable age of onset, to be used in a SIBPAL analysis. The mean, variance, and susceptibility can each depend linearly on covariates. By default a class susceptibility covariate is generated according to the value of the parental binary trait(s).

16.1 Limitations

No account is taken of ascertainment or familial correlations; i.e. all individuals are assumed to be randomly sampled. This does not affect the validity or robustness of any SIBPAL analysis. Genetic susceptibilities are not estimated for classes with fewer than 5 informative members. If for any reason the power parameter λ_1 is fixed by the user at 0, then the initial value of the shift parameter λ_2 *must* be greater than the inverse of the minimum value for age-of-onset or age-at-exam, whichever is smaller.

16.2 Theory

The purpose of the AGEON program is to estimate the parameters needed to calculate either of two new quantitative variable that can be used in SIBPAL. These variables are to detect linkage to

1. genes that affect susceptibility to disease, and
2. genes that affect age of onset of disease.

16.2.1 Susceptibility

Susceptibility to disease conditional on whether the individual is affected or not by age a'_j is given by

$$x_j = \begin{cases} 1 & \text{, if affected} \\ \dfrac{\gamma_j - \gamma_j \Phi\left[\dfrac{\frac{\left(a'_j+\lambda_2\right)^{\lambda_1}}{\lambda_1}-\mu_j}{\sigma_j}\right]}{1-\gamma_j \Phi\left[\dfrac{\frac{\left(a'_j+\lambda_2\right)^{\lambda_1}}{\lambda_1}-\mu_j}{\sigma_j}\right]} & \text{, if not affected by age } a'_j \end{cases}$$

where Φ is the standard cumulative normal distribution function.

The disease age of onset is given by the survival analysis residual

$$x_j = \begin{cases} 1 - \gamma_j \Phi\left[\dfrac{\frac{\left(a_j+\lambda_2\right)^{\lambda_1}}{\lambda_1}-\mu_j}{\sigma_j}\right] & \text{, if affected at age } a_j \\ -\gamma_j \Phi\left[\dfrac{\frac{\left(a'_j+\lambda_2\right)^{\lambda_1}}{\lambda_1}-\mu_j}{\sigma_j}\right] & \text{, if not affected by age } a'_j \end{cases}$$

where again Φ is the standard cumulative normal distribution function.

The program AGEON estimates λ_1, λ_2, μ and γ wherein the parameters are defined, possibly as functions of specified covariates, as follows:

$$\mu_j = \mu_0 + \xi_1 x_{1j} + \xi_2 x_{2j} + \cdots,$$

$$\gamma_j = \frac{e^{\theta_j}}{1+e^{\theta_j}}, \text{ where } \theta_j = \gamma_0 + \xi_1 x_{1j} + \xi_2 x_{2j} + \cdots, \text{ and}$$

$$\sigma^2 = \sigma_0^2 + \xi_1 x_{1j} + \xi_2 x_{2j} + \cdots$$

In particular, γ_0 can be a function of parental affection status if sufficient parental data are available.

Using one of these values of *x* as a quantitative trait can be more powerful in the usual Haseman-Elston test for linkage than using disease status as a simple binary trait (Zhu, et al., 1997; Hanson and Knowler, 1998).

The log likelihood maximized is $\sum_i^n \ln L(i)$, where n is the number of sibs in the sample and $L(i)$ is the likelihood for the i-th sib. Let a_i denote age of onset and a'_i the age at examination, the latter being available for all unaffected persons to be included in the analysis. Define

$$\varphi(x,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right\}$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left\{-\frac{1}{2}u^2\right\} du$$

$$h(x) = \begin{cases} \frac{(x+\lambda_2)^{\lambda_1}-1}{\lambda_1}, & \text{if } \lambda_1 \neq 0 \\ ln(x+\lambda_2), & \text{if } \lambda_1 = 0 \end{cases}$$

$$sign\,(x) = \begin{cases} 1, \text{ if x } \geq 0 \\ -1, \text{ if x < 0} \end{cases}$$

Then the likelihood $L\,(i)$ is given by:

Category	$L\,(i)$
Affected individuals with known age of onset	$L(i) = \gamma_i \varphi \left[h(a_i) - h(\mu_i), \sigma_i^2\right] (a_i + \lambda_2)^{\lambda_1 - 1}$
Affected individuals with unknown age of onset	$L^*(i) = \gamma_i \Phi \left[\frac{h(a_i') - h(\mu_i)}{\sigma_i}\right]$
Unaffected individuals	$L(i) = 1 - L^*(i)$

As mentioned above, the mean μ_i, variance σ_i^2 , and susceptibility γ_i, may depend on covariates. In the case of susceptibility, the logit is assumed to be a linear function of the covariates.

Because $a_i + \lambda_2$ must be positive to apply the Box and Cox (1964) transformation, so that prior to transformation $a_i + \lambda_2$ cannot strictly follow a normal distribution, there the maximization is also performed using the following likelihoods, which allow for the truncation (see Pericak-Vance et al, 1983):

Category	$L\,(i)$
Affected individuals with known age of onset	$L\,(i) = \frac{\gamma_i \varphi\left[h(a_i) - h(\mu_i), \sigma_i^2\right] (a_i + \lambda_2)^{\lambda_1 - 1}}{\Phi\left[sign(\lambda_1)\left(\frac{\mu_i - h(0)}{\sigma_i}\right)\right]}$
Affected individuals with unknown age of onset	$L^*\,(i) = \frac{\gamma_i sign(\lambda_1)\left[\Phi\left(\frac{h(a_i') - h(\mu_i)}{\sigma_i}\right) - \Phi\left(\frac{-\frac{1}{\lambda_1} + h(\mu_i)}{\sigma_i}\right)\right]}{\Phi\left[sign(\lambda_1)\left(\frac{\mu_i - h(0)}{\sigma_i}\right)\right]}$
Unaffected individuals	$L\,(i) = 1 - L^*\,(i)$

16.3 Program Input

File Type	Description
Parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data File	Contains delimited records for each individual including fields for identifiers, parents, trait and covariates.

16.3.1 The ageon Parameter

The following syntax table specifies the permissible parameter and attribute settings for the main ageon parameter.

parameter [, attribute]	**Explanation**
ageon	Starts an AGEON analysis block Value Range: N/A Default Value: None Required: Yes Applicable Notes: None
, out	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension. Value Range: Character string representing a valid file name. Default Value: ageon Required: No Applicable Notes: None

16.3.2 The ageon Parameter Block

The following table shows the syntax for the ageon block:

parameter [, attribute]	**Explanation**
title	Specifies a title for the analysis defined within the block. Value Range: Quoted character string Default Value: "AGEON Analysis" Required: No Applicable Notes: None
affected, affectedness	Specifies name of a binary trait containing affection status. Must be the name of a trait, covariate or phenotype in the pedigree data file or created by means of a function block. Value Range: Character string representing a valid file name. Default Value: None Required: Yes Applicable Notes: None
age_of_onset	Specifies name of a trait containing age of onset for affected individuals. Must be the name of a trait, covariate or phenotype in the pedigree data file or created by means of a function block. Value Range: Character string Default Value: None Required: Yes Applicable Notes: 1
age_of_exam	Specifies name of a trait containing age of examination for unaffected individuals. Must be the name of a trait, covariate or phenotype in the pedigree data file or created by means of a function block. Value Range: Character string Default Value: None Required: Yes Applicable Notes: 1
allow_averaging	Specifies option to substitute covariate mean values for missing covariate data Value Range: {mean, none} Default Value: none Required: No Applicable Notes: 2

`mean_cov`	Starts a sub-block for specifying covariates for the mean. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`var_cov`	Starts a sub-block for specifying covariates for the variance. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`suscept_cov`	Starts a sub-block for specifying covariates for the trait susceptibility. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`transformation`	Starts a sub-block for specifying transformation options. Value Range: N/A Default Value: None Required: No Applicable Notes: 3, 4

Notes:

1. The `age_of_onset` parameter and the `age_of_exam` parameter can both specify the same pedigree data field.

2. If the value of **none** is specified and any single individual's covariate value is missing, then that individual will be treated as uninformative for the purpose of the analysis. If **mean** is specified, missing covariate values will be replaced with the covariate's mean value as calculated from the sample.

3. See section 5.3.2.2 for details on the transformation theory implemented in this program.

16.3.3 Sub-Block Syntax: `mean_cov`

The following table shows the syntax for the `mean_cov` sub-block:

`parameter [, attribute]`	**Explanation**	
`covariate`	Covariate to modify the mean value of the age of onset. This parameter may be specified multiple times.	
	Value Range	Character string representing the name of a trait, covariate or phenotype from the pedigree data file or a name created by means of a `function` block.
	Default Value	None
	Required	No
	Applicable Notes	1
`, val`	Specifies the value of the covariate coefficient.	
	Value Range	(-∞, +∞)
	Default Value	None
	Required	No
	Applicable Notes	None
`, fixed`	Specifies option to fix the given value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	2

Notes:

1. The default is to include no mean covariates in the analysis. The means indicated in the `type_mean` sub-block are a linear function of this covariate. All covariates are centered, the centering (average) value being included as part of the output.

2. The `fixed` attribute, used in conjunction with the `val` attribute, specifies that the covariate's value will remain fixed at the given value and will not be estimated by the program.

16.3.4 Sub-Block Syntax: `var_cov`

The following table shows the syntax for the `var_cov` sub-block:

`parameter [, attribute]`	**Explanation**	
`covariate`	Covariate to modify the variance of the transformed age of onset. This parameter may be specified multiple times.	
	Value Range	Character string representing the name of a trait, covariate or phenotype from the pedigree data file or a name created by means of a `function` block.
	Default Value	None
	Required	No
	Applicable Notes	1
`, val`	Specifies value of the covariate coefficient	
	Value Range	(-∞, +∞)
	Default Value	None
	Required	No
	Applicable Notes	None
`, fixed`	Specifies option to fix the given value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	2

Notes

1. The default is to include no covariates in the analysis. The variances indicated in the `type_var` sub-block are a linear function of this covariate . All covariates are centered, the centering (average) value being included as part of the output.

2. The `fixed` attribute, used in conjunction with the `val` attribute, specifies that the covariate's value will remain fixed at the given value and will not be estimated by the program.

