Statistical Analysis for Genetic Epidemiology
(S.A.G.E.)
Version 6.4.2
User Reference Manual

Department of Epidemiology and Biostatistics
Wolstein Research Building
2103 Cornell Rd
Case Western Reserve University
Cleveland, Ohio 44106-7281

January, 2021

S.A.G.E. is an open-source software available through our web site:
http://darwin.cwru.edu

NOTICE

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

S.A.G.E. 6.4.2 [2021]. Statistical Analysis for Genetic Epidemiology http://darwin.cwru.edu.

You are requested to send bibliographic information to opensagecwru@gmail.com about every paper in which S.A.G.E. is used (author(s), title, journal, volume and page numbers) to be posted in https://github.com/elstonsage/SageCore/wiki/Publication, where you can find a list of papers by other users of S.A.G.E. that we know of.

Contents

[1] Covariates are pair-specific and are allowed only in the one-parameter model.

Chapter 1

Introduction

Statistical Analysis for Genetic Epidemiology (S.A.G.E.) is a collection of freely available compiled C++ programs that perform a wide variety of genetic analyses on both family data and data on unrelated individuals. The range of functionality includes tools for

- extracting summary statistics describing the data and evaluating general data quality,
- estimating allele frequencies and testing Hardy-Weinberg proportions,
- 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 in both family data and unrelated individuals.

S.A.G.E. runs on a variety of platforms: Linux, Windows, Solaris and Mac/OSX. The programs may be run either from a command line or 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 data files and, unless otherwise stated, the dependent traits may be discrete (including dichotomous data) or quantitative.

Users may download the current version of S.A.G.E. at any time from our web site, and please check for the most up-to-date information on the current version of S.A.G.E. programs at the following URL:

http://darwin.cwru.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 INFO*rmation*: 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 founder allele frequencies and the inbreeding coefficient from each of a set of markers on related individuals with known pedigree structure, and in the case of co-dominant markers generates marker locus description files, needed by GENIBD, MLOD, and other S.A.G.E. programs.

1.1.4 Familial Aggregation

ASSOC

Marker-Trait ***ASSOC****iations in Pedigree Data* : Simultaneously estimates and tests from pedigree data the association between a trait and covariates, which can include marker phenotypes (e.g. for genomewide association,GWAS) 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. This program can also provide output that can be used by the GMDR Utility in the S.A.G.E. GUI to provide input for the program GMDR (not part of S.A.G.E.) that performs Multifactor Dimensionality Reduction.

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 quantitative, 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.

1.1.7 IBD Allele Sharing Analysis

GENIBD

***GEN**erate **IBD**sharing probabilities* : Generates both single- and multi-marker 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 a Monte Carlo Method is 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-marker 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-marker 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. Alternatively, the one-parameter model can incorporate unaffected and discordant pairs.

SIBPAL

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

RELPAL

***REL**ative **P**air **A**na**L**ysis* : Performs a regression-based univariate or multivariate model-free two-level Haseman-Elston linkage analysis that models trait data from relative pairs as a function of marker allele sharing identity-by-descent (IBD), as proposed by Wang and Elston (2005, 2006). Available analyses can use both single- and multi-point IBD information, and models allow for both binary and quantitative traits caused by segregation at multiple genetic loci, including epistatic interactions and covariate effects.

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. These estimates are then used to produce two new quantative traits, trait susceptability and an age of onset residual, that can then be used in model-free linkage analyses.

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 in the presence of familial correlations the association between trait (binary or quantitative) and covariates, which can include marker phenotypes that have been transformed into quantitative covariates, from pedigree data and/or unrelated individuals. Together with the *Transmitted Allele Indicator* (available as a user-defined function), performs a pedigree transmission disequilibrium test (TDT). This program will also estimate heritability and environmental familial correlations.

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, and performs likelihood ratio tests and permutation tests to compare haplotype frequency distributions famong populations.

1.1.13 Study Design

DESPAIR

Determines optimal two-stage linkage study design for affected relative pairs. Determines the minimally sufficient number of concordant and/or discordant pairs, and also the number of equally spaced markers, needed for the initial phase of a proposed linkage study.

1.1.14 SNP Marker filtering and SNP sequence analysis

SNPCLIP

Filters SNP markers on the basis of their minor allele frequencies, consecutive correlations, map locations and missingness. Within SNPCLIP is MUGS (Maximum Unbroken Genotypic Sequence), an analysis that aims to find maximum length common haplotypes in a group of individuals, or in each of two groups of individuals (e.g. affected and unaffected). This program is only accessible from the GUI and is described in the S.A.G.E. GUI Manual.

1.2 Program Limitations

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

1. each genetic marker has a known genotype-phenotype relation (which may be either deterministic or probabilistic),
2. the founders of each constituent pedigree[1] are not inbred[2] and are unrelated to one another, and the pedigrees do not contain loops (except that MLOD can analyze small peigrees that have loops).
3. the members of each constituent pedigree are unrelated to the members of any other constituent pedigree.

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 3.2.4) are printed using a **bold** font.
3. Examples of parameter files are printed using a `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.

[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.

Chapter 2

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[1].

2.1 Graphical User Interface (GUI)

Please refer to the S.A.G.E. GUI user manual for how to run S.A.G.E. programs using the Graphical User Interface (GUI).

2.2 Command Line

Note that, whereas every program described in this manual (except DESPAIR, which is a web-based interactive program) can be run using the GUI, there are some programs that can only be run using the GUI. See the S.A.G.E. GUI user manual for these programs.

A S.A.G.E. program is run by specifying the name of the program followed by the input files on the command line. The input files can be listed in two different ways, one with flags and another without.

With flags, the syntax requires the user to precede each filename with a flag indicating the filetype as follows. Order does not matter if flags are used.

```
Flag Filetype       Description
---- ------------   ------------------------
-p   parameter      Parameter File
-d   pedigree       Pedigree Data File
```

[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 may nevertheless be possible for the user to generate an invalid S.A.G.E. command through the GUI. The user is therefore advised to check the information file (*.inf) as well as the standard S.A.G.E. outputs to the console when the program does not seem to work as expected.

```
-l   locus          Locus Description File
-g   genome         Genome Description File
-m   trait model    Trait Locus Description OR
                    Type Probability File
-i   ibd            IBD Sharing File
```

In the absense of flags, filenames must be given in the specified order.

The program specific usage information can be displayed by entering an application name or an application name followed by -h or -? at the command prompt. For example, entering "`decipher -h`" will result in the following.

```
usage (without flags): decipher <parameters> <pedigree> [locus]
usage (with flags): decipher <-p parameters> <-d pedigree> [-l locus] [-g genome]
Command line parameters:
parameters - Parameter File
pedigree   - Pedigree Data File
locus      - Locus Description File (optional)
genome     - Genome Description File (optional)
```

Items in angled brackets (<>) are required. Those in square brackets ([]) are optional. Note that in this example the genome file is not listed in the unflagged format. Therefore, a user wishing to use a genome file MUST use the flagged format.

Here is a second example.

```
usage (without flags): sibpal <parameters> <pedigree> <ibd> ...
usage (with flags): sibpal <-p parameters> <-d pedigree> <-i ibd> ...
Command line parameters:
parameters  - Parameter File
pedigree    - Pedigree Data File
ibd         - IBD Sharing File
```

The ellipsis, (. . .), indicates that more than one ibd file may be supplied, for example "`sibpal -p my_par -d my_ped -i ibd1 -i ibd2 -i ibd3`".

Multiple pedigree files may be supplied for any of the S.A.G.E. applications, in which case only the name of the first pedigree file is given on the command line. Pedigree blocks in the parameter file must be supplied for each pedigree file. If there are multiple files, file name must be part of these pedigree blocks.

A typical run of a S.A.G.E. program, for example FCOR, may look like the following:

```
>fcor -p data.par -d data.ped

FCOR -- xx Nov 201x hh:mm:ss -- [S.A.G.E. v6.x.0; xx Nov 201x]

Reading Parameter File.....................done.
Reading pedigree file.....................
              from data.ped..............done.
Sorting pedigrees.........................done.

.
.
.

Analysis complete!
```

Chapter 3

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

[a] This file, as well as the genome description file, can be produced using the graphical user interface (GUI).

[b] Single-marker 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
3.6	Information output file	Contains informational diagnostic messages, warnings and program errors. Each program generates one information output file. Information files are automatically named with a "inf" extension (for example, "segreg.inf"). **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
3.7	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.

3.1 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[1].

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] Some 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.

3.1.1 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. Each record must end with a carriage return or linefeed character[2]

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

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

and contains the following fields:

Field	Value Type	Description	Note
Pedigree ID	character string or numeric	Uniquely identifies a particular pedigree within the file.	1
Individual ID	character string or numeric	Uniquely identifies a particular individual WITHIN a pedigree.	2
Parent ID	character string or numeric	Identifier of the individual's parent (either father or mother).	3
Parent ID	character string or numeric	Identifier of the individual's other parent (either mother or father, depending on which was specified previously).	3
Sex	character string or numeric	Individual's sex.	4
Quantitative Traits, and Covariates	numeric	Observational trait data specific to the individual.	5
Discrete Traits, and Covariates	character string or numeric	Observational trait data specific to the individual.	5
Genotype Data	character string or numeric	Genotypic data specific to the individual.	5
Other Fields	character string or numeric	Other data specific to the individual.	5

1. If the Pedigree ID field is absent in the pedigree data file, all individuals are assumed to belong to the same pedigree.

2. 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.

3. At the user's option, the pedigree data file may list the father's ID first, followed by the mother's ID, or vice versa. Also note that partial lineage, i.e. only one parent specified, is not allowed. If Parent ID fields are absent in the pedigree data file, all individuals are treated as singletons unless there exists a Pedigree ID field and the user chooses to treat the individuals sharing the same Pedigree ID as sibs, by including the `treat_as_sibs` parameter in the parameter file (see 3.2.5.2).

4. Incorrect use of the word *gender* is studiously avoided here. As the poet says, "Nouns have gender, whereas people have sex ... and enjoy it!" If the Sex field is absent while the Parent ID fileds exist in the pedigree data file, the user is required to explicitly acknowledge this situation by including the `no_sex_field` parameter in the parameter file (see 3.2.5.2) to proceed with any analyses. However, the analyses may produce unpredictable results.

5. Even though some fields are not required to be included in the pedigree data file, they are requred for many analyses to be meaningful. For eaxample, genotypic data are required for programs that perform linkage analysis, allelic association analysis, etc.

The individual fields in a record for an individual 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*[3] that specify the names and types of the fields in your pedigree 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[4]. Programs that readily generate data in a character delimited form are spreadsheet programs like Microsoft Excel, most pedigree drawing programs, and most database programs. Microsoft Excel files can be used to run S.A.G.E. using the GUI. Please refer to the GUI user manual.

THE FORMAT OF THE CHARACTER DELIMITED 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 DATA FILE; OR, ALTERNATIVELY, IT MAY BE GIVEN AS A SET OF PARAMETERS IN A PEDIGREE BLOCK WITHIN THE PARAMETER FILE.

The name 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 parameters and attributes in the pedigree block (see 3.2.5) of the parameter file to map the field names to data field types. It is not necessary to specify all fields that exist in the pedigree data file in the pedigree block of the parameter file, but it is important that no field in the pedigree data file be used as a parameter value more than once. Whitespace is stripped from the beginning and end of the content of each field.

Several options are provided to let the user modify the way a character delimited pedigree data file 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 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
```

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

[4] 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.

Here is a typical pedigree data file in comma delimited pedigree data file that includes the name of each field in a header line:

```
PID, ID,P1,P2, SEX, JUNK, D42S1,D42S2,D42S3,D42S4,D42S5,D42S6, TRAIT1,CAT
1018, 1, 0, 0,   m,  0/0,   0/0,  0/0,  0/0,  0/0,  0/0,  0/0,   XXXX,RED
1018, 2, 0, 0,   x,  0/0,   0/0,  0/0,  0/0,  0/0,  0/0,  0/0,   XXXX,GREEN
1018, 3, 1, 2,   f,  5/3,   1/3,  6/7,  8/1,  2/2,  7/2,  7/1,   23.1,BLUE
1018, 4, 1, 2,   f,  5/3,   1/3,  8/7,  8/1,  2/2,  7/2,  7/2,   44.1,BLUE
1018, 5, 1, 2,   f,  5/3,   1/3,  8/7,  8/2,  2/2,  7/2,  7/1,   XXXX,RED
1018, 6, 1, 2,   m,  5/3,   1/3,  8/7,  8/1,  3/2,  7/2,  4/1,    9.3,RED
```

Suppose each record in the above 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	TRAIT1
Categorical trait	CAT
Marker D42S1	D42S1
Marker D42S2	D42S2
Marker D42S3	D42S3
Marker D42S4	D42S4
Marker D42S6	D42S6

then the following `pedigree` block in the parameter file can be used to read this pedigree data file (in the following and elsewhere in this document, any line that starts with '#' is a comment line and is ignored by the program; see 3.2.2):

```
# Example - character delimited
pedigree
{
   # 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, male="m",female="f",missing="x"
   parent_id=P1
   parent_id=P2
   trait="TRAIT1",name="DBP",missing="XXXX" # order is irrelevant
   trait="CAT",categorical,values="RED,GREEN,BLUE"
   marker=D42S4
   marker=D42S6
   marker=D42S1
   marker=D42S2
   marker=D42S3
   # Pedigree encoding information:
   individual_missing_value="0"
}
```

3.1.2 Pedigree Data Quality

Users are always well-advised to ensure that their pedigree data files are as error-free as possible[5], 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. Missing data will typically 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.*

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

3.2 The Parameter File

User options for analysis are specified to S.A.G.E. programs as a list of instructions within a *parameter file* [6]. 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 3.2.2). It commonly consists of four different types of instruction blocks; one or more `pedigree` blocks to determine how to read and interpret the contents of the given pedigree data file(s) (see 3.2.5), one `marker` block to set overall options on how to read marker phenotype data (see 3.2.6) if there exists marker phenotype data in the pedigree data file(s), zero or more `function` blocks to create new traits or covariates as a function of existing data (see 3.2.7), and zero or more program-specific `analysis` blocks which include the program specific options (see the program-specific Input section).

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[7]. 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.

3.2.1 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 file, and is expected to be particularly beneficial to novice users of the software, who may initially skip many of the details in the rest of this chapter, but it is recommended that they read 3.2.7. A function tool is available in the GUI and can be used at any time after the pedigree file has been imported into a project. Experienced users who prefer to create and edit their parameter files directly continue to have the option of doing so.

[6] 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 (see1.3) is to make the context of this word as clear as possible.

[7] 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.

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.

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[8].

3.2.2 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[9], 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. 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. Dut to the recursive nature of their definition, statements can be *nested* when listed in the parameter file. That is, a particular statement may be

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

[9] `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.

specified as containing another *statement* which itself comprises one or more *statements*, etc. This manual refers to outermost *statements* containing one or more *statements* within a pair of braces as *blocks.* When a brace-enclosed block is nested within another enclosing block, the nested blocks are referred to as *sub-blocks* .

Some possible statement structures are:

```
Example 1: parameter

Example 2: parameter, attribute = value

Example 3: parameter = value

Example 4: parameter = value, attribute

Example 5: parameter = value, attribute = value, attribute = value

Example 6: parameter = value
           {
             parameter = value
             parameter, attribute = value
           }

Example 7: parameter = value
           {
             parameter = value
             parameter
             {
               parameter = value
               parameter, 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 provides clarifying details and examples of parameter file syntax.

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 3.2.4).

2. The names of 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[10] by

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

S.A.G.E. programs. They may usually be inserted or omitted from statements at the user's discretion.[11]

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

5. 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
   }
}
```

6. 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
   }
}
```

3.2.3 `Parameter` and `Attribute` Values

3.2.3.1 Character Strings

When a particular parameter or attribute takes a *character string*[12] for its value, the user should enter the desired *alphanumeric* character sequence [13] after the equal sign (=). Enclosing *double quotes*[14] are only required in the following cases:

- the string contains blank spaces (e.g., "Alice in Wonderland"),

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

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

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

[14] 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.

- the string contains *non-alphanumeric* characters[15] (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 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* .

3.2.3.2 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 described as follows:

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 numberπ = 3.141592654...
- **e**, designating the base of the natural logarithms = 2.718281828459...

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

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
   }
}
```

3.2.4 Reading and Interpreting the Syntax Tables

For every *statement* 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
- the default value
- whether or not the `parameter` or `attribute` 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.

Note that the information on the range of valid or possible values and the default value is only relevant when a value is required for the given `parameter` or `attribute`.

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[a]	Declare respective pedigree field names for pedigree ID[b], individual ID, parental ID.
	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_field	Specifies pedigree field name for Sex.
	Value Range: Character string representing the valid name of a field in the pedigree data file.
	Default Value: None
	Required: No
	Applicable Notes: None
, male[g]	Specifies male sex code
	Value Range: Character string. Typical values are: 1, 0, M, m
	Default Value: M
	Required: No
	Applicable Notes: None
no_sex_field	Specifies that there are no field exist for sex_field in the pedigree data file.
	Value Range: N/A[h]
	Default Value: N/A
	Required: Yes if parent fields are present but sex field is not. Otherwise, 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 any 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 there is no default value specified.

[e] If **Yes**, then the listed parameter or attribute is required, and the user must explicitly enter the listed parameter or attribute into the parameter file, optionally followed by an assignment expression with a value. If **No**, then the listed parameter or attribute is NOT required because either it is an optional feature or 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] Attributes are indented with respect to their associated parameters, but appear in the same cell. Relevant explanatory information appears to their immediate right.

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

3.2.5 The `pedigree` Block

A `pedigree` block determines how to read and interpret the contents of the given pedigree data file. Unless otherwise noted, all parameters and their corresponding attributes must be specified within a `pedigree` block of a parameter file which starts with a `pedigree` parameter.

The following table shows the syntax for a `pedigree` parameter:

`parameter` `[, attribute]`	**Explanation**	
`pedigree`	Starts a pedigree specification block.	
	Value Range	N/A
	Default Value	N/A
	Required	Yes
	Applicable Notes	None
`, 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	1, 2

Notes

1. 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 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, 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
   ...
}

pedigree, 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
   ...
}

pedigree, 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
   ...
}
```

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

3.2.5.1 Parameters for General Pedigree File Formatting

The following table lists the parameter for the pedigree data file formatting that may occur in a `pedigree` block.

parameter [, attribute]	Explanation
`format`	Specifies a delimited, sequential listing of each field of a character delimited pedigree data file. Value Range: Quoted character string. Default Value: None Required: No if the pedigree data file has a header record as the first entry. Otherwise, yes. Applicable Notes: 1
`delimiters`	Specifies delimiter characters. Value Range: Quoted character string. Default Value: "`,\t`" Required: No Applicable Notes: 2
`whitespace`	Specifies characters that must be treated as whitespace. Value Range: Quoted character string. Default Value: " " (blank space) Required: No Applicable Notes: 2
`delimiter_mode`	Specifies delimiter interpretation mode. If set to `multiple`, then a set of successive delimiters in the data file will be treated as a single delimiter. Value Range: {single, multiple} Default Value: single Required: No Applicable Notes: 3
`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	4

Notes

1. The `format` parameter is used to list the name of each field in the character delimited 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. 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.