16.3.5 Sub-Block Syntax: `suscept_cov`

The following table shows the syntax for the `suscept_cov` sub-block:

`parameter [, attribute]`	**Explanation**	
`covariate`	Covariate to modify the mean of a continuous trait. This parameter may be specified multiple times.	
	Value Range	Character string representing the name of a trait, covariate or phenotype from the pedigree data file or a name created by means of a `function` block.
	Default Value	None
	Required	No
	Applicable Notes	1
`, val`	Specifies the value of the covariate coefficient.	
	Value Range	(-∞, ∞)
	Default Value	None
	Required	No
	Applicable Notes	None
`, fixed`	Specifies option to fix this value.	
	Value Range	{true, false}
	Default Value	false
	Required	No
	Applicable Notes	2
`class`	Starts a sub-block for specifying classification options.	
	Value Range	N/A
	Default Value	None
	Required	No
	Applicable Notes	None

Notes

1. The default is to include no susceptibility covariates in the analysis. The `suscept_cov` sub-block indicates which covariates are to modify the logits of susceptibilities. All covariates are centered, the centering (average) value being included as part of the output.

2. The `fixed` attribute, used in conjunction with the `val` attribute, specifies that the covariate's value will remain fixed at the given value and will not be estimated by the program.

16.3.6 Sub-Block Syntax: `transformation`

The following table shows the syntax for the `transformation` sub-block:

`parameter [, attribute]`	**Explanation**
`lambda1`	Specifies of the power parameter, λ_1 Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`, val`	Specifies the value for λ_1. Value Range: (-∞, +∞) Default Value: 1.0 Required: No Applicable Notes: None
`, fixed`	Specifies option to fix $\lambda_1$at the given value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`lambda2`	Specifies the shift parameter, λ_2 Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`, val`	Specifies the value for λ_2. Value Range: (-∞, +∞) Default Value: 0.05 Required: No Applicable Notes: 2 None
`, fixed`	Option to fix $\lambda_2$at the given value. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None

16.3.7 class sub-block

The following lists all parameters that may occur in a `class` sub-block.

parameter [, attribute]	**Explanation**
`trait`	Specifies the name of an alternate trait on which to base the individual's classification code. Value Range: Character string Default Value: None Required: No Applicable Notes: None
`num_of_classes`	Specifies the number of classes to be considered during the analysis Value Range: {1, 2, 3, ...} Default Value: 6 Required: No Applicable Notes: None

16.3.8 Sample parameter file

```
pedigree {
   delimiter_mode = multiple
   delimiters = " "
   individual_missing_value = "0"
   sex_code, male = "1", female = "2", unknown = "?"
   pedigree_id = fam
   individual_id = id
   parent_id = mom
   parent_id = dad
   sex_field = sex
   trait = aff, binary, affected = 1, unaffected = 0, missing = -1
   trait = ao, missing = -1
   trait = ae, missing = -1
   trait = cov1, missing = -999
   trait = cov2, missing = -999
   trait = classx
}
ageon {
   title = "analysis"
   affectedness = aff
   age_of_onset = ao
   age_of_exam = ae

   mean_cov {
      covariate = cov2
   }

   suscept_cov {
      covariate = cov1
      class {
         trait = classx
         num_of_classes = 10
      }
   }
}
```

16.3.9 Exclusion Criteria for Individuals and Pedigrees

Under some conditions AGEON will exclude individuals and/or pedigrees from analysis, and the user is advised to take note of program outputs that indicate the numbers of valid and invalid individuals being counted. The exclusion criteria are:

1. Any structurally invalid families will be excluded from analysis (see 16.1).
2. Any individual whose primary trait is missing or who is missing at least one covariate value (if the `allow_averaging` option is disabled – the default behavior) will be retained for

analysis, but all of its trait information will be treated as missing.

16.3.10 Program Execution

AGEON is run via a command line interface on the supported UNIX and Windows platforms. This requires that the S.A.G.E. programs are properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running AGEON from the command prompt with no arguments, or the wrong number of arguments, will result in the program printing its usage statement. This lists the input files the program requires on the command line:

```
>ageon
S.A.G.E. v5.x -- AGEON
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
usage: ./ageon <parameters>  <pedigree>
Command line parameters:
parameters - Parameter File
pedigree - Pedigree Data File
As indicated in the program usage statement, input files are
listed on the command line. A typical run of AGEON may look like
the following:
>ageon ageon.par example.ped
S.A.G.E. v5.x -- AGEON
COPYRIGHT (C) CASE WESTERN RESERVE UNIVERSITY.
Reading parameter file.....................done.
Reading pedigree file.....................
from example.ped............done.
Sorting pedigrees.........................done.
Parsing Ageonset Analyses...
Beginning new analysis block....
Analysis parsing complete. Analysis valid.
============================================================
Importing individual data into AGEON...
Import complete.
============================================================
```

16.3.11 Program Output

Output files produced by AGEON containing results and diagnostic information are:

File Name	File Type	Description
ageon.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
ageon.out	AGEON summary output file	Contains the table of final estimates of the parameters and their standard errors and other results.

16.3.11.1 Information Output File

The AGEON Information file contains a variety of useful information, including:

- Information on fields read from the Pedigree Data File. These tables, which provide information about what the program has read from the Pedigree Data File, are included with all programs in S.A.G.E. Release and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that this file be checked for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "ageon.inf" should be checked for errors and diagnostic information after each run of the program.

16.3.12 Output Files

The AGEON Output Files present the tables of final estimates of the parameters, along with relevant model information. By default a class susceptibility covariate is generated according to the value of the parental binary trait(s), as shown in the following table:

Class	**Description**
0	Both parents are unknown
1	One of the parents is unknown, the other is affected
2	One of the parents is unknown, the other is unaffected
3	Both parents are affected
4	One of the parents is affected, the other is unaffected
5	Both parents are unaffected

16.3.12.1 Example Summary Output File

The summary output file contains descriptive information about each of the six classifications, as well as final estimates, standard errors, and p-values of the parameters estimated in the model, including

- mean intercept and covariates
- susceptibility intercept(s) and covariates
- variance intercept and covariates
- transformation parameters ($\lambda_1$and λ_2).

The file also includes a likelihood ratio test statistic for the comparison of separate susceptibilities for each category of the classification variable against a common susceptibility for all categories.

Here is an example of an AGEON summary output file:

```
============================
     Sample description
============================
  Number of pedigrees in dataset             200
  Number of analyzable pedigrees             200

  Number of individuals in dataset           787
  Number of analyzable individuals           787
  Number of analyzable invalid individuals   149
  Number of analyzable valid individuals     638
============================
     MODEL DESCRIPTION
============================
  Title               AGEONSET Analysis 1
  Affectedness trait  AFF
  Age-of-onset trait  AO
  Age-at-exam trait   AE
==============================
     CLASSIFICATION SYSTEM
==============================
  Using default classification system:

   0  Both parental traits are unknown.
   1  One of the parental traits is unknown, the other is affected.
   2  One of the parental traits is unknown, the other is unaffected.
   3  Both parental traits are affected.
   4  One of the parental traits is affected, the other is unaffected.
   5  Both parental traits are unaffected.
==========================
     CLASS STATISTICS
==========================
=================
     CLASS 0
=================
  TOTAL NUMBER OF INDIVIDUALS USED IN ANALYSIS          60
  NUMBER OF INDIVIDUALS WITH AN AGE OF ONSET            25
  MEAN OF AGE OF ONSET                           75.458537
```

```
  VARIANCE OF AGE OF ONSET                          4.872671
  NUMBER OF INDIVIDUALS AFFECTED                           9
  PROPORTION OF INDIVIDUALS AFFECTED                0.150000
  MEAN OF AGE AT EXAM OF THE UNAFFECTED            74.816667
  VARIANCE OF AGE AT EXAM OF THE UNAFFECTED         6.949722
=================
     CLASS 1
=================
  TOTAL NUMBER OF INDIVIDUALS USED IN ANALYSIS            80
  NUMBER OF INDIVIDUALS WITH AN AGE OF ONSET              33
  MEAN OF AGE OF ONSET                              74.696970
  VARIANCE OF AGE OF ONSET                          6.514233
  NUMBER OF INDIVIDUALS AFFECTED                          20
  PROPORTION OF INDIVIDUALS AFFECTED                0.250000
  MEAN OF AGE AT EXAM OF THE UNAFFECTED            75.350000
  VARIANCE OF AGE AT EXAM OF THE UNAFFECTED         4.377500
.
.
.
=================
     CLASS 5
=================
  TOTAL NUMBER OF INDIVIDUALS USED IN ANALYSIS            43
  NUMBER OF INDIVIDUALS WITH AN AGE OF ONSET              15
  MEAN OF AGE OF ONSET                              76.266667
  VARIANCE OF AGE OF ONSET                          4.595556
  NUMBER OF INDIVIDUALS AFFECTED                           9
  PROPORTION OF INDIVIDUALS AFFECTED                0.209302
  MEAN OF AGE AT EXAM OF THE UNAFFECTED            75.744186
  VARIANCE OF AGE AT EXAM OF THE UNAFFECTED         5.446187
=============================================================================
     MAXIMIZATION RESULTS susceptibilities equal, no truncation
=============================================================================
-------------------------------------------------------
  Parameter           Estimate    S.E.       P-value
-------------------------------------------------------
  Susceptibility intercepts
            Class 0   -0.489288   0.162019   0.002528
            Class 1   -0.489288   0.162019   0.002528
            Class 2   -0.489288   0.162019   0.002528
            Class 3   -0.489288   0.162019   0.002528
            Class 4   -0.489288   0.162019   0.002528
            Class 5   -0.489288   0.162019   0.002528
      Mean intercept   73.682004   0.182158  < 1e-07
  Mean covariates
               cov1    0.343027   0.172083   0.046219
  Variance intercept    6.308833   1.561902   0.000054
  Variance covariates
               cov2    1.886914   1.201458   0.116294
  Transformation
            Lambda1    1.000000         Fixed
            Lambda2    0.050000         Fixed
-------------------------------------------------------
Final ln likelihood: -383.779624
=============================================================================
     MAXIMIZATION RESULTS susceptibilities free, no truncation
=============================================================================
-------------------------------------------------------
```

```
  Parameter           Estimate    S.E.       P-value
-----------------------------------------------------
  Susceptibility intercepts
            Class 0   -0.893122   0.422414   0.034487
            Class 1   -0.295137   0.330400   0.371711
            Class 2   -0.437648   0.322448   0.174696
            Class 3   -0.761468   0.647488   0.239582
            Class 4   -0.396851   0.305525   0.193974
            Class 5   -0.564788   0.451150   0.210612
     Mean intercept   73.683039   0.182786  < 1e-07
  Mean covariates
               cov1    0.346353   0.172818   0.045054
  Variance intercept   6.358667   1.578147   0.000056
  Variance covariates
               cov2    1.907274   1.214693   0.116376
  Transformation
            Lambda1    1.000000         Fixed
            Lambda2    0.050000         Fixed
-----------------------------------------------------
Final ln likelihood: -382.959654
====================
     JOINT TEST
====================
  H0 ln likelihood susceptibilities free, no truncation   -382.959654
  H1 ln likelihood susceptibilities equal, no truncation  -383.779624

  2 * |H0 - H1|                                             1.639942
  Degrees of freedom                                               5
  P-value                                                   0.896377
```

16.3.12.2 Example Detailed Output File

The detailed output file contains all information present in the summary output file, and has the following additional information:

- first partial derivatives for all parameters
- estimates for all four models (with/without truncation)
- variance-covariance matrices for all four models
- additional likelihood ratio test statistics for the models not listed in the summary output file.