3. 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**.

4. 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**.

3.2.5.2 Parameters for Individual and Family Identification Fields

The following table lists the parameters and attributes for individual and family identification fields in the pedigree data file that may occur in a `pedigree` block.

parameter [, attribute]	Explanation
`pedigree_id`	Specifies pedigree data field name for pedigree ID. Value Range: Character string representing the valid name of a field in the pedigree data file. Default Value: None Required: No Applicable Notes: 1
`individual_id`	Specifies pedigree data field name for individual ID. Value Range: Character string representing the valid name of a field in the pedigree data file. Default Value: None Required: Yes Applicable Notes: None
`parent_id`	Specifies pedigree data field names for parental ID. Value Range: Character string representing the valid name of a field in the pedigree data file. Default Value: None Required: No Applicable Notes: 2
`treat_as_sibs`	If parent id fields are not specified, instructs S.A.G.E. to create dummy parents for every unique pedigree id. All individuals sharing the same pedigree id will be treated as full sibs. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 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_field	Specifies pedigree data field name for Sex.	
	Value Range	Character string representing the valid name of a field in the pedigree data file.
	Default Value	None
	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 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
no_sex_field	If parent id fields are present but sex field is not, this parameter must be included as an acknowledgement that S.A.G.E. will not be able to infer the sex of any parent-offspring relationships.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	5
no_sex_ok	Directs S.A.G.E. to continue analyses in the presence of missing sex information for some individuals.	
	Value Range	{true, false}
	Default Value	None
	Required	No
	Applicable Notes	6

Notes

1. When the `pedigree_id` parameter is specified, 3 new traits (`FAMILIAL_INDICATOR`, `FOUNDER_INDICATOR`, `PEDIGREE_SIZE`) are automatically created for each individual with values as follows:

   - FAMILIAL_INDICATOR

     $$\begin{cases} = 1 \text{ if an individual belongs to a pedigree of size} > 1 \\ = 0 \text{ otherwise} \end{cases}$$

   - FOUNDER_INDICATOR

     $$\begin{cases} = 1 \text{ if an individual has both parents missing} \\ \quad \text{(i.e., a founder except no descendent is required)} \\ = 0 \text{ otherwise} \end{cases}$$

   - PEDIGREE_SIZE - the number of individuals in the pedigree to which an individual belongs

2. The `parent_id` parameter is typically specified twice, once for each parent. If the `parent_id` parameters are not specified because they are absent in the pedigree data file, all individuals are treated as singletons unless there exists a Pedigree ID field specified with a `pedigree_id` parameter and the user chooses to treat the individuals sharing the same Pedigree ID as sibs by including the `treat_as_sibs` parameter in the pedigree block.

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

4. When the `sex_field` parameter is specfied, a new trait called `SEX_CODE` is automatically created for each individual with values as follows:

   - male = 0
   - female = 1

5. If `sex_field` is absent while `parent_id` fields exist in the pedigree data file, the user is required to explicitly acknowledge this situation by including the `no_sex_field` parameter in the pedigree block.

6. Many S.A.G.E. algorithms rely on sex for structural information. Even when sex does not affect the outcome of an analysis, missing sex information may cause the program to behave improperly or even crash. If your analyses involve pedigree structure, this should be corrected before continuing. If you would like to continue the analysis without correcting this issue, you must place a `no_sex_ok` parameter in the pedigree block.

3.2.5.3 Parameters for Trait and Covariate Fields

The following table lists the parameters and attributes for trait and covariate fields in the pedigree data file that may occur in a `pedigree` block.

parameter [, attribute]	Explanation	
`trait` `covariate`	Specifies pedigree names for 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 trait.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None
`, affected`	Specifies code for affected status of binary traits.	
	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 traits.	
	Value Range	Character string. Typical values are: U, 0, UNAFFECTED, Unaffected, no, false, neg
	Default Value	0
	Required	No
	Applicable Notes	None

, categorical	Indicator for categorical (nominal) trait. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 3
, values	Specifies which categorical values are considered valid (for a categorical trait). Value Range: A comms-delimited list of alphanumeric strings. Default Value: None Required: No Applicable Notes: 4
, name	Specifies a trait name different from the field name given in the pedigree header line. Value Range: Character string. Default Value: field name Required: No Applicable Notes: 5
covariate_list	Specifies a contiguous range of covariates, as listed in the pedigree data file, over which the subsequent attributes will apply. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 6
, start	Specifies the name of the first covariate in the list. Value Range: Character string Default Value: None Required: Yes if `covariate_list` is specified. Otherwise, no. Applicable Notes: 7, 8
, end	Specifies the name of the last covariate in the list. Value Range: Character string Default Value: None Required: Yes if `covariate_list` is specified. Otherwise, no. Applicable Notes: 7, 8
, missing , binary , affected , unaffected , categorical , values	Specifies single-covariate attributes to be applied to every covariate in the specified list (see attribute descriptions above). Value Range: See above Default Value: See above Required: See above Applicable Notes: See above

`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).	
	Value Range	Character string.
	Default Value	None
	Required	No
	Applicable Notes	9

Notes

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

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

   (b) `trait` fields are selected by many (but not all) analyses to be automatically used as major (dependent) 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 trait code (for a binary trait), or
   - a numeric value (for a quantitative trait).

2. The `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. **Remember: any field specified as a `trait` will automatically be analyzed by some S.A.G.E. programs**, whereas fields specified as covariate will be analyzed optionally, depending on whether or not they have been listed within the relevant analysis block.

3. If the parameter file indicates that a trait be interpreted as `categorical`, then S.A.G.E will interpret each unique alphanumeric trait value as a distinct categorical code. For instance, if the trait in question has values "red", "green", and "blue", then S.A.G.E. will automatically recognize those alphanumeric strings as unique categorical values. Please note that S.A.G.E. examines the entire alphanumeric string to determine its uniqueness. If a trait is specified as categorical, then values "1" and "1.000" will be understood as two different categorical values.

4. If a trait is specified as `categorical`, the user can (optionally) include the `values` attribute. This attribute takes the form of a comma-delimited list of alphanumeric strings. Each string is understood to be a unique and valid categorical value. When reading in the pedigree data,

if this attribute is included, S.A.G.E. will consider invalid any categorical value it encounters that is not already in the `values` list. For instance, `values = ''red, green, blue''` would limit allowable categorical values to those three strings. An individual with the value "redd" or "_green" would be considered missing for the trait in question.

5. A `name` attribute may optionally be specified for `trait` and `covariate` parameters. If a `name` attribute is not specified, the trait name is assumed to be the field name. This feature is useful when the field 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. If a `name` attribute is specified, this user specified trait name should be used in the analysis blocks and the name will be used in all output files.

   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:

   ```
   covariate = Trait1,     name = "Generic trait", 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" should be referenced as "Generic trait" within S.A.G.E. analyses, and similarly, the field originally designated as "Covariate1" should be referenced as "Covariate #1" within subsequent analyses.

6. The `covariate_list` parameter specifies multiple covariates simultaneously. This is preferred over specifying covariates individually when covariates all have the same formatting and there are many covariates in the pedigree data file. The `covariate_list` parameter has the following attributes, corresponding to the same attributes as used for specifying covariates individually except `name` attribute:

   - `missing`
   - `binary`
   - `affected`
   - `unaffected`
   - `categorical`
   - `values`

7. The `start` attribute indicates the first covariate in the set, and the `end` attribute specifies the last. The `start` and `end` attributes are required if the `covariate_list` parameter is specified, and their omission prevents parsing the covariate list (a fatal error is generated). Both the start covariate and end covariate must be present in the pedigree data file, with end following start. A fatal error will result if this is not the case. A `covariate_list` with `start` and `end` being the same value will be interpreted as a single covariate.

8. Multiple covariate lists can be specified, but they may not overlap. Similarly they may not overlap with any other fields. When multiple covariate lists and fields are not disjoint, a fatal error is issued.

9. Users may set the `missing` option for a `string` parameter just as with the `trait` and `covariate` parameters.

3.2.5.4 Parameters for Genotype Data Fields

The following table lists the parameters and attributes for genotype data fields in the pedigree data file that may occur in a `pedigree` block.

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
, x_linked	Indicator for X-linked marker. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 13
, y_linked	Indicator for Y-linked marker. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 13
, 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: 5
, 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: 6, 7
, maximum_allele_freq , maximum	Specifies maximum allele frequency for the marker. Value Range: [0, 1] Default Value: None Required: No Applicable Notes: 6
, equal , equal_allele_freq	Sets all allele frequencies to be equal. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 7
, complement , compl_allele_freq	Sets allele frequencies proportional to complementary values. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 8
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: 9
marker_list	Specifies a contiguous range of markers, as listed in the pedigree data file, over which the subsequent attributes will apply. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 10
, start	Specifies the name of the first marker in the list. Value Range: Character string Default Value: None Required: Yes if marker_list is specified. Otherwise, no. Applicable Notes: 11, 12
, end	Specifies the name of the last marker in the list. Value Range: Character string Default Value: None Required: Yes if marker_list is specified. Otherwise, no. Applicable Notes: 11, 12

, x_linked , y_linked , missing , delimiter , minimum , maximum , equal , complement	Specifies single-marker attributes to be applied to every marker in the specified list (see attribute descriptions above). Value Range: See above Default Value: See above Required: See above Applicable Notes: See above

Notes

1. For each locus, the information can be modified by adding the proper attributes to `marker` or `allele` parameter within the `pedigree` block. Each `marker` parameter and associated attributes in the `pedigree` block specifies the infomation relevant to that specific marker locus only while the information on `marker` block (see 3.2.6) applies to all marker loci.

2. The values assigned to them identify fields in the pedigree data file that contain allele or marker phenotipic information.

   Each allele field in a record may be any character string that represents:

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

   Each marker field in a record may be any character string that represents:

   - a missing value code,
   - an allele name, followed by 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. (Provided that the markers are codominant, the marker locus description file can be generated automatically by the FREQ program, so in general this requirement should not pose a great problem to the user. See 3.3 for markers that are not codominant.)

4. E.g., to specify three markers named "D42S1", "D42S2", "D42S3", a trait named "Trait1", a trait-marker[16] called "MOD", and a binary covariate named "Cov"; marker D42S1 is given by two allele fields, and the others are marker genotype fields:

[16] A *trait-marker* is simply an observable biological trait, such as blood type or an enzyme activity level, that is used in a model-based linkage analysis. We use the term *trait-marker* because the trait is used in the analysis like any other marker phenotype (eg., SNP or microsatellite phenotype) but *without* any assumption of codominance.

```
# Order is irrelevant
allele = D42S1a, name = D42S1 # First allele of D42S1
allele = D42S1b, name = D42S1 # Second allele of D42S1
marker = D42S2
marker = D42S3
trait_marker = MOD
trait = Trait1
covariate = Cov, binary, affected=1, unaffected=2, missing=3
```

5. A `name` attribute should be specified for `allele` and may optionally be specified for `marker` parameters. It should be specified whenever 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 data file.

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

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

   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.

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

8. This will complement all allele frequencies and then normalize them to sum to 1. In other words, the frequencies listed in the locus description file will be individually complemented, and the complement added to a sum for all alleles at the locus. Each complemented frequency is normalized by dividing by the sum of the complemented frequencies.

9. A `trait_marker` parameter should be included for each trait in the 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.

10. The `marker_list` parameter specifies multiple markers simultaneously. This is prefered over specifying markers individually when markers all have the same formatting and there are many markers in the pedigree file. The `marker_list` parameter has the following attributes, corresponding to the same attributes as used for specifying markers individually:

    - `x_linked`
    - `y_linked`
    - `missing`
    - `delimiter`
    - `minimum`
    - `maximum`
    - `equal`

- `complement`

11. The `start` attribute indicates the first marker in the set, and the `end` attribute specifies the last. `start` and `end` are required attributes if the `marker_list` parameter is specified, and their omission prevents parsing the marker list (a fatal error is generated). Both the start marker and end marker must be present in the pedigree data file, with end following start. A fatal error will result if this is not the case. A `marker_list` with `start` and `end` being the same value will be interpreted as a single marker.

12. Multiple marker lists can be specified, but they may not overlap. Similarly they may not overlap with any other fields. When multiple marker lists and fields are not disjoint, a fatal error is issued.

13. For X-linkage, hemizygous males need to be coded as homozygotes. Y-linkage is implemented only in the MARKERINFO program.

3.2.6 The `marker` Block

The `marker` block sets global options on how to read and interpret marker genotype fields in the pedigree data file. Unless otherwise noted, all parameters and their corresponding attributes must be specified within a `marker` block of a parameter file which starts with a `marker` parameter.

The following table shows the syntax for a `marker` parameter:

parameter [, attribute]	Explanation
marker	Starts a marker specification block. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `marker` block.

parameter [, attribute]	Explanation
allele_frequency	Specifies allele frequency adjustment. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
, equal	Sets all allele frequencies to be equal. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 1
, complement	Sets all allele frequencies proportional to complementary values Value Range: N/A Default Value: N/A Required: No Applicable Notes: 1
, 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. Value Range: [0,1] Default Value: None Required: No Applicable Notes: 1

, maximum	Ensures that allele frequencies are not set above a maximum value. Note that after normalization (to sum to 1) some allele frequencies may be larger than the set maximum.
	Value Range: [0,1]
	Default Value: None
	Required: No
	Applicable Notes: 1
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: 1
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: 1
covariate_function cov_func	Specifies the option to read in marker data as covariates.
	Value Range: {add, dom, rec}
	Default Value: none
	Required: No
	Applicable Notes: 2
, base_allele	Specifies the allele to based on to convert marker to covariate.
	Value Range: Character string
	Default Value: None
	Required: Yes if covariate_function is specified. Otherwise, no.
	Applicable Notes: 2
, allow_hemizygote , allow_hemi	Specifies the option to allow hemizygote allele to be read as covariate.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: 2

Notes

1. See notes 6 through 8 of Section 3.2.5.4. Any value specified for parameters and attributes in this `marker` block will be used for all marker and allele fields in the pedigree data file unless each `marker` or `allele` statement in the `pedigree` block overrides them otherwise.

2. This option causes all marker data to be read in as covariate values without using a function block, with the result that any analysis that uses "marker" data (see 3.2.5.4) cannot be performed. The value for `covariate_function` specifies the marker function: additive, dominant, or recessive. The `base_allele` attribute specifies the specific allele that will be considered to be 'Ai' in the marker function (see 3.2.7.3). Note that this option allows a marker to be used for some programs (e.g. TDEX) but is not intended for X-linkage: for X-linkage male hemizygotes must be coded as homozygotes unless `allow_hemizygote` attribute is set to true, then marker data with only one allele specified, the other a missing value, will be treated as homozygote.

3.2.7 The `function` Block

User-defined `function` parameters specify the creation of new traits 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 or covariate read from a pedigree data file in all S.A.G.E programs that use traits and covariates in the analyses.

The following table shows the syntax for a `function` parameter:

`parameter` `[, attribute]`	**Explanation**
`function`	Starts a function block. Value Range: N/A Default Value: None Required: No Applicable Notes: None
`, list`	Indicates that the function block should be executed once for every element in the given list (see note for further explanation). Value Range: markers, traits Default Value: None Required: No Applicable Notes: 1

Notes

1. If the `list` attribute is included, then the given function block will be executed for every item in the given list. The possible lists include all traits (traits and covariates, that is) or all markers given in the `pedigree` block. That is, the user may include either `list=traits` or `list=markers`. S.A.G.E. will interpret this to mean that it should generate a new, user-defined trait for every trait (or marker) in the list. To make the process more intuititve, S.A.G.E. will prepend the name of the newly-generated trait with the user-given name. Also, S.A.G.E. will recognize the `$name$` token in both the function expressions and constant expressions; for each element in the list, S.A.G.E. will substitute the element's name for the `$name$` token. For instance, consider a user who wants to apply the dominant() function to all markers present in the data. To apply the dominant() function to a single marker, the user would enter the following function block: `function {trait = dom, expression = ‘‘dominant(marker_name, ‘A’)’’}` (see 3.2.7.3). If, however, the user wants to use the list attribute to apply the dominant() function to all markers, the user would enter the following function block: `function, list = markers {trait = dom, expression = ‘‘dominant(marker_name, ‘A’)’’}`. Notice that the user has changed two aspects of the function block: the `list = markers` portion has been added, and the explicit name of the marker has been changed to the `$name$` token. S.A.G.E. will now apply the function block for every marker specified in the `pedigree` block using `marker`, `allele`, and `marker_list` parameters. If, for instance, three markers M1, M2, and M3 are specified, then S.A.G.E. will create three new traits (dom_M1, dom_M2, and dom_M3). Also, note that the `$name$` token

can be used in both the function expression itself and in any of the user-defined constants. The following are all valid uses of the list attribute:

```
function, list = markers
{
   trait = dom, expression = "dominant($name$, 'A')"
}

function, list = traits
{
   trait = squared, expression = "$name$ * $name$"
}

function, list = traits
{
   constant = c, expression = "0.275"
   trait = added, expression = "$name$ + c"
}
```

The following table lists the parameters and attributes 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: 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: Yes if `constant` parameter is specified. Otherwise, no. Applicable Notes: 1
`trait` `covariate`	Names function variable. Only one function variable per function statement is allowed Value Range: Character string. Default Value: None Required: Yes Applicable Notes: 2, 3
`, expression`	Specifies the algebraic expression used to calculate a value for the function variable. Value Range: Quoted character string. Default Value: None Required: Yes if `trait` or `covariate` parameter is specified. Otherwise, no. Applicable Notes: 4, 5, 6
`, 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 trait. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

, affected	Specifies code for affected status of binary traits.	
	Value Range	Character string
	Default Value	1
	Required	No
	Applicable Notes	None
, unaffected	Specifies code for unaffected status of binary traits.	
	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	7

Notes

1. The following constants[17] may be used for `expression`:

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

2. The two possible function variable types are `trait` and `covariate`. `Covariate` field is 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.

3. The value may be a character string representing the name of a new 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 and covariates. Thus the name "mean_bmi" is considered identical to "mean_BMI".

4. The value of `expression` should be an algebraic expression referring to one or more existing variables (traits, 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 for 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 to derive a new trait from existing traits:

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

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

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

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

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

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

7. 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.

   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"
}
```

The followng sections describe the operators and functions that may be used in `function` block expressions.

3.2.7.1 Operators

The following table indicates the operators that may be used. Those below the empty row are used to evaluate either zero (false) or one (true).