Here is an example of an AGEON detailed output file:

```
============================
      Sample description
============================
   Number of pedigrees in dataset             200
   Number of analyzable pedigrees             200

   Number of individuals in dataset           787
   Number of analyzable individuals           787
   Number of analyzable invalid individuals   149
   Number of analyzable valid individuals     638
============================
      MODEL DESCRIPTION
============================
   Title               AGEONSET Analysis 1
   Affectedness trait  AFF
   Age-of-onset trait  AO
   Age-at-exam trait   AE
==============================
      CLASSIFICATION SYSTEM
==============================
   Using default classification system:

    0  Both parental traits are unknown.
    1  One of the parental traits is unknown, the other is affected.
    2  One of the parental traits is unknown, the other is unaffected.
    3  Both parental traits are affected.
    4  One of the parental traits is affected, the other is unaffected.
    5  Both parental traits are unaffected.
==========================
      CLASS STATISTICS
==========================
=================
      CLASS 0
=================
   TOTAL NUMBER OF INDIVIDUALS USED IN ANALYSIS          60
   NUMBER OF INDIVIDUALS WITH AN AGE OF ONSET            25
   MEAN OF AGE OF ONSET                           75.458537
   VARIANCE OF AGE OF ONSET                        4.872671
   NUMBER OF INDIVIDUALS AFFECTED                         9
   PROPORTION OF INDIVIDUALS AFFECTED              0.150000
   MEAN OF AGE AT EXAM OF THE UNAFFECTED          74.816667
   VARIANCE OF AGE AT EXAM OF THE UNAFFECTED       6.949722
=================
      CLASS 1
=================
   TOTAL NUMBER OF INDIVIDUALS USED IN ANALYSIS          80
   NUMBER OF INDIVIDUALS WITH AN AGE OF ONSET            33
   MEAN OF AGE OF ONSET                           74.696970
   VARIANCE OF AGE OF ONSET                        6.514233
   NUMBER OF INDIVIDUALS AFFECTED                        20
   PROPORTION OF INDIVIDUALS AFFECTED              0.250000
   MEAN OF AGE AT EXAM OF THE UNAFFECTED          75.350000
   VARIANCE OF AGE AT EXAM OF THE UNAFFECTED       4.377500

.

.

.
=================
      CLASS 5
=================
   TOTAL NUMBER OF INDIVIDUALS USED IN ANALYSIS          43
   NUMBER OF INDIVIDUALS WITH AN AGE OF ONSET            15
   MEAN OF AGE OF ONSET                           76.266667
```

```
   VARIANCE OF AGE OF ONSET                          4.595556
   NUMBER OF INDIVIDUALS AFFECTED                           9
   PROPORTION OF INDIVIDUALS AFFECTED                0.209302
   MEAN OF AGE AT EXAM OF THE UNAFFECTED            75.744186
   VARIANCE OF AGE AT EXAM OF THE UNAFFECTED         5.446187
=====================================================================
      MAXIMIZATION RESULTS susceptibilities equal, no truncation
=====================================================================
---------------------------------------------------------------------
   Parameter           Estimate    S.E.       P-value    Deriv
---------------------------------------------------------------------
   Susceptibility intercepts
              Class 0   -0.489288   0.162019   0.002528  -0.0000002360
              Class 1   -0.489288   0.162019   0.002528   0.0000000000
              Class 2   -0.489288   0.162019   0.002528   0.0000000000
              Class 3   -0.489288   0.162019   0.002528   0.0000000000
              Class 4   -0.489288   0.162019   0.002528   0.0000000000
              Class 5   -0.489288   0.162019   0.002528   0.0000000000
       Mean intercept   73.682004   0.182158  < 1e-07     0.0000006171
   Mean covariates
                 cov1    0.343027   0.172083   0.046219  -0.0000017870
   Variance intercept    6.308833   1.561902   0.000054  -0.0000006066
   Variance covariates
                 cov2    1.886914   1.201458   0.116294   0.0000008041
   Transformation
              Lambda1    1.000000                 Fixed
              Lambda2    0.050000                 Fixed
---------------------------------------------------------------------
Final ln likelihood: -383.779624
=====================================================================
      MAXIMIZATION RESULTS susceptibilities free, no truncation
=====================================================================
---------------------------------------------------------------------
   Parameter           Estimate    S.E.       P-value    Deriv
---------------------------------------------------------------------
   Susceptibility intercepts
              Class 0   -0.893122   0.422414   0.034487  -0.0000000189
              Class 1   -0.295137   0.330400   0.371711   0.0000000000
              Class 2   -0.437648   0.322448   0.174696  -0.0000002023
              Class 3   -0.761468   0.647488   0.239582   0.0000000000
              Class 4   -0.396851   0.305525   0.193974  -0.0000000337
              Class 5   -0.564788   0.451150   0.210612   0.0000000597
       Mean intercept   73.683039   0.182786  < 1e-07    -0.0000000016
   Mean covariates
                 cov1    0.346353   0.172818   0.045054  -0.0000001012
   Variance intercept    6.358667   1.578147   0.000056  -0.0000000080
   Variance covariates
                 cov2    1.907274   1.214693   0.116376  -0.0000000088
   Transformation
              Lambda1    1.000000                 Fixed
              Lambda2    0.050000                 Fixed
---------------------------------------------------------------------
Final ln likelihood: -382.959654
=====================================================================
      MAXIMIZATION RESULTS susceptibilities equal, using truncation
=====================================================================
---------------------------------------------------------------------
   Parameter           Estimate    S.E.       P-value    Deriv
---------------------------------------------------------------------
   Susceptibility intercepts
              Class 0   -0.489288   0.162019   0.002528  -0.0000002360
              Class 1   -0.489288   0.162019   0.002528   0.0000000000
              Class 2   -0.489288   0.162019   0.002528   0.0000000000
              Class 3   -0.489288   0.162019   0.002528   0.0000000000
              Class 4   -0.489288   0.162019   0.002528   0.0000000000
              Class 5   -0.489288   0.162019   0.002528   0.0000000000
       Mean intercept   73.682004   0.182158  < 1e-07     0.0000006171
   Mean covariates
                 cov1    0.343027   0.172083   0.046219  -0.0000017870
   Variance intercept    6.308833   1.561902   0.000054  -0.0000006066
   Variance covariates
                 cov2    1.886914   1.201458   0.116294   0.0000008041
   Transformation
              Lambda1    1.000000                 Fixed
              Lambda2    0.050000                 Fixed
---------------------------------------------------------------------
Final ln likelihood: -383.779624
=====================================================================
      MAXIMIZATION RESULTS susceptibilities free, using truncation
=====================================================================
---------------------------------------------------------------------
   Parameter           Estimate    S.E.       P-value    Deriv
---------------------------------------------------------------------
   Susceptibility intercepts
              Class 0   -0.893122   0.422414   0.034487  -0.0000000189
              Class 1   -0.295137   0.330400   0.371711   0.0000000000
              Class 2   -0.437648   0.322448   0.174696  -0.0000002023
              Class 3   -0.761468   0.647488   0.239582   0.0000000000
              Class 4   -0.396851   0.305525   0.193974  -0.0000000337
              Class 5   -0.564788   0.451150   0.210612   0.0000000597
       Mean intercept   73.683039   0.182786  < 1e-07    -0.0000000016
   Mean covariates
```

```
                cov1    0.346353   0.172818   0.045054  -0.0000001012
   Variance intercept    6.358667   1.578147   0.000056  -0.0000000080
   Variance covariates
                cov2    1.907274   1.214693   0.116376  -0.0000000088
   Transformation
             Lambda1    1.000000                Fixed
             Lambda2    0.050000                Fixed
-----------------------------------------------------------------------
Final ln likelihood: -382.959654
====================
      JOINT TEST
====================
   HO ln likelihood susceptibilities free, no truncation   -382.959654
   H1 ln likelihood susceptibilities equal, no truncation  -383.779624

   2 * |H0 - H1|                                            1.639942
   Degrees of freedom                                              5
   P-value                                                  0.896377
====================
      JOINT TEST
====================
   HO ln likelihood susceptibilities free, using truncation   -382.959654
   H1 ln likelihood susceptibilities equal, using truncation  -383.779624

   2 * |H0 - H1|                                              1.639942
   Degrees of freedom                                                5
   P-value                                                    0.896377
=========================================================================
      VARIANCE-COVARIANCE MATRIX susceptibilities equal, no truncation
=========================================================================
------------------------------------------------------------------------------------------ ... ------------
                      Mean intercept  Variance intercept  cov1       cov2       Class 0     ...  Class 5
------------------------------------------------------------------------------------------ ... ------------
   Mean intercept       0.033182        0.057390          -0.003310   0.032536   0.010836   ...   0.010836
   Variance intercept   0.057390        2.439538          -0.000076   1.257227   0.041093   ...   0.041093
   cov1                -0.003310       -0.000076           0.029612   0.006981  -0.000758   ...  -0.000758
   cov2                 0.032536        1.257227           0.006981   1.443502   0.021532   ...   0.021532
   Class 0              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 1              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 2              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 3              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 4              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 5              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
=========================================================================
      VARIANCE-COVARIANCE MATRIX susceptibilities free, no truncation
=========================================================================
Error: Matrix is not available.
=========================================================================
      VARIANCE-COVARIANCE MATRIX susceptibilities equal, using truncation
=========================================================================
------------------------------------------------------------------------------------------ ... ------------
                      Mean intercept  Variance intercept  cov1       cov2       Class 0     ...  Class 5
------------------------------------------------------------------------------------------ ... ------------
   Mean intercept       0.033182        0.057390          -0.003310   0.032536   0.010836   ...   0.010836
   Variance intercept   0.057390        2.439538          -0.000076   1.257227   0.041093   ...   0.041093
   cov1                -0.003310       -0.000076           0.029612   0.006981  -0.000758   ...  -0.000758
   cov2                 0.032536        1.257227           0.006981   1.443502   0.021532   ...   0.021532
   Class 0              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 1              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 2              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 3              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 4              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
   Class 5              0.010836        0.041093          -0.000758   0.021532   0.026250   ...   0.026250
=========================================================================
      VARIANCE-COVARIANCE MATRIX susceptibilities free, using truncation
=========================================================================
Error: Matrix is not available.
```

Chapter 17

DECIPHER

DECIPHER obtains maximum likelihood estimates of population haplotype frequencies for autosomal or X-linked markers, and determines all possible diplotypes and the most likely diplotypes for each individual. Genotypes of other pedigree members can be used to infer phase for ambiguous individuals, which improves the population haplotype frequency estimates over those obtained using unrelated individuals only. Haplotype frequencies can be estimated separately for different populations that are specified by the user. A likelihood ratio test and a permutation test are provided to compare haplotype frequency distributions for dichotomous phenotypes.