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[a]	expression = "Age / 10"
*	Multiplication	expression = "Weight * 1.19"
-	Subtraction	expression = "Height - AvgHeight"
+	Addition	expression = "Var_X + Var_Y"
!=, <>	Not equal to	expression = "Affected <> 0" expression = "Sex != M"
==	Equal to[b]	expression = "Affected == 0" expression = "Weight * (Affected == 1)"
>=	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)"

[a] If both operands are in integer form (contain no decimal point and are not in scientific notation), the result is integer also. Thus, 2.0 / 3 = .66667, but 2 / 3 = 0.

[b] The use of the comparison operator (**two** equal signs), "==", creates a logical expression whose evaluation results in either 1 (if true) or 0 (if untrue).

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. For example, we may have `expression = "(x + (y * z))"`.

3.2.7.2 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(x)	$\|x\|$	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

3.2.7.3 Marker Functions

The following functions are available for markers. 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 as follows:[18]

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

  $A*/A*$ returns 0,

  where $A*$ is any allele other than A_i.

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

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

  returns the value 1 or 0 based on the alleles present at the specified marker locus as follows:

  A_i/A_i returns 1

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

  where $A*$ is any allele other than A_i.

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

  returns the value 2, 1, or 0 based on the alleles present at the specified marker locus as follows:

  A_i/A_i returns 2

  $A_i/A*$ returns 1

  $A*/A*$ returns 0,

  where $A*$ is any allele other than A_i.

- `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 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,

  where $A*$ is any allele other than A_i or A_j.

Here are some examples.

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. Equivalent 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: If ABO is missing, the trait x will also be missing.

3. Another marker example:

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

3.2.7.4 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 or covariate. There are two basic steps to creating an adjusted variable:

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

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

3.2.7.4.1 Specifying the Classes for Adjusting Data

Specify each class within a `function` block as the values of an `expression` attribute for the `covariate` parameter[19]. The following example shows how to create three classes of a covariate[20] 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.

3.2.7.4.2 Creating a Mean-Adjusted Variable

Once the classes of the stratification variable have been defined, the mean-adjusted values of some other trait (or covariate) can be calculated using the classes of the stratification variable that defines **classes** of the trait. Specify the mean-adjustment within a `function` block as the value of an `expression` attribute for the `trait` parameter[21]. Assuming the data file lists a trait or covariate called "BP", the following example shows how to create a new mean-adjusted variable named "BP_AgeAdjMean":

[19] The classes could also be created from a trait; however, it usually makes more sense to create them from a covariate.
[20] In this example, "Age" is assumed to be a covariate; however the stratification variable could also be a trait.
[21] The variable could also be created for a covariate.

```
function
{
   trait = BP_AgeAdjMean, expression = "mean_adj(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 second 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 neighboring classes in the ordered list of values 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 variable, *BP_AgeAdjMean*, essentially becomes a new trait, and is surreptitiously added to the internal representation of the 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 less than or equal to 15,
- $\bar{x}_2$: the mean BP for individuals whose age is greater than 15 and less than or equal to 30,
- $\bar{x}_3$: the mean BP for individuals whose age is greater than 30.

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 less than or equal to 15,
- $BP_i - \bar{x}_2$; if the individual's age is greater than 15 and less than or equal to 30,
- $BP_i - \bar{x}_3$; if the individual's age is greater than 30.

3.2.7.4.3 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(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 less than or equal to 15,
- BP_i/s_2: if the individual's age is greater than 15 and less than or equal to 30,
- BP_i/s_3: if the individual's age is greater than 30,

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

3.2.7.4.4 Creating a Z-Score Variable

A standardized variable is obtained as follows:

```
function
{
   trait = BP_AgeZScore, expression = "z_score(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 less than or equal to 15,
- $(BP_i - \bar{x}_2)/s_2$: if the individual's age is greater than 15 and less than or equal to 30,
- $(BP_i - \bar{x}_3)/s_3$: if the individual's age is greater than 30.

3.2.7.4.5 Creating Adjusted Variable 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)"
}
```

3.2.7.5 Data Trimming and Winsorizing

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.

3.2.7.5.1 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 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 or covariate within S.A.G.E. applications.

3.2.7.5.2 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 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 or covariate within S.A.G.E. applications.

3.2.7.6 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 trait (assumed to be quantitative) is the dependent variable, Y . The primary independent predictor 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).

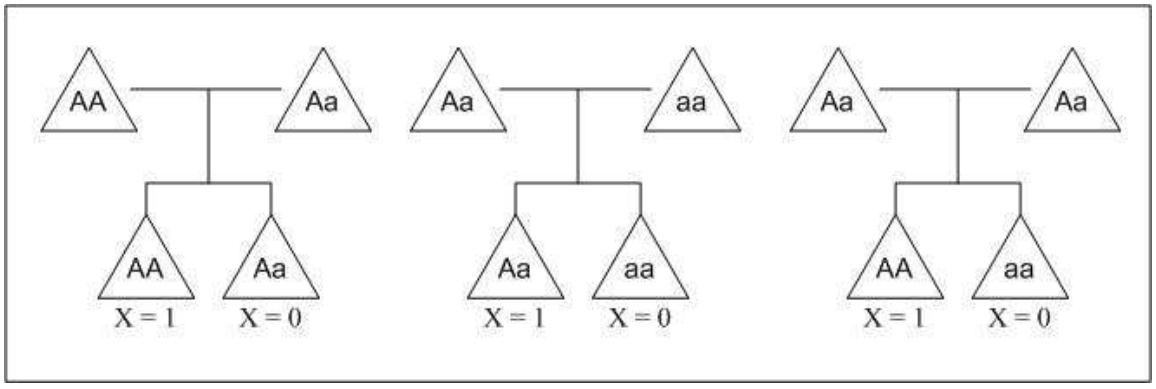


Figure 1: Offspring who are informative for linkage, from relevant parental matings. $\boldsymbol{A}$ is the associated allele of interest, and X is the transmitted allele indicator variable such that X =1 if $\boldsymbol{A}$ was transmitted from a heterozygous parent, and X =0 otherwise.

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[22]:

[22] 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		

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 or covariate within S.A.G.E. applications.

3.3 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 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 corresponding 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.

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. Allele frequencies should sum to 1. Otherwise, they will be normalized to do so. 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. 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 marker 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. A phenoset is **not** required for a codominant locus that is fully penetrant,

and the penetrance values are strictly optional. If no value is indicated, the phenotype is assumed to be fully penetrant and a value of 1 is assumed.

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. Consecutive genotypes 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.

In the following example, A, B, C, A1, A2, and O are the values of the alleles and, 1, 2, 3, 4, 5, and 6 are the values of the marker phenotypes as entered in the pedigree data file.

```
LOCA                        # locus name
A = 0.5                     #(allele/phenotype names are arbitrary
B = 0.25                    # and need not be in any particular order)
C = 0.25
;                           # first semicolon means no more alleles to list
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}
;                           # second semicolon means no more phenotyes/phenosets


ABO                         # name of next locus
A1 = 0.1904                 # allele frequencies should  sum to 1.0 (otherwise
A2 = 0.0612                 # they will be normalized to do so.)
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. 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.

For example, the following two locus descriptions are equivalent:

```
A                       A
1 = 0.645               1 = 0.6455
2 = 0.223               2 = 0.2230
3 = 0.1325              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}
;
```

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 genotype, the sum of all (here two) penetrances should equal 1.

3.4 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. The map function is used soley to convert genetic distances between consecutive markers into recombination fractions. As is the case for all other linkage programs, S.A.G.E. does not incorporate interference into linkage analysis. Map functions are not used during single-marker (two-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 the region name to indicate the region to be X-linked 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
`distance`	Specifies genetic distance, in centimorgans, between adjacent marker parameters. Value Range: (0,∞) Default Value: None Required: Yes Applicable Notes: 2, 3
`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. It is therefore advisable to include an initial marker "pter" as the initial marker for each chromosome.

2. There is a maximum of one genetic distance or recombination fraction value between each pair of markers. When doing multi-point analysis, there must be either a genetic distance or recombination fraction value between each pair of adjacent markers.

3. The S.A.G.E. GUI provides a genome map file wizard that can convert the marker coordinates to the genetic distance values between each pair of markers. Please refer to the GUI manual.

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
   {
      marker = " pter"              # Dummy marker name for p-terminal end of the chromosome
      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
        .
        .
        .
    }
}
```

3.5 IBD Sharing File

The IBD sharing file, produced by the S.A.G.E. program GENIBD, 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 generated in a prior analysis by the program GENIBD and used as input to the programs, LODPAL, SIBPAL, and RELPAL.
2. Packages other than S.A.G.E. may be able to use IBD sharing files produced by GENIBD 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.

3.6 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, traits, covariates and marker loci. This file is named "*program.inf*", indicating the name of the specific program that was run[23]. **ALTHOUGH NO ANALYSIS RESULTS ARE STORED IN THIS FILE, THE USER SHOULD MAKE A HABIT OF EXAMINING THE CONTENTS OF THIS FILE BEFORE OPENING ANY OTHER FILE PRODUCED BY A PROGRAM.**

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 and genotype. This file is named "*genome.inf*". Although no analysis results are stored in this file, errors relating to the markers and traits may be there.

3.7 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.

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

Chapter 4

AGEON

This program fits by maximum likelihood a mixed distribution for a binary trait (affected versus unaffected) with variable age of onset. The parameters estimated are the susceptibility to disease at age infinity (γ), the mean (μ) and variance (σ^2) of a power-normal age of onset distribution, and the power parameter (λ_1); a shift parameter (λ_2) may be specified. The mean, variance, and logit of susceptibility can each depend linearly on covariates. A class susceptibility covariate with six class values is generated according to the values of a parental binary trait. The parameter estimates can be used in a special function to produce of a set of eight new variables, for a binary trait with variable age of onset, any one of which may be used in a SIBPAL analysis.

4.1 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 value of the shift parameter λ_2 *must* be set such that $\lambda_2 +$ the minimum value for age-of-onset or age-at-exam, whichever is smaller, is > 0.

4.2 Theory

4.2.1 Basic notation

Let the number of sibs in the sample be n .

Let i index the sib: $i = 1, 2, \ldots, n$.

Let j index the class of sib generated according to the values of a parental binary trait.

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 φ and Φ be the normal density and cumulative distribution functions, and h be the transformation function as follows:

$$\varphi\left(x,\mu,\sigma^2\right)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\},$$

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left\{-\frac{1}{2}u^2\right\}du,\ \text{and}$$

$$h(x)=\begin{cases}\frac{(x+\lambda_2)^{\lambda_1}-1}{\lambda_1},\ \text{if}\ \lambda_1\neq 0\\ ln(x+\lambda_2),\ \text{if}\ \lambda_1=0\end{cases}.$$

AGEON uses an extension of the Box and Cox (1964) transformation to estimate λ_1, μ_i, σ_i^2 and γ_i, wherein γ_i is automatically a function of a parental binary trait (γ_j), assumed here to be affection status, and other user-specified covariates, and the last three parameters are defined possibly as functions of user-specified covariates (x_1, x_2, ...), as follows:

$$\mu_i=\mu+\xi_1x_{1i}+\xi_2x_{2i}+\cdots,$$

$$\sigma_i^2=\sigma^2+\xi_1x_{1i}+\xi_2x_{2i}+\cdots,\ \text{and}$$

$$\gamma_i=\frac{e^{\theta_i}}{1+e^{\theta_i}},\quad\text{where}\ \theta_i=\gamma_j+\xi_1x_{1i}+\xi_2x_{2i}+\cdots.$$

AGEON allows the susceptibility and the mean and variance of the power (λ_1) transformed shifted (λ_2) age of onset to be dependent on different sets of covariates, and in each case the maximum likelihood estimates of the regression coefficients are obtained.

As done elsewhere in S.A.G.E., the transformation is applied to both sides of the equation for the mean age of onset, so the estimates $\hat{\mu}$, $\hat{\xi_1}$, $\hat{\xi_2}$, ... are asymptotically median unbiased estimates on the original scale, rather than on the transformed scale, and so more meaningful. But note that the estimate of variance is now on the transformed scale, and is not very relevant unless $\lambda_1=1$, in which case the variance is not changed.

4.2.2 Classification

AGEON classifies sibs according to whether or not each parent's affection status is known and, if known, whether each parent is or is not affected. Using '?' for unknown affection status, 'A' for affected, and 'U' for unaffected, an individual's class can be expressed as an order-independent combination of any two of those symbols according to the value of the parental affection status, as shown in the following table:

Class	Description
??	Both parents are unknown
?A	One of the parents is unknown, the other is affected
?U	One of the parents is unknown, the other is unaffected
AA	Both parents are affected
AU	One of the parents is affected, the other is unaffected
UU	Both parents are unaffected

AGEON first estimates one overall susceptibility intercept, so that the class susceptibilites are equal, and then intercepts for each of the six classes, or for fewer classes if some of the classes are pooled; the corresponding likelihoods are used to perform an (asymptotic) likelihood ratio test that all the class susceptibilities are equal.

4.2.3 Likelihood

The log likelihood maximized is $\sum_{i}^{n} \ln L(i)$, where $L(i)$ is the likelihood for the i-th sib. 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 the last, 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, prior to transformation $a_i + \lambda_2$ cannot strictly follow a normal distribution. This is usually of no consequence but, letting

$$sign(x) = \begin{cases} 1, \text{ if x } \geq 0 \\ -1, \text{ if x} < 0 \end{cases},$$

the maximization can also be 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	$L(i) = \frac{\gamma_i \varphi\left[h(a_i), h(\mu_i), \sigma_i^2\right](a_i+\lambda_2)^{\lambda_1-1}}{\Phi\left[sign(\lambda_1)\left(\frac{h(\mu_i)-h(0)}{\sigma_i}\right)\right]}$
Affected individuals with unknown age of onset	$L^*(i) = \frac{\gamma_i sign(\lambda_1)\left[\Phi\left(\frac{h(a_i')-h(\mu_i)}{\sigma_i}\right)-\Phi\left(\frac{h(0)-h(\mu_i)}{\sigma_i}\right)\right]}{\Phi\left[sign(\lambda_1)\left(\frac{h(\mu_i)-h(0)}{\sigma_i}\right)\right]}$
Unaffected individuals	$L(i) = 1 - L^*(i)$

4.2.4 New Variables

The primary purpose of the AGEON program is to estimate the parameters needed to calculate either of two new quantitative variables 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.

The first quantitative variable, susceptibility to disease conditional on whether the individual i is affected or not by age a_i', called the *susceptibility trait* (Schnell et al., 2012), is given by

$$y_i = \begin{cases} 1 & \text{, if affected} \\ \frac{\gamma_i - \gamma_i \Phi\left[\frac{h\left(a_i'\right)-h(\mu_i)}{\sigma_i}\right]}{1-\gamma_i \Phi\left[\frac{h\left(a_i'\right)-h(\mu_i)}{\sigma_i}\right]} & \text{, if not affected by age } a_i' \end{cases}$$

where Φ is the standard cumulative normal distribution function, λ_1 and λ_2 are the transformation parameters, and γ_i is the susceptability.

The second quantitative variable, the disease age of onset is the *survival analysis residual* given by

$$y_i = \begin{cases} 1 - \gamma_i \Phi\left[\frac{h(a_i)-h(\mu_i)}{\sigma_i}\right] & \text{, if affected at age } a_i \\ -\gamma_i \Phi\left[\frac{h\left(a_i'\right)-h(\mu_i)}{\sigma_i}\right] & \text{, if not affected by age } a_i' \end{cases}$$

where again Φ is the standard cumulative normal distribution function, λ_1 and λ_2 are the transformation parameters, and γ_i is the susceptability.

Using one of these values of y 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).

4.3 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 covariates.

4.3.1 Running `ageon`

A typical run of the AGEON program may use flags to identify the file types like the following:

```
>ageon -p data.par -d data.ped
```

or, rely on a set file order like the following:

```
>ageon data.par data.ped
```

where `data.par` is the name of the parameter file and `data.ped` is the name of the pedigree data file.

4.3.2 The `ageon` Block

An `ageon` block in the parameter file sets the options on how to perform an analysis using AGEON.

The following table shows the syntax for an `ageon` parameter which starts the `ageon` block.

`parameter [, attribute]`	**Explanation**
`ageon`	Starts a AGEON 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: Valid file name in the form of a quoted character string. Default Value: ageon_analysis*n*, where *n* = 1, 2, ..., *k* for a given set of *k* specified AGEON analyses. Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in an `ageon` block.

parameter [, attribute]	Explanation
`title`	Specifies the title of the run. Value Range: Quoted character string. Default Value: AGEON analysis n, where n = 1, 2, ..., k for a given set of k specified AGEON analyses. Required: No Applicable Notes: None
`affectedness`	Specifies name of a binary trait containing affection status. Must be the name of a trait or covariate in the data file or created by means of a function block. Value Range: Character string representing a valid trait name Default Value: None Required: Yes Applicable Notes: None
`age_onset`	Specifies name of a trait containing age of onset for affected individuals. Must be the name of a trait or covariate in the data file or created by means of a function block. Value Range: Character string representing a valid trait name Default Value: None Required: Yes Applicable Notes: 1
`age_exam`	Specifies name of a trait containing age of examination for unaffected individuals. Must be the name of a trait or covariate in the data file or created by means of a function block. Value Range: Character string representing a valid trait name Default Value: None Required: Yes Applicable Notes: 1
`mean_cov`	Starts a sub-block for specifying covariates for the mean age of onset. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

`var_cov`	Starts a sub-block for specifying covariates for the variance of age of onset. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`suscept_cov`	Starts a sub-block for specifying covariates for the trait susceptibility. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`allow_averaging`	Specifies option to substitute covariate mean values for missing covariate data. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2
`transformation`	Starts a sub-block for specifying transformation options. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 3
`pool`	Specifies option to pool classes. Value Range: Quated character string. Default Value: None Required: No Applicable Notes: 4

Notes:

1. It is permissible for the `age_onset` and `age_exam` parameters to specify the same quantitative 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. This can only be done if the age given for an affected person is the age of onset or unknown, i.e. this disallows the possibility of using the information on age at examination of an affected person when age of onset is unknown.

2. If the value of **false** 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 **true** is specified, missing covariate values will be replaced with the covariate's mean value as calculated from the sample used in the analysis.

3. See section 4.2 for details on the transformation theory implemented in this program.

4. The value of `pool` should be one or more algebraic expressions, where each expression refers to two or more default class names (??, ?A, ?U, AA, AU and UU) and equal (=) signs. Expressions should always be enclosed in double quotes (" "), and MUST BE ALL ON ONE LINE.

   Examples:

```
pool = "??=UU"        # Pools the ?? class with the UU class.

pool = "??=?A=AA"     # Pools the ??, ?A and AA classes.