17.1 Limitations

Genotypes of other pedigree members can be used to infer phase for ambiguous individuals only for non-recombinant regions (i.e., no recombination is observed in the pedigree between those markers). Memory constraints may be encountered in situations where a large fraction of markers are missing, or when a large number of markers (more than 25) are haplotyped. Finally, markers in the haplotyping region must be codominant, and family information may not be used with X-linked markers. The current version of DECIPHER limits the number of individuals per pool, k, to at most 1; this limitation will eventually be removed in a future version of the program.

17.2 Theory

Maximum likelihood estimates of haplotype frequencies can be obtained from pooled DNA using a form of the expectation-maximization (EM) algorithm developed expressly for that purpose (Quade et al. 2005; Ito et al. 2003; Wang, Kidd, and Zhao 2003). The approach incorporates a variety of data types, including unrelated individuals, sets of related individuals (i.e., families), and pooled samples, or combinations of these data types. The key modification was the recognition that each of the other types of data can be considered a special case of pooled data. For example, unrelated individuals can be considered a pool with one individual. Groups of founders in a pedigree can be considered a pool of f individuals, where f is the number of founders. To allow combinations of the data types and to allow variation in the number of founders per pedigree, we modified the EM algorithm to allow different numbers of individuals in each unit.

To estimate population haplotype frequencies, a random set of unrelated individuals or pedigrees must be used. For pedigree data, the user can specify a single representative from each pedigree. The family representative is either selected by the user or, if no individual is indicated for a particular pedigree, the program will randomly select one individual out of those individuals in the pedigree with the most marker genotypes (i.e., we are assuming the genotypes are missing at random).

The form of the EM algorithm for pooled data is as follows. Suppose we are given *n* pools and each pool contains *k* individuals. The total number of markers is *m*. In this description, we primarily focus on single nucleotide polymorphisms (SNPs) which are diallelic markers with alleles encoded as 0 or 1; however, DECIPHER allows more than two alleles per locus. For each pool, at each marker position, we are given the number of 0s and the number of 1s. The summation of these two numbers is 2*k* since each individual provides 2 alleles and there are *k* individuals in each pool. The input data can be represented by a nonnegative integer matrix ***M*** of size *n* x *m*, where the *i*-th row, $M_{i.}$, represents one pool and the *j*-th column, $M_{.j}$, represents one SNP, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Each entry, M_{ij}, is an integer representing the number of allele 1s in pool *i* at SNP *j*. The value of each entry thus should be $\leq 2k$ and ≥ 0. For *m* markers, there are total of

$T = 2^m$ possible haplotypes. Let h_t denote the *t*-th haplotype and let f_t denote its population frequency for $0 \leq t \leq T$. Let $H = \{h_t : 0 \leq t \leq T\}$ and $F = \{f_t : 0 \leq t \leq T\}$ be the set of all haplotypes and the set of haplotype frequencies, respectively. For a given pool $M_{i.}$, let H_i denote the set of all possible haplotype assignments for $M_{i.}$, i.e., each element Δ of H_i contains 2*k* haplotypes for the *k* individuals in pool *i*. Under the assumption of Hardy-Weinberg equilibrium and random mating, and assuming that all the individuals are independent, the likelihood for the given data can be expressed as

$$P(M, F) = \prod_{i=1}^{n} \sum_{\Delta \in H_i} P(\Delta) \tag{17.1}$$

The standard EM algorithm starts with an initial assignment of the haplotype frequencies for *F*. During the E step, the expected number of each haplotype is calculated under the assumption that the haplotype frequencies are known, and during the M step the haplotype frequencies are updated according to the haplotype counts calculated in the previous E step. The two steps are iterated until convergence, defined as the minimum difference between haplotype frequencies in successive iterations being less than a small number, ε, which is specified by the user. To ensure that a global maximum is reached rather than a local maximum, the user can specify the number of starting points that will be used. DECIPHER will obtain maximum likelihood estimates for each of these randomly selected starting points, and the set of estimates corresponding to the maximum likelihood will be displayed. We have modified this algorithm so that the value of *k* can differ for each pool. Note that for a pool that consists of a single male with X-linked data, *k* equals 1/2; however, in this instance the haplotype is always known with certainty.

For pedigree data, we use descent graphs to identify compatible haplotypes for a particular individual in the pedigree consistent with the observed data in the pedigree. We assume all markers are in a region with no observed recombination within the pedigrees. Using the method of descent graphs described by (Sobel and Lange 1996), we can identify all possible allele states at each locus for each individual. A complete list of all possible haplotype states for each individual can then be obtained by taking the Cartesian product of the possible allele states at each locus. The possible founder haplotypes are linked through the descent graphs, such that sets of founder haplotypes that are simultaneously consistent with the observed data can be obtained. These sets of possible haplotypes, H_i, are then used in equation 17.1 above.

There are several types of information that can be obtained from this procedure. First, haplotype frequencies can be estimated for sets of individuals. The user has the option of partitioning the individuals into subpopulations (e.g., case-control status, ethnic groups, etc) and obtaining haplotype frequencies separately for each subpopulation. Second, we can obtain a list of all possible non-recombinant diplotypes for each individual or pool (with the constraint of < 30 markers). In addition, we can obtain the posterior probability of each of these possibilities. Third, we can obtain a list of the most likely pairs of haplotypes (i.e., diplotypes) for each individual, together with the relative likelihood of each, based on population data. The list of haplotypes or diplotypes can be quite large, particularly when there is a large number of markers and/or alleles. Therefore, an option is provided to specify a threshold, such that only haplotypes (diplotypes) with a frequency (posterior probability) greater than this threshold will be displayed. In the case where the most likely diplotypes are requested, more than one diplotype will be returned if they have the same (maximum) posterior probability.

A likelihood ratio test is available to compare the distribution of haplotypes in populations (e.g., cases versus controls). Assume we have *N* groups, and we have estimated haplotype frequencies separately for each group and for the whole sample combined. Assume there are h_j haplotypes with frequency p_{ij} for haplotype *i* in group *j*. For the likelihood ratio test, the null hypothesis is $H_0 : p_{i1} = p_{i2} = ... = p_{in}$, versus the alternative hypothesis, $H_A : p_{ij} \neq p_{ik}$ for at least one haplotype *i*, and at least one pair of groups *j* and *k*. The likelihood is maximized under these two conditions (i.e., forcing p_{ij} to be the same for all *j* versus allowing them to be different). The likelihood ratio (LR) is then formed, and -2ln(LR) asymptotically follows a chi-square distribution with $(N-1)(h_T-1)$ degrees of freedom, where h_T is the number of haplotypes for the whole sample. This asymptotic distribution is conservative when there are rare haplotypes, and is not recommended under those circumstances. Therefore, we also provide a method for obtaining an empirical p-value for the LR test statistic. This is obtained by sampling permutations of the category assignment (e.g., case-control status), and recomputing the LR test statistic for each permutation. The empirical p-value is determined from the sample permutations as the number of permutations where the LR test statistic exceeds the observed LR test statistic, divided by the total number of permutations.

17.3 Program Input

File Type	Description
Parameter file	Specifies the parameters and options with which to perform a particular analysis.
Pedigree data file	Contains delimited records for each individual, including fields for identifiers, sex, parents, trait and marker data.
Genome description file	Contains a description of the linked marker regions, including distances between consecutive markers.

Note: A marker locus description file is not accepted by DECIPHER. Missing allele and allele delimiter symbols may be specified using the marker sub-block in the parameter file. See section 2.3.3 for details.

17.3.1 Parameter File

The following syntax table specifies the permissible parameter and attribute settings for the main DECIPHER parameter.

parameter [, attribute]	**Explanation**
decipher	Starts a DECIPHER analysis block. Value Range: N/A Default Value: N/A Required: Yes Applicable Notes: None
, out	Specifies the root name to be used for output files. Output file names will be formed by concatenating the root name and an appropriate extension. Value Range: Character string representing a valid file name Default Value: decipher.out Required: No Applicable Notes: None

17.3.2 The `decipher` Parameter Block

The following syntax table specifies the permissible parameter and attribute settings for the `decipher` parameter block.

`parameter [, attribute]`	**Explanation**
`title`	Specifies the title of the analysis Value Range: Character string Default Value: Analysis 1 Required: No Applicable Notes: None
`region`	Specifies the name of the chromosomal region to be analyzed. Value Range: Character string naming a region listed in the genome description file. Default Value: None Required: Yes Applicable Notes: 1
`epsilon`	Specifies the minimum difference between haplotype frequencies in successive iterations as a convergence criterion for the EM algorithm. Value Range: (0, 1) Default Value: 0.00001 Required: No Applicable Notes: None
`starting_points`	Specifies number of randomly chosen starting points against which the EM algorithm is to run. Value Range: {1, 2, 3, ...} Default Value: 10 Required: No Applicable Notes: None
`dump`	Enables option to write haplotype frequencies (and their log-likelihoods) for each set of EM algorithm starting points to an output file. Value Range: {**true**, **false**} Default Value: **false** Required: No Applicable Notes: 2, 3

, cutoff	Specifies minimum haplotype frequency threshold value for display. If none of the estimated haplotype frequencies meet or exceed the specified value, then the haplotype with the greatest estimated frequency is displayed. Value Range: [0, 1] Default Value: 0.001 Required: No Applicable Notes: None
data	Starts a sub-block to specify 1. how to treat relatedness of individuals, 2. individuals to represent families, and 3. individuals to represent subpopulations. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
tasks	Starts a sub-block to specify analysis tasks to be performed. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

Notes

1. Used for establishing marker order (no marker order is implied by parameter or pedigree files). Distances between the markers are ignored by DECIPHER.
2. Applicable only if `pop_freq` or `most_likely_diplotypes` in tasks sub-block is set to **true**.
3. Starting points shown in the dump file output are generated by choosing random phase probabilities for each pool and then calculating the haplotype frequencies.