pool = "??=UU, ?A=AU" # Pools ?? with UU, and ?A with AU.
```

4.3.2.1 The `mean_cov` Sub-Block

The following table lists the parameters and attributes that may occur in a `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. A covariate that is specified in this sub-block may not be used in a `var_cov` or `suscept_cov` sub-block.	
	Value Range	Character string representing the name of a trait or covariate from the data file or a name created by means of a `function` block.
	Default Value	None
	Required	No
	Applicable Notes	1

Notes:

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

4.3.2.2 The `var_cov` Sub-Block

The following table lists the parameters and attributes that may occur in a `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. A covariate that is specified in this sub-block may not be used in a `mean_cov` or `suscept_cov` sub-block.	
	Value Range	Character string representing the name of a trait or covariate from the data file or a name created by means of a `function` block.
	Default Value	None
	Required	No
	Applicable Notes	1

Notes

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

   Examples:

```
ageon
{
   title = "analysis"
   affectedness = aff
   age_of_onset = ao
   age_of_exam = ae

   var_cov
   {
      covariate = cov2
   }
}
```

4.3.2.3 The `suscept_cov` Sub-Block

The following table lists the parameters and attributes that may occur in a `suscept_cov` sub-block.

parameter [, attribute]	Explanation	
`covariate`	Covariate to modify the logit of susceptability. This parameter may be specified multiple times. A covariate that is specified in this sub-block may not be used in a `mean_cov` or `var_cov` sub-block.	
	Value Range	Character string representing the name of a trait or covariate from the data file or a name created by means of a `function` block.
	Default Value	None
	Required	No
	Applicable Notes	1

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.

   Examples:

```
ageon
{
   title = "analysis"
   affectedness = aff
   age_of_onset = ao
   age_of_exam = ae
```

```
        suscept_cov
        {
           covariate = cov1
        }
     }
```

4.3.2.4 The `transformation` Sub-Block

The following table lists the parameters and attributes that may occur in a `transformation` sub-block.

parameter [, attribute]	Explanation
lambda1	Specifies the power parameter. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
, val	Specifies the value for lambda1. Value Range: (-∞,∞) Default Value: 1.0 Required: No Applicable Notes: None
, fixed	Specifies option to fix this value. If set to false (the default), the coefficient is estimated. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
lambda2	Specifies the shift parameter. Value Range: (-∞,∞) Default Value: 1.0 Required: No Applicable Notes: None
, val	Specifies the value for lambda2. Value Range: (-∞,∞) Default Value: 0.05 Required: No Applicable Notes: None
, fixed	Specifies option to fix this value. If set to false (the default), the coefficient is estimated. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 1

Notes

1. If this parameter is fixed, it must be > 0.

4.4 Program Output

AGEON produces several output files that contain results and diagnostic information:

File Name	File Type	Description
ageon.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
analysis.sum	Analysis summary output file	Contains the summary table of final estimates of the parameters and their standard errors and other results for the case without truncation.
analysis.det	Analysis detailed output file	Contains the detailed table of final estimates of the parameters and their standard errors and other results for the case with and without truncation.
analysis.ped	Pedigree output file	Contains values for eight new traits (see 4.4.3).
analysis.par	Parameter output file	Contains the pedigree block to be used with above pedigree file (see 4.4.3).

4.4.1 Summary Output File

The summary output file contains descriptive information about each of the six classifications, and results for the case without truncation: final estimates, standard errors, and p-values of the parameters estimated in the model, including

- susceptibility intercept(s) and covariates
- mean intercept and covariates
- variance intercept and covariates
- transformation parameters (λ_1 and λ_2).

The file also includes a likelihood ratio test statistic (under the model assumptions) for the comparison of separate susceptibilities for each category of the classification variable and all the susceptibilities constrained to be equal.

Example:

```
============================
     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               AGEON Analysis 1
  Affectedness trait  AFF
  Age-of-onset trait  AO
  Age-at-exam trait   AE

==============================
     CLASSIFICATION SYSTEM
==============================

  Using default classification system:

  ??  Both parents are unknown.
  ?A  One of the parents is unknown, the other is affected.
  ?U  One of the parents is unknown, the other is unaffected.
  AA  Both parents are affected.
  AU  One of the parents is affected, the other is unaffected.
  UU  Both parents are unaffected.

=========================
     CLASS STATISTICS
=========================

==================
     CLASS ??
==================

  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 UU
==================

  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
                 ??   -0.489289   0.162019   0.002528
                 ?A   -0.489289   0.162019   0.002528
                 ?U   -0.489289   0.162019   0.002528
                 AA   -0.489289   0.162019   0.002528
                 AU   -0.489289   0.162019   0.002528
                 UU   -0.489289   0.162019   0.002528

     Mean intercept   73.682004   0.182158  < 1.00e-07

  Variance intercept    6.308832   1.561898   5.36e-05

  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
                 ??   -0.893122   0.422414   0.034487
                 ?A   -0.295137   0.330399   0.371710
                 ?U   -0.437648   0.322448   0.174696
                 AA   -0.761468   0.647488   0.239581
                 AU   -0.396851   0.305525   0.193974
                 UU   -0.564788   0.451151   0.210612

     Mean intercept   73.683039   0.182786  < 1.00e-07

  Mean covariates
               cov1    0.346353   0.172817   0.045054

  Variance intercept    6.358667   1.578149   5.60e-05

  Variance covariates
               cov2    1.907274   1.214695   0.116376

  Transformation
            Lambda1    1.000000          Fixed
            Lambda2    0.050000          Fixed
-------------------------------------------------------

Final ln likelihood: -382.959654

================================
     Likelihood Ratio 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
```

4.4.2 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 of the log likelihood 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.

Example of additional part of an AGEON detailed output file:

```
=========================================================================
     MAXIMIZATION RESULTS susceptibilities equal, using truncation
=========================================================================


-------------------------------------------------------------------------
  Parameter           Estimate    S.E.       P-value     Deriv
-------------------------------------------------------------------------
  Susceptibility intercepts
                 ??   -0.489288   0.162019   0.002528    0.0000002697
                 ?A   -0.489288   0.162019   0.002528    0.0000000000
                 ?U   -0.489288   0.162019   0.002528    0.0000000000
                 AA   -0.489288   0.162019   0.002528    0.0000000000
                 AU   -0.489288   0.162019   0.002528    0.0000000000
                 UU   -0.489288   0.162019   0.002528    0.0000000000


     Mean intercept   73.682004   0.182158  < 1.00e-07  -0.0000000421


  Mean covariates
               cov1    0.343027   0.172082   0.046219    0.0000001686


  Variance intercept    6.308833   1.561900   5.36e-05    0.0000000027


  Variance covariates
               cov2    1.886915   1.201454   0.116293    0.0000000000


  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
                 ??   -0.893122   0.422414   0.034487   -0.0000000189
                 ?A   -0.295137   0.330399   0.371710   -0.0000000337
                 ?U   -0.437648   0.322448   0.174697    0.0000000000
                 AA   -0.761468   0.647488   0.239581   -0.0000000221
                 AU   -0.396851   0.305525   0.193974    0.0000001686
                 UU   -0.564788   0.451150   0.210612    0.0000000000


     Mean intercept   73.683039   0.182786  < 1.00e-07  -0.0000001938
```

```
  Mean covariates
               cov1    0.346353   0.172817   0.045054   -0.0000000337

  Variance intercept    6.358667   1.578147   5.60e-05    0.0000000000

  Variance covariates
               cov2    1.907274   1.214693   0.116376    0.0000000707

  Transformation
            Lambda1    1.000000                Fixed
            Lambda2    0.050000                Fixed
-------------------------------------------------------------------------


Final ln likelihood: -382.959654


==============================
     Likelihood Ratio Test
==============================
.
.
.
==============================
     Likelihood Ratio Test
==============================


  H0 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       ??          ?A          ?U          AA          AU          UU
-------------------------------------------------------------------------------------------------------------------------------------------
  Mean intercept       0.033182        0.057390           -0.003310   0.032535   0.010835    0.010835    0.010835    0.010835    0.010835    0.010835
  Variance intercept   0.057390        2.439526           -7.59e-05   1.257216   0.041093    0.041093    0.041093    0.041093    0.041093    0.041093
  cov1                -0.003310       -7.59e-05            0.029612   0.006981  -0.000758   -0.000758   -0.000758   -0.000758   -0.000758   -0.000758
  cov2                 0.032535        1.257216            0.006981   1.443488   0.021532    0.021532    0.021532    0.021532    0.021532    0.021532
  ??                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  ?A                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  ?U                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  AA                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  AU                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  UU                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    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       ??          ?A          ?U          AA          AU          UU
-------------------------------------------------------------------------------------------------------------------------------------------
  Mean intercept       0.033182        0.057390           -0.003310   0.032536   0.010835    0.010835    0.010835    0.010835    0.010835    0.010835
  Variance intercept   0.057390        2.439531           -7.59e-05   1.257220   0.041093    0.041093    0.041093    0.041093    0.041093    0.041093
  cov1                -0.003310       -7.59e-05            0.029612   0.006981  -0.000758   -0.000758   -0.000758   -0.000758   -0.000758   -0.000758
  cov2                 0.032536        1.257220            0.006981   1.443491   0.021532    0.021532    0.021532    0.021532    0.021532    0.021532
  ??                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  ?A                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  ?U                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  AA                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  AU                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250
  UU                   0.010835        0.041093           -0.000758   0.021532   0.026250    0.026250    0.026250    0.026250    0.026250    0.026250