17.3.3 The `data` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `data` sub-block.

`parameter [, attribute]`	**Explanation**
`related`	Specifies whether familial relationships are to be considered in determining possible diplotypes. If this parameter is set to **false**, individuals will be assumed to be independent and they will all be used in the estimation of haplotype frequencies. If set to **true**, one genotyped person per constituent pedigree will be used. Familial information will be considered in determining possible diplotypes. Unconnected individuals will be treated as if the parameter where set to **false**.
	Value Range: {**true**, **false**}
	Default Value: true
	Required: No
	Applicable Notes: 1
`family_rep`	Variable used to specify one genotyped individual per constituent pedigree when `related` equals **true**.
	Value Range: Character string representing the name of a trait, phenotype, covariate or string field listed in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: 2, 8
`, family_rep_value`	Specifies the value of the `family_rep` variable that identifies a genotyped individual for haplotype analysis.
	Value Range: Character string representing the (literal) value to be matched in the designated `family_rep` field.
	Default Value: None
	Required: Yes, if `family_rep` is specified
	Applicable Notes: 3

`partition`	Starts a sub-block for specifying subpopulations of individuals from the original data set.	
	Value Range	Character string representing the name of a trait, phenotype, covariate or string field listed in the pedigree data file. The named field will be used as the basis of classification for the created partition (see notes).
	Default Value	None
	Required	No
	Applicable Notes	4, 5, 6, 7, 8

Notes:

1. If this option is selected and a Mendelian inconsistency is detected in a constituent pedigree at a particular locus, all members of the constituent pedigree are treated as if they had missing values for that locus.
2. If no variable is specified, the program will arbitrarily pick a genotyped individual in each constituent pedigree to be used in haplotype calculations.
3. If more than one individual in a pedigree has this value, the one with the most genotyped loci in the haplotype region is chosen. In the case of a tie, the selection is made at random.
4. This sub-block may appear no more than twice per analysis block and each partition sub-block in an analysis block must have a unique value.
5. If this sub-block is not specified, all individuals will be treated as a single population.
6. All individuals having the same value for this variable belong to the same subpopulation.
7. The order in which the partitions are listed is significant. See note #3 of the `tasks` sub-block for details.
8. The same trait, phenotype or covariate may not be used as a value for both the `family_rep` and `partition` parameters.

17.3.4 The `partition` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `partition` sub-block.

`parameter [, attribute]`	**Explanation**	
`sub_pop`	Specifies name of subpopulation.	
	Value Range	Character string
	Default Value	None
	Required	No
	Applicable Notes	1, 2
`, sub_pop_value`	Specifies value of partition variable common to all individuals in this subpopulation.	
	Value Range	Character string representing the (literal) value to be matched in the designated `sub_pop` field.
	Default Value	None
	Required	No
	Applicable Notes	3, 4, 5

Notes:

1. If not specified, subpopulation name is the same as `sub_pop_value`.
2. This parameter may be repeated as needed but `sub_pop` and `sub_pop_value` must be unique within a partition.
3. Required if `sub_pop` is specified.
4. If no valid values are specified for the `sub_pop_value` option, then every distinct value of the partition variable found in the pedigree data file (except the missing value), will designate a subpopulation.
5. Missing value code may not be specified as a `sub_pop_value`.

17.3.5 The `tasks` Sub-Block

The following syntax table specifies the permissible parameter and attribute settings for the `tasks` sub-block.

parameter [, attribute]	**Explanation**
`pop_freq`	Specifies option to estimate population haplotype frequencies. Value Range: {**true**, **false**} Default Value: **true** Required: No Applicable Notes: None
`, cutoff`	Specifies minimum haplotype frequency threshold value for display. If none of the estimated haplotype frequencies meet or exceed the specified value, then the haplotype with the greatest estimated frequency is displayed. Value Range: [0, 1] Default Value: 0.001 Required: No Applicable Notes: 1
`all_possible_diplotypes_table`	Specifies option to display diplotypes for each individual in tabular form. Value Range: {**true**, **false**} Default Value: **false** Required: No Applicable Notes: None
`most_likely_diplotypes`	Specifies option to display the most likely diplotypes for each individual. Value Range: {**true**, **false**} Default Value: **false** Required: No Applicable Notes: None
`, cutoff`	Specifies minimum haplotype frequency threshold value for display. If none of the estimated haplotype frequencies meet or exceed the specified value, then the haplotype with the greatest estimated frequency is displayed. Value Range: [0, 1] Default Value: 0.05 Required: No Applicable Notes: 2
`likelihood_ratio_test`	Specifies option to perform likelihood ratio test. Value Range: {**true**, **false**} Default Value: **false** Required: No Applicable Notes: 3

`compute_empirical_pvalue`	Specifies option to estimate p-values by permutation methods. Value Range: {**true**, **false**} Default Value: **false** Required: No Applicable Notes: None
`, permutations`	Specifies an exact number of permutations to be performed. Use of this option effectively overrides all of the following attributes. Value Range: {1, 2, 3, ...} Default Value: None Required: No Applicable Notes: None
`, max_permutations`	Specifies the maximum number of permutations that should be performed. Value Range: {1, 2, 3, ...} Default Value: 10,000 Required: No Applicable Notes: None
`, width`	Specifies the relative precision of the empirical p-value. For example, if `width` = 0.2, then p-values will be estimated to be within 20% of their true value with a given confidence level. This value is used to choose the number of permutations necessary. Note that the number of permutations required varies quadratically with the inverse of the width. Value Range: [0, 1] Default Value: 0.2 Required: No Applicable Notes: None
`, confidence`	Specifies the confidence with which an empirical p-value is required to be within a relative interval (i.e., the width) of its true value. Value Range: [0, 1] Default Value: 0.95 Required: No Applicable Notes: None

Notes:

1. To display all *haplotype* frequency estimates, specify a `cutoff` of 0.
2. To display all *diplotype* probabilities, specify a `cutoff` of 0.
3. If two `partition` sub-blocks are specified, the first partition listed is *Partition 1*, and the second partition listed is *Partition 2*. Likelihood ratio tests are performed across the subpopulations of Partition 1 for each of the subpopulations in Partition 2. At least two subpopulations must be specified in a partition sub-block to do this test.

The following are all valid decipher analysis blocks:

```
decipher, region = “chrom 1” {
}
decipher, region = Chr12 { # Quotes not required since the region name does not contain spaces
   related = false         # Do not use family information in determining
                           # possible haplotypes.
}
decipher, out="run 1" {
   title = "1st run"
   region = "chrom 14"
   epsilon = .0001       # End EM algorithm when differences in frequency estimates for
                         # successive iterations are less than .0001 for all haplotypes.
   starting_points = 3   # Run EM algorithm 3 times with a different set of
                         # starting points each time.
   data {
      related = true
      family_rep = "T1", # Values for this trait designate a
                         # genotyped individual in each family whose haplotypes
                         # are to be determined.

         family_rep_value = 1  # Haplotypes to be determined for genotyped individuals
                               # whose value for trait, T1, equals 1.
      partition = "T2" {       # Values for this trait will determine membership
                               # in subpopulations.
         sub_pop = "pop1", sub_pop_value = 1  # If individual has value of 1 for T2, he
                                              # belongs to pop1.

         sub_pop = "pop2", sub_pop_value = 2  # if individual has value of 2 for T2, he
                                              # belongs to pop2.
      }
    }
  tasks {
     pop_freq=true, cutoff = .1  # Show only haplotype frequency estimates greater
                                 # than .1.
     likelihood_ratio_test = true
     compute_empirical_pvalue = true, permutations = 1000
  }
}
```

17.4 Program Execution

DECIPHER is run via a command line interface on the supported UNIX and Windows platforms. This requires the S.A.G.E. programs to be properly installed and in the current execution path. Input files are specified on the command line and all output files are created in the current working directory.

Running DECIPHER from the command prompt with no arguments, or the wrong number of arguments, will result in the program displaying its usage statement, which lists the input files names required on the command line.

```
>decipher
DECIPHER Output -- 13 Jul 2005 10:43:16 -- [S.A.G.E. v5.0.3; bld 27 Jun 2005]
COPYRIGHT (C) 2005 CASE WESTERN RESERVE UNIVERSITY
usage: ./decipher <parameters> <pedigree> <map>
Command line parameters:
  parameters   - Parameter File
  pedigree     - Pedigree Data File
  map          - Genome Description File
```

A typical run of DECIPHER may look like the following:

```
>decipher par ped mld gen
DECIPHER Output -- 13 Jul 2005 10:52:36 -- [S.A.G.E. v5.0.3; bld 27 Jun 2005]
COPYRIGHT (C) 2005 CASE WESTERN RESERVE UNIVERSITY
Reading Parameter File.....................done.
Reading Pedigree File.....................
              from ped...................done.
Sorting Pedigrees.........................done.
Reading Genome Description File...........done.
Parsing DECIPHER analyses ...
Parsing new analysis block ...
Parsing of Analysis 1 complete.  Analysis valid.
-------------------------------------------------------------------------------
  Performing analysis: Analysis 1
  =============================================================================
    Total population.
      Determining possible diplotypes ............................. done.
      Maximizing likelihood ....................................... done.
  =============================================================================
```

17.5 Program Output

DECIPHER produces four types of output files that contain results and diagnostic information:

File Name	File Type	Description
decipher.inf	DECIPHER Information output file	Contains informational diagnostic messages, warnings and program errors. No analysis results are stored in this file.
decipher.sum	DECIPHER summary output file	Contains population haplotype frequency estimates.
decipher.det	DECIPHER detailed output file	Contains possible diplotypes and most likely diplotypes for individuals.
decipher.dmp	DECIPHER data dump file	Contains haplotype frequency estimates and ln likelihoods for each set of starting points for which the EM algorithm is run.

17.5.1 Information Output File

The Information Output File contains a variety of useful information, including:

- Information on fields read from the Pedigree Data File. These tables, which provide information about what the program has read from the Pedigree Data File, are included with all programs in S.A.G.E. and are very useful for debugging most common errors caused when reading the pedigree data. When first analyzing new data, it is recommended that these tables be checked carefully to make sure pedigree data are being correctly read.
- Information, warning and error messages generated throughout the program. It is recommended that this file be checked for warning and error messages before examining the results of any run of the program. The program attempts to correct many common errors and this sometimes means analyses are not as expected. The file "decipher.inf" should be checked for errors and diagnostic information after each run of the program.

17.5.2 Summary Output File

Contains results pertaining to whole data set specifically haplotype frequency estimates, likelihood ratio test results and empirical p-values.

17.5.3 Detail Output File

Contains results on an individual basis, specifically possible and most likely diplotypes.

17.6 Example Output Files

17.6.1 Example Summary Output File

```
==========
Analysis 1
==========
Options Selected
================
Haplotype region                          one
EM algorithm convergence criterion        1e-05
Number of EM algorithm starting states    10
  Dump                                    no
Subpopulations specified                  no
Use family information                    yes
  Family representatives specified        yes
Estimate haplotype frequencies            yes
  Cutoff                                  1e-06
List all possible diplotypes              no
Show all possible diplotypes table        no
List most likely diplotypes               no
Do likelihood ratio test                  no
Compute empirical p-value                 no
Results
=======
Markers in order:
  M1 M2 M3 M4 M5 M6 M7 M8 M9
Missing allele symbol (first marker):
  '?'
                         Haplotype Frequency Estimates
Note:  Haplotypes listed have estimated frequencies greater than or equal to
       the cutoff or have the greatest frequency estimate.
Haplotype                  Frequency
---------                  ---------
1-1-2-2-2-2-1-1-1          0.190474
1-1-2-2-1-1-2-2-1          0.187969
2-1-2-2-1-2-1-1-1          0.186462
2-2-2-1-1-2-2-1-2          0.172941
1-1-1-1-1-1-2-1-1          0.163969
1-2-2-2-1-1-2-2-1          0.00303146
1-2-1-1-1-1-1-1-1          0.00301807
2-2-1-2-1-1-2-1-2          0.00300129
1-1-2-2-1-2-2-1-2          0.00300000
1-1-2-1-2-2-2-2-1          0.00300000
1-1-2-1-2-1-1-2-2          0.00300000
.
.
.
1-1-1-1-1-2-1-2-1          0.00100000
2-2-2-2-2-2-1-2-1          0.000992059
1-2-2-1-1-1-1-2-1          0.000753648
1-2-1-1-1-2-1-1-2          0.000702381
1-2-1-2-1-2-1-1-2          0.000297619
1-2-2-1-1-2-1-2-1          0.000246352
1-2-2-2-1-1-2-2-2          7.94147e-06
2-2-2-1-2-2-1-2-2          1.28784e-06
1-2-2-1-2-2-1-1-2          1.27373e-06
                           ---------
Total                      0.999999
Ln likelihood    -1860.13
```

17.6.2 Example Detail Output File

```
==========
Analysis 1
==========
Markers in order:
  M1 M2
Missing allele symbol (first marker):
  '?'
                     Most Likely Diplotypes
Note:  Diplotypes listed have estimated probabilities greater than or equal
       to the cutoff or have the greatest probability estimate.
Pedigree     Member     Diplotype     Probability
--------     ------     ---------     -----------
1            1          a-A  a-a      1.00000
1            10         A-a  a-a      1.00000
1            2          A-A  A-a      1.00000
1            3          A-a  A-a      1.00000
1            4          A-a  a-A      0.934722
                        A-A  a-a      0.0652776
1            5          A-a  a-A      0.934722
                        A-A  a-a      0.0652776
1            6          a-A  a-A      1.00000
1            7          A-a  a-A      0.934722
                        A-A  a-a      0.0652776
1            8          A-a  a-A      0.934722
                        A-A  a-a      0.0652776
1            9          A-a  a-a      1.00000
                      All Possible Diplotypes
                     pedigree:member
           1:1  1:10 1:2  1:3  1:4  1:5  1:6  1:7  1:8  1:9
A-A  A-a   -    -    x    -    -    -    -    -    -    -
A-A  a-a   -    -    -    -    x    x    -    x    x    -
A-a  A-a   -    -    -    x    -    -    -    -    -    -
A-a  a-A   -    -    -    -    x    x    -    x    x    -
A-a  a-a   -    x    -    -    -    -    -    -    -    x
a-A  A-A   -    -    -    -    -    -    x    -    -    -
a-A  a-a   x    -    -    -    -    -    -    -    -    -
```

17.6.3 Example Dump File

```
DECIPHER Output -- 13 Jul 2005 11:14:37 -- [S.A.G.E. v5.0.3; bld 27 Jun 2005]
COPYRIGHT (C) 2005 CASE WESTERN RESERVE UNIVERSITY
Maximizing total population...
Note:  Haplotypes listed have estimated frequencies greater than or equal to
       the cutoff or have the greatest frequency estimate.
run 1
start point
Haplotype          Frequency
---------          ---------
A-a                0.401786
a-A                0.301786
a-a                0.198214
A-A                0.0982143
                   ---------
Total              1.000000
end point
Haplotype          Frequency
---------          ---------
A-a                0.436945
a-A                0.336945
a-a                0.163055
A-A                0.0630555
                   ---------
Total              1.000000
Ln likelihood    -17.45593957
run 2
start point
Haplotype          Frequency
---------          ---------
A-a                0.370000
a-A                0.270000
a-a                0.230000
A-A                0.130000
                   ---------
Total              1.000000
end point
Haplotype          Frequency
---------          ---------
A-a                0.436945
a-A                0.336945
a-a                0.163055
A-A                0.0630555
                   ---------
Total              1.000000
Ln likelihood    -17.45593957
.
.
.
run 9
start point
Haplotype          Frequency
---------          ---------
A-a                0.313158
a-a                0.286842
a-A                0.213158
```

```
A-A                0.186842
                   ---------
Total              1.000000
end point
Haplotype          Frequency
---------          ---------
A-a                0.436944
a-A                0.336944
a-a                0.163056
A-A                0.0630555
                   ---------
Total              1.000000
Ln likelihood    -17.45593957
run 10
start point
Haplotype          Frequency
---------          ---------
A-a                0.344915
a-a                0.255085
a-A                0.244915
A-A                0.155085
                   ---------
Total              1.000000
end point
Haplotype          Frequency
---------          ---------
A-a                0.436944
a-A                0.336944
a-a                0.163056
A-A                0.0630557
                   ---------
Total              1.000000
Ln likelihood    -17.45593957
```

Chapter 18

DESPAIR

DESPAIR is a program to help in designing linkage studies for searching the whole autosomal genome. Originally created for a study comprising affected pairs of relatives of a particular type, the latest version of DESPAIR has been modified to further incorporate discordant relative pairs into the study. The program can be used to determine, for specified power and significance level, the optimal two-stage study design – i.e., how many pairs of relatives should be studied, how many equally spaced markers should be used initially, and what criterion should be used to specify the markers around which further searching should be done. Alternatively, the program will calculate either the number of relative pairs required for a given number of first-stage markers, or the number of markers required for a given number of relative pairs.

18.1 Limitations

18.1.1 Theoretical Limitations

The method used assumes that independent pairs of relatives of a single particular type (full sibling, half-sibling, grandparent-grandchild, avuncular, or first cousin) are being sampled. Only three levels of interference are considered, corresponding to Haldane's mapping function (no interference), Kosambi's mapping function (moderate interference), and Morgan's linear mapping function (extreme interference). The spacing between markers is not allowed to be less than one tenth of a centimorgan, nor as much as one morgan, and markers are assumed to be in linkage equilibrium. Two test statistics are allowed for in the cases of sibling pairs, but only one (that based on the mean test) is implemented for designs that use both affected and discordant pairs.

18.2 Theory

It is well understood that linkage of a putative disease locus to a polymorphic marker can be conducted through a study design of affected pairs of relatives, and this is usually the most powerful sampling strategy for binary traits (Blackwelder and Elston, 1985; Risch, 1990). However, recent research shows that, under certain situations, using discordant relative pairs can be as powerful as, or even more powerful than, using affected relative pairs. Moreover, combining discordant with

affected relative pairs provides a more valid and reasonable study from both a theoretical and practical point of view (Guo and Elston, 2000). Specifically, linkage can be studied by typing pairs of relatives and examining the proportions of the pairs sharing 0, 1, or 2 alleles identical by descent (IBD) at the marker locus. The test for linkage in DESPAIR is based on either the proportion of pairs sharing 0 alleles IBD or the mean proportion of marker alleles shared IBD, which depend on the type of relative pair.

Denote the expected values of either of these proportions under the null hypothesis of free linkage π_0. If there is linkage, the expected values are $\pi_0 + \delta_c$ and $\pi_0 - \delta_d$, corresponding to a design using affected relative pairs alone and a design using discordant relative pairs alone, respectively; δ_c and δ_d are the expected deviations respectively for affected pairs and discordant pairs due to linkage. Both these measures depend not only on the type of relative pair, but also on the recombination fraction θ between the marker and disease loci. In addition, δ_c depends on the relative recurrence risk of disease, due to the disease locus, to first degree relatives of affected persons:

$$\lambda = \frac{Pr(\text{first degree relative of affected person is affected})}{Pr(\text{random member of population is affected})}$$

and δ_d depends on the corresponding relative non-recurrence risk ratio for an affected-unaffected first degree relative pair:

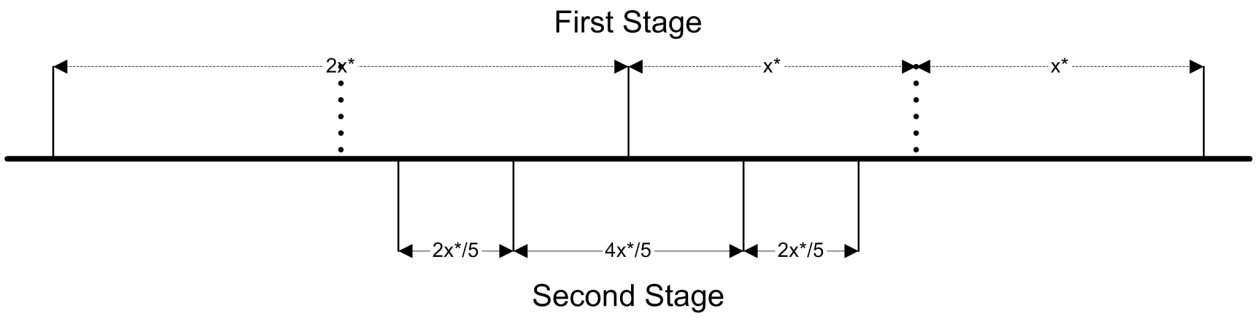
$$\lambda^- = \frac{Pr(\text{first degree relative of affected person is unaffected})}{Pr(\text{random member of population is unaffected})}.$$

Each of these relative risks, often called risk ratios, can be to either a parent/offspring (λ_o,λ_o^-) or to a full sibling (λ_s,λ_s^-).

If several disease loci act multiplicatively, the relative risk is the product of λ's, one for each locus. For a study design that combines affected relative pairs with discordant relative pairs, the test statistic is based on the notion that, in the presence of linkage, affected relative pairs are expected to share a larger proportion of marker alleles IBD, whereas discordant relative pairs are expected to share a smaller proportion of alleles IBD. The difference in the proportion of alleles shared IBD between affected pairs and discordant pairs is quantified by Δ, a weighted difference in the deviations of the mean proportions from π_0. Δ equals zero under the null hypothesis of no linkage, and is greater than zero when linkage is present. The values of Δ can be expressed as a function of θ, λ, λ^-, and the ratio (r_p) of the number of affected relative pairs to the number of discordant relative pairs that are sampled. Values of $\pi_0 + \delta_c$ were given by Risch (1990), and values of $\pi_0 - \delta_d$, and Δ were given by Guo and Elston (2000), for five relative pairs: full sibling, half sibling, avuncular, grandparent-grandchild, and first cousin.