=========================================================================
```

```
     VARIANCE-COVARIANCE MATRIX susceptibilities free, using truncation
================================================================================


Error: Matrix is not available.
```

4.4.3 Pedigree and Parameter Output Files

For each of the four models maximized in an AGEON analysis, the two variables (susceptability trait and survival analysis residual) are calculated (for each individual) on the basis of each model's final estimates and are made available for subsequent analysis in an AGEON-generated pedigree file. This file, named for the analysis conducted (such as 'ageon_analysis*n*.ped') is created automatically; the user need not worry about explicitly requesting AGEON to generate the file. This pedigree file contains, for each individual, eight columns with the following names:

1. `nt_equal_trait` - Susceptibility trait for the model with no truncation and susceptibilities equal.
2. `nt_equal_residual` - Survival analysis residual for the model with no truncation and susceptibilities equal.
3. `nt_free_trait` - Susceptibility trait for the model with no truncation and susceptibilities free or as pooled.
4. `nt_free_residual` - Survival analysis residual for the model with no truncation and susceptibilities free or as pooled.
5. `t_equal_trait` - Susceptibility trait for the model with truncation and susceptibilities equal.
6. `t_equal_residual` - Survival analysis residual for the model with truncation and susceptibilities equal.
7. `t_free_trait` - Susceptibility trait for the model with truncation and susceptibilities free or as pooled.
8. `t_free_residual` - Survival analysis residual for the model with truncation and susceptibilities free or as pooled.

In addition to this pedigree file, the parameter file is also created, named for the analysis conducted (such as 'ageon_analysis*n*.par'), containing the corresponding pedigree block that describes the new pedigree file. Using these new files, you can now conduct analyses on the susceptibility traits and survival analysis residuals from any of four AGEON models.

Chapter 5

ASSOC

ASSOC assesses the association between a quantitative or binary trait and one or more covariates (which may include marker phenotypes that have been transformed into quantitative covariates) from extended pedigree data in the presence of familial correlations, simultaneously estimating familial variance components (and hence familial correlations and heritability). 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 a baseline model, as well as those of alternate models that include specified sets of covariates, and performs a likelihood ratio test for the significance of covariates not included in the baseline model. 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. In addition, ASSOC can perform tests that are robust to population stratification (e.g. QTDT) by use of a transmitted allele indicator (see 3.2.7.6) and, using the GUI, independent model residuals produced by ASSOC can be easily imported into the program GMDR to perform a multidimensional reduction analysis with family data.

5.1 Limitations

Pedigrees must not have loops, and a string of mates of length more than three is not allowed; i.e. a person may have multiple mates, but none of the mates may have another mate (e.g: woman-man-woman-man is not allowed). 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 the global maximum. Also, situations can occur in which it is not numerically possible to calculate the variance-covariance matrix of the estimates.

5.2 Theory

5.2.1 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 similar to that described in George and Elston (1987). For individual i, let:

Y_i = a quantitative trait
x_i = a vector of covariates
G_i = a random additive polygenic effect
F_i = a random nuclear family effect
M_i = a random marital effect
S_i = a random sibship effect
E_i = a random individual (environmental and/or measurement error effect)

F_i is an effect common to all members of the same nuclear family; M_i is an effect that spouses share with each other; S_i is an effect that full sibs share with each other (and hence allows for dominance variance and common sibling environmental variance); and E is a person-specific random effect. Note that an individual may belong to several different nuclear families: together with a spouse and children, and/or together with sibs and parents; if a person has children by k different spouses, that person will belong to those k different nuclear families as a parent, and could additionally belong to a family as an offspring with sibs and parents. In these situations the person will have more than one distinct family effect.

Then the default model for a quantitative trait is of the form

$$h(\frac{Y_i - \beta^T x_i}{s}) = G_i + F_i + M_i + S_i + E_i, \tag{5.1}$$

where h is a transformation[1], s is the estimated standard deviation of the residuals that have been minimized prior to transformation, and the polygenic effect (G_i) and each of the random environmental effects (F_i, M_i, S_i, E_i) are assumed to be normally distributed with zero mean. For a quantitatative trait there is also the option to transform "both sides" as in the original model described in George and Elston(1987):

$$h(Y_i) = h\left(\beta^T x_i\right) + G_i + F_i + M_i + S_i + E_i. \tag{5.2}$$

All covariates are mean centered prior to inclusion in the likelihood (see below). If the transformation h is applied to the residuals as in (5.1) the estimates of the parameter values in β are asymptotically unbiased and on the original scale on which on which Y_i is measured; if h is applied to both sides as in (5.2) the estimates are median unbiased. The random effects are assumed to have variances σ^2_G, σ^2_F, σ^2_M, σ^2_S *and* σ^2_E, respectively, and these variances are on the transformed scale.

Thus,

$$V\left[h\left(Y\right)\right] = \sigma^2_G + \sigma^2_F + \sigma^2_M + \sigma^2_S + \sigma^2_E. \tag{5.3}$$

For σ^2_F to be estimable, it is often necessary to have large pedigrees or a large number of pedigrees, or both, and therefore σ^2_F 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, heritability in the case of polygenic variance); it is not possible to estimate any variances to be less than zero (in the implementation, less than .0000001).

More generally, the user can add other variance components by specifying classes of individuals who share common random effects.

[1] See section 5.2.2 for details of the transformation implemented in this program.

For a binary trait that takes on the value 1 or 0, $\beta^T x_i$ is replaced by $\frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$ (so that $\beta^T x_i$ is the logit of Y_i in (5.1)) and the variances in (5.3) are rescaled to sum to 1. No further transformation is allowed in view of the finding by McCulloch and Neuhaus (2011) that in logistic regression the shape of a random effects distribution has little effect on estimating its mean and variance.

5.2.2 Transformation of the Trait

A quantitative trait y may be transformed by:

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

where

$$y_{G2} = \left[\prod_{i=1}^{N} (|y_i+\lambda_2|+1) \right]^{\frac{1}{N}}$$

and N = number of individuals in the sample (possibly including more than one pedigree) with complete trait and covariate values (nothing missing). This is the standardized generalized modulus power transformation (George and Elston, 1988) with power parameter λ_1 and shift parameter λ_2. When this transformation is applied to the standardized residuals as in (5.1), λ_2 is fixed at 0.

5.2.3 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_i.

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.

5.2.4 Estimation of Parameters

Estimation is performed by maximizing the natural log of the likelihood numerically. If several independent pedigrees (including constituent pedigrees) are analyzed jointly, the logarithms of the likelihoods are summed overall pedigrees. The program itself determines initial estimates for the maximizing process. The user, however, may override the initial estimate of any of the parameters, or may fix them at predetermined values.

5.2.5 Models and Sample Formation

An ASSOC analysis consists of a set of models which in turn contain one or more covariates for which a test of significance may be applied. In the simplest form, each analysis contains at least two models: (1) a **baseline model** which includes, at minimum, an intercept and an individual random variance component, but may additionally include other random components and any number of covariates and (2) an **alternate model** which includes the same random components, the intercept, all baseline covariates, plus one or more covariates of interest that we wish to test. More complex models may be specified to contain either multiple versions of alternate models, or models containing multiple test covariates or some combination of the two.

Setting aside the transformation function $h()$, consider a simple model containing an intercept and three predictors, (on the logit scale for a binary trait):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3,$$

and further suppose that in addition to the three fixed effects represented by X_1, X_2 and X_3, there are three additional covariates, A, B and C that we would like to test, either individually or jointly. Then each of the possible combinations of test covariates may be designated by a different alternate model, for example as shown in the table below:

Model Designation	Test Covariate(s)
M1	*A*
M2	*B*
M3	*C*
M4	*A, B*
M5	*A, C*
M6	*B, C*
M7	*A, B, C*

Thus M1 designates a model in which the covariate A is included in the alternate hypothesis, thereby rendering A as the sole test covariate. Similarly, M5 designates a model in which A and C are tested jointly. The ASSOC parameter file syntax for testing all seven combinations would be:

```
assoc {
   trait = Y                          # Dependent trait
   cov   = X1                         # First  baseline predictor
   cov   = X2                         # Second baseline predictor
   cov   = X3                         # Third  baseline predictor
   cov   = A, models = M1, M4, M5, M7  # First  test covariate
   cov   = B, models = M2, M4, M6, M7  # Second test covariate
   cov   = C, models = M3, M5, M6, M7  # Third  test covariate
}
```

The above example would direct ASSOC to perform a likelihood ratio test for of each of the seven different alternate models M1, M2, ..., M7. The LRT for model M1 would test the effect of adding A into the model, the LRT for model M5 would test the effect of adding both A and C into the model, and so on.

The situation is complicated by the fact that, with respect to a given covariate, not all of the observations within a sample may be informative (i.e. because of missing covariate values). Therefore, to use the maximum amount of information for each covariate, the likelihood under the null hypothesis with respect to one covariate may differ from the null likelihood for a different covariate. (This is avoided, and the computation time is shortened, if the user imputes missing values for all covariates. Mean imputation is included as an option.)

An ASSOC analysis begins by constructing from the pedigree data a sample containing only individuals with complete information for the specific analysis to be performed, defined by the model designation. An individual has incomplete information if he/she has a missing value for either the primary trait or any of the covariates specified by either the baseline or the alternate model. If an individual is determined to have incomplete information, ASSOC will treat the individual as uninformative for all data points required by that analysis, although it will retain the relationship information for the analysis. Having constructed this **baseline sample**, ASSOC then finds the maximum likelihoods for the model specified in the analysis: the likelihood for the sample data is calculated once with respect to the model containing only the baseline terms