The test based on the proportion sharing 0 alleles IBD and the mean test give identical results except in the case of full sib pairs. The test based on the proportion sharing 0 alleles IBD is not implemented for designs using both concordant affected and discordant full sib pairs.

Assume that at a first stage, *m* fully informative markers, equally spaced along an autosomal genome *M* morgans long, are determined on *n* pairs of relatives of a particular type. For each marker, a one-sided test is performed at the α^* significance level to decide whether the sample proportion of alleles shared IBD deviates significantly from π_0, suggesting linkage. Around each marker suggesting linkage at the first stage, a further 2*k* fully informative markers are tested for linkage at a second stage, assuming that these are placed, *k* on either side of the first stage marker, to span in an optimal manner the interval of interest suggested by the significant first-stage marker (see Figure 18.1).



Genetic map segment (thick horizontal line) showing the placement of three first-stage markers (solid vertical bars above the line) spaced 2x* = M/m Morgans apart, and 2k = 4 second-stage markers (vertical bars below the line) spaced 2x = 2x*/(2k+1) Morgans apart. Dotted vertical bars indicate midway points between the first-stage markers.

Figure 18.1: Stage-1 and Stage-2 Marker Placement

Assume that the study is designed to have power 1 - β of detecting a disease locus with relative risk ratio λ at a significance level α at the second stage, and that there are actually *d* such disease loci present. Finally, assume that the cost of recruiting a person into the study is *R* times the cost of determining one marker on one person. Under these assumptions, if at most one first stage marker is linked to any disease locus, the expected cost of the study is proportional to

$$2n\{R + m + 2k[\alpha^* m + (1 - \beta)d]\} \tag{18.1}$$

However, because there may be more than one first-stage marker linked to the disease locus, the total expected cost is more appropriately reflected by

$$C = 2n\{R + m + 2k[\alpha^*(m - \sum_{i=1}^{d} l_i) + \sum_{i=1}^{d}\sum_{j=1}^{l_i}(1 - \beta_{ij})]\} \tag{18.2}$$

where l_i is the number of first stage markers linked to disease locus *i*, and $1 - \beta_{ij}$ is the probability that 2*k* second stage markers are typed around marker *j* that is linked to disease locus *i* (Ziegler et al. 2001). In this revised version of DESPAIR, which implements cost function (18.2), users have the option to input a maximum distance (g) between any disease locus and a "linked" marker. Then significant results obtained within *g* morgans from any disease locus are considered to be successes, and any outside that range are considered to be false positives. By making the distance *g* small in comparison to the distance between first stage markers, for a large number of markers cost function (18.2) approaches cost function (18.1), which was the function used in the original version of DESPAIR.

Given α, β , λ, *R*, *d*, *g*, *M*, the type of relative pairs, and the type of data (affected relative pairs, discordant relative pairs, or both discordant and affected pairs: for the latter two cases, λ^- must also be specified; and for the last one case, the ratio (r_p) of the number of affected to the number

of discordant relative pairs to be sampled must also be specified), DESPAIR finds the values of *m*, *n*, and α^* that minimize this expected cost for different mapping functions (linear, Kosambi's, and Haldane's), and for values of *k* from 0 (a one-stage design) to a specified maximum value of *k*, subject to the limitation *M* < *m* < 1000M (i.e., the markers must be spaced less than one morgan apart, and must be no closer than one tenth of a centimorgan apart). There is an option (c) to include the cost of screening the population to find the desired sample (the cost of screening is taken to be the same as the cost of recruiting), in which case the user must also enter the proportion of the screened population (r_s) that becomes the final sample.

It is assumed that *n* is large enough, in determining the test criterion corresponding to α^* and β, that the distribution of the proportion of pairs sharing 0 alleles IBD or the mean proportion of marker alleles shared IBD is normally distributed. However, in the case of α, which is typically much closer to zero, there is the option of using either this same approximation assumption (the approximate method), or exact binomial distribution probabilities (the exact method, not implemented for the case where the sample includes both affected and discordant pairs).

To allow for less than fully informative markers, a value of the polymorphism information content (PIC), which measures the markers informativeness (assumed to be the same for all markers), can be specified. This is converted by the program to the corresponding type-of-pair-specific LIC value (Guo and Elston, 1999; Guo et al. 2002). Similarly, a fraction *h*, heterogeneity, can be specified that represents the proportion of the sample pairs affected due to causes other than segregation at the linked locus (in this case one would typically specify a large value for λ and/or a small value of λ^-).

Further details of the method are given in the references.

18.3 Running the Program

DESPAIR can be run by clicking on the DESPAIR GUI link on the S.A.G.E. website

http://darwin.cwru.edu/despair/

and inputting one or more sets of parameters for which the sample size (numbers of affected sib pairs and/or number of markers) is desired. The parameters may be specified as follows:

parameter	Explanation	
relative_pair_type	Specifies relative pair type.	
	Value Range	**S** (full siblings), **G** (grand-parental), **A** (avuncular), **H** (half siblings), **C** (first cousins)
	Default Value	S
	Required	Yes
	Applicable Notes	None
concordance_type	Specifies the phenotypic concordance status (adb) of the observations.	
	Value Range	**A** (affected relative pairs) **D** (discordant relative pairs) **B** (both)
	Default Value	A
	Required	Yes
	Applicable Notes	None
method	Specifies the analysis method to be used.	
	Value Range	**A** (approximate) **E** (exact)
	Default Value	A
	Required	Yes
	Applicable Notes	1
significance	Specifies the statistical significance level α.	
	Value Range	(0, 1)
	Default Value	0.000101
	Required	Yes
	Applicable Notes	2
power	Specifies the statistical power level $1 - \beta$.	
	Value Range	(α, 1)
	Default Value	None
	Required	Yes
	Applicable Notes	None

`test_statistic`	Specifies the type of test statistic to be employed (mp).	
	Value Range	**M** (mean statistic) **P** (proportion statistic)
	Default Value	**M**
	Required	Yes
	Applicable Notes	3
`offspr_recurrence_risk`	Specifies λ_o, the locus-specific relative recurrence risk ratio of disease for an offspring of the affected person.	
	Value Range	(1, $+\infty$) for ARPs (0, 1) for DRPs
	Default Value	None
	Required	Yes
	Applicable Notes	3
`offspr_nonrecurrence_risk`	Specifies λ_o^-, the locus-specific relative nonrecurrence risk ratio of disease for an offspring of the affected person.	
	Value Range	(0, 1) for ARPs (1, $+\infty$) for DRPs
	Default Value	None
	Required	Yes
	Applicable Notes	3
`sib_recurrence_risk`	Specifies λ_s, the locus-specific relative recurrence risk ratio of disease for a sibling of the affected person.	
	Value Range	(1, $+\infty$) for ASPs (0, 1) for DSPs
	Default Value	None
	Required	Yes
	Applicable Notes	3
`sib_nonrecurrence_risk`	Specifies λ_s^-, the locus-specific relative nonrecurrence risk ratio of disease for a sibling of the affected person.	
	Value Range	(0, 1) for ASPs (1, $+\infty$) for DSPs
	Default Value	None
	Required	Yes
	Applicable Notes	3
`cost_ratio`	Specifies the ratio (R) of the cost of recruiting a person to the cost of performing one marker assay.	
	Value Range	(0, $+\infty$)
	Default Value	None
	Required	Yes
	Applicable Notes	None

`num_loci`	Specifies the number (d) of disease loci being analyzed.	
	Value Range	{1, 2, 3, ...}
	Default Value	1
	Required	Yes
	Applicable Notes	None
`genome_length`	Specifies the length (M), in morgans, of the underlying genome.	
	Value Range	{1, 2, 3, ...}
	Default Value	36
	Required	Yes
	Applicable Notes	None
`linked_distance`	Specifies the maximum distance (g), in morgans, between any disease locus and a "linked" marker.	
	Value Range	$(0, +\infty)$
	Default Value	0.4
	Required	Yes
	Applicable Notes	None
`pic`	Specifies the value of the polymorphism information content (PIC) used to constrain marker selection.	
	Value Range	(0, 1]
	Default Value	1
	Required	Yes
	Applicable Notes	None
`heterogeneity`	Specifies the heterogeneity proportion (h) of sample pairs affected due to causes other than segregation at the linked locus.	
	Value Range	[0, 1)
	Default Value	0
	Required	Yes
	Applicable Notes	None
`screening_cost`	Specifies option (c) to include the cost of screening the population to obtain desired pairs.	
	Value Range	**Y** (include the cost) **N** (do not include the cost)
	Default Value	**N**
	Required	Yes
	Applicable Notes	None
`screened_proportion`	Specifies proportion of collected samples in the screened population (r_s)	
	Value Range	(0, 1]
	Default Value	1
	Required	Yes
	Applicable Notes	4

`conc_disc_ratio`	Specifies the ratio (r_p) of concordantly affected to discordant relative pairs to be sampled. Value Range: $(0, +\infty)$ Default Value: None Required: Yes Applicable Notes: 5
`num_stage_one_markers`	Specifies the number (m) of first-stage markers to be used. Value Range: $\{M+1, M+2$ ..., 1000M$\}$ Default Value: None Required: No Applicable Notes: 6
`num_stage_two_markers`	Specifies the maximum value for the number of markers (k) to be typed, during the second stage, on each side of the markers found to be significant during the first stage. Value Range: {0, 1, 2, ...} Default Value: None Required: Yes Applicable Notes: None
`num_pairs`	Specifies the number of relative pairs (n) to be analyzed. Value Range: {1, 2, 3, ...} Default Value: None Required: No Applicable Notes: 6

Notes

1. The `method` parameter is not applicable for sample data comprising both affected pairs and discordant pairs; only the approximate method (**A**) is implemented for such data.
2. The default value for α corresponds to a lod score of 3 if the `method` parameter is set to **A** (approximate).
3. The parameters `offspr_recurrence_risk` and `offspr_nonrecurrence_risk` are used by the proportion test for linkage, while the parameters `sib_recurrence_risk` and `sib_nonrecurrence_risk` are used by the mean test.
4. When the value of the `screening_cost` parameter is set to **N**, the `screened_proportion` parameter will be ignored.
5. When the value of the `screening_cost` parameter is set to **N**, or the `concordance_type` parameter is set to either A or D, the `conc_disc_ratio` parameter will be ignored. In other words, the `conc_disc_ratio` parameter is applicable only when the `concordance_type` parameter is set to **B**.
6. The user may specify a value for either `num_stage_one_markers` or `num_pairs`, but not both. If a value for either one of the parameters is specified, the other will be determined by the program. If neither parameter is specified, the program will determine both.

18.4 Output

DESPAIR produces a Standard Output File that includes:

- Title, version, and date of the program for each problem
- Control values specified by user
- For each k = 0, ..., max k, and for each mapping function, tabulation of optimal values of m and n with corresponding α*, cost (in units of the cost of typing one marker on one person), and the first and second stage marker spacings in centimorgans

For an example output, see the end of this document.

18.4.1 Error Messages

DESPAIR has an error checking routine. Values of any parameter that are out of bounds are not allowed. When an error is detected during the analysis, DESPAIR will identify the error and display the error message associated with it. The error messages that may be displayed are as follows:

- `The following fields were set to values out of bounds: <FIELD LIST>`
- `The exact test is not implemented for the case in which both concordant and discordant pairs are available.`
- `The test based on the proportion sharing 0 alleles i.b.d. is not available. The above results are for the mean test.`

Chapter 19

References

Amos CI, Dawson DV, Elston RC (1990) *The Probabilistic Determination of Identity-by-Descent Sharing for Pairs of Relatives from Pedigrees.* American Journal of Human Genetics 47:842-853

Bickeboller H, Clerget-Darpoux F. (1995) *Statistical Properties of the Allelic and Genotypic Transmission/Disequilibrium Test for Multiallelic Markers*. Genetic Epidemiology 12(6):865-870

Boehnke M (1991) *Allele Frequency Estimation from Data on Relatives*. American Journal of Human Genetics 48:22-25

Bonney GE (1984) *On the statistical determination of major gene mechanisms in continuous human traits: regressive models*. Am J Med Genet; 18: 731-749

Bonney GE (1986) *Regressive logistic models for familial disease and other binary traits*. Biometrics; 42: 611-625

Bonney GE (1998) *Regressive Models*. Encyclopedia of Biostatistics, Vol 5; 3755-3762

Box GEP, Cox DR (1964) *An analysis of transformations*. J Roy Stat Soc [B]; 26: 211-252

Cannings C, Thompson EA, Skolnick MH (1978) *Probability functions on complex pedigrees*. Adv Appl Prob; 10:26-61

Carroll RJ, Ruppert D (1984) *Power Transformations When Fitting Theoretical Models to Data*. American Journal of the Statistical Association 79:321-328

Cleves MA, Olson JM, Jacobs KB (1997) *Exact Transmission-Disequilibrium Tests with Multiallelic Markers*. Case Western Reserve University School of Medicine Internal Paper

Chen H, Chen J, Kalbfleisch JD (2001) *A Modified Likelihood Ratio Test for Uncertain-Haplotype Transmission*. Journal of the Royal Statistical Society (B); 63:19-29

Curtis D and Sham PC (1995) *An Extended Transmission/Disequilibrium Test (TDT) for Multi-Allele Marker Loci.* Ann Hum Genet; 59: 323-336

Demenais FM, Murigande C, Bonney GE (1990) *Search for faster methods of fitting the regressive models to quantitative traits*. Genet Epidemiol; 7: 319-334

Demenais FM, Elston RC (1981) *A General Transmission Probability Model for Pedigree Data*. Am J Hum Genet; 33:300-306

Elston RC, Stewart J (1971) *A general model for the genetic analysis of pedigree data*. Hum Hered; 21: 523-542

Elston RC, Bonney GE (1986) *Sampling via Probands in the Analysis of Family Studies*. Proceedings of the 13^{th} International Biometric Conference

Elston RC, George VT, Severtson F (1992) *The Elston-Stewart algorithm for continuous genotypes and environmental factors*. Human Hered; 42:16-27

Feingold E, Brown PO, Siegmund S (1993) *Gaussian Models for Genetic Linkage Analysis Using Complete High-Resolution Maps of Identity by Descent.* American Journal of Human Genetics 53:234-251

Fernando RL, Stricker C, Elston RC (1993) *An Efficient Algorithm to Compute Posterior Genotypic Distribution for Every Member of a Pedigree Without Loops.* Theory of Applied Genetics 87:89-93

Fernando RL, Stricker C, Elston RC (1994) *The finite polygeinc mixed model: An alternative formulation for the mixed model of inheritance*. Theor Appl Genet; 88: 573-580

George VT, Elston RC (1987) *Testing the association between polymorphic markers and quantitative traits in pedigrees*. Genetic Epidemiol; 4:193-201

George VT, Elston RC (1988) *Generalized modulus power transformations.* Commun Statist – Theory Meth; 17: 2933-2952

George VT et. al (1999) *A Test of Transmission/Disequilibrium for Quantitative Traits in Pedigree Data, by Multiple Regression*. American Journal of Human Genetics 65:236-245

Ginsburg E, Malkin I, Elston RC (2003) *Sampling correction in pedigree analysis*. Stat Appl Genet Mol Biol; 2: Article 2

Go RCP, Elston RC, Kaplan EB. (1978) *Efficiency, robustness of pedigree segregation analysis*. Am J Hum Genet;30: 28-37

Goddard KAB, Witte JS, Suarez, BK, Catalona, WJ, Olson, JM (2001) *Model-free Linkage Analysis with Covariates Confirms Linkage of Prostate Cancer to Chromosomes 1 and 4*. American Journal of Human Genetics 68:1197-1206

Idury RM, Elston RC (1996) *A Faster and More General Hidden Markov Model Algorithm for Multipoint Likelihood Calculations.* Human Heredity 47: 197-202

Ito et al. (2003) *Estimation of Haplotype Frequencies, Linkage-Disequilibrium Measures, and Combination of Haplotype Copies in Each Pool by Use of Pooled DNA Data*. American Journal of Human Genetics 72(2):384-398

Karunaratne PM, Elston RC (1998) *A multivariate logistic model (MLM) for analyzing binary family data*. Am J Med Genet; 76: 428-437

Kuglyak L, Lander ES (1995) *Complete Multipoint Sib-Pair Analysis of Qualitative and Quantitative Traits.* American Journal of Human Genetics 57: 439-454

Lander ES, Green P (1987) *Construction of Multilocus Genetic Maps in Humans.* Proceedings National Academy of Science USA 84:2363-2367

Lange K, (1997) *An Approximate Model of Polygenic Inheritance*. Genetics 147:1423-1430

Lange K, Elston RC (1975) *Extensions to Pedigree Analysis I-Likelihood Calculations for Simple and Complex Pedigrees*. Human Heredity 25:95-105

Olson JM, and Wijsman EM (1993) *Linkage between quantitative trait and marker loci: methods using all relative pairs*. Genet Epidemiol; 10:87-102

Olson JM, Jacobs KB, Cleves MA (1997) *Exact tests of table symmetry*. Internal paper

Olson JM (1999) *A General Conditional-Logistic Model for Affected-Relative-Pair Linkage Studies*. American Journal of Human Genetics 65: 1760-1769

Olson JM, Song Y, Lu Q, Wedig GC, Goddard KB (2004) *Using overall allele-sharing to detect the presence of large-scale data errors and parameter misspecification in sib-pair linkage studies*. Human Heredity 58:49-54

Parzen E (1962) *On Estimation of a Probability Density Function and Mode.* Annals of Mathematical Statistics 33: 1065-1076

Pericak-Vance MA, Elston RC, Conneally PM, Dawson DV (1983) *Age-of-Onset Heterogeneity in Huntington's Disease Families.* Journal of Human Genetics 14: 49-59

Quade SR, Elston RC, Goddard KA (2005) *Estimating Haplotype Frequencies in Pooled DNA Samples when there is Genotyping Error*. BMC.Genet. 6 (1):25

Rice JP, Neuman RJ, Hoshaw SL, Daw EW, Gu C (1995) *TDT with covariates and genomic screens with mod scores: their behavior on simulated data*. Genet Epidemiol; 12: 659-664

Risch, N (1987) *Assessing the role of HLA-linked and unlinked determinants of disease.* American Journal of Human Genetics 40:1-14

Risch, N (1990) *Linkage Strategies for Genetically Complex Traits. III. The effect of Marker Polymorphism on Analyses of Affected Relative Pairs.* American Journal of Human Genetics 46: 242-253

Scott D, Szewczyk W (2000), *Fitting Mixtures of Regression Models by L2E*

Self S, Liang K (1987), *Asymptotic properties of maximum likelihood estimators and likehood ratio tests under nonstandard conditions*. J Am Stat Assoc 82:605-610

Sobel E, Lange K (1996) *Descent Graphs in Pedigree Analysis: Applications to Haplotyping, Location Scores, and Marker-Sharing Statistics.* American Journal of Human Genetics 58:1323-1337

Spielman RS, Ewens WJ. (1996) *The TDT and Other Family-Based Tests for Linkage Disequilibrium and Association*. American Journal Human Genetics 59:983-989

Spielman RS, McGinnis RE, Ewens WJ (1993) *Transmission Test for Linkage Disequilibrium: The Insulin Gene Region and Insulin-Dependent Diabetes Mellitus (IDDM).* American Journal Human Genetics 52:506-516

Wang S, Kidd KK, Zhao H (2003) *On the Use of DNA Pooling to Estimate Haplotype Frequencies*. Genetic Epidemiology 24:74-82

Wijsman EM, Amos CI (1997) *Genetic Analysis of Simulated Oligogenic Traits in Nuclear and Extended Pedigrees: Summary of GAW10 Contributions*. In: Goldin L, Bailey-Wilson J, Borecki I, Falk C, Goldstein A, Suarez B, and MacCluer J. *Genetic Analysis Workshop 10: Detection of genes for complex traits*. Genetic Epidemiology, 14, S719-S736

Whittemore AS, Tu IP (1998) *Simple, robust linkage tests for affected sibs*. American Journal of Human Genetics 62:1228-1242

< <DESPAIR REFERENCES> >

Blackwelder WC, Elston RC. [1985]: A comparison of sib-pair linkage tests for disease susceptibility loci. Genet Epidemiol; 2: 85-97.

Guo X, Elston RC. [1999]: Linkage information content of polymorphic genetic markers. Hum Hered; 49:112-118.

Guo X, Elston RC. [2000]: Two-stage global search designs for linkage analysis II: Including discordant relative pairs in the study. Genet Epidemiol; 18:111-127.

Guo X, Olson JM, Elston RC, Niu T. [2002]: The linkage information content value of polymorphism genetic markers in model-free linkage analysis. Hum Hered; 53:45-48.

Elston RC. [1992]: Designs for the global search of the human genome by linkage analysis. In: Proceedings of the XVIth International Biometric Conference, Hamilton, New Zealand, December 7-11, 1992, pp 39-51. Elston RC, Guo X, Williams L. [1996]: Two-stage global search designs for linkage analysis using pairs of affected relatives. Genet Epidemiol; 13:535-558. Risch N. [1990]: Linkage strategies for genetically complex traits. II. The power of affected relative pairs. Am J Hum Genet; 46:229-241.

Ziegler A, Böddeker I, Geller F, Müller H, Guo X. [2001]. On the total expected study cost in two-stage genome-wide search designs for linkage analysis using the mean test for affected sib pairs. Genet Epidemiol; 20:397400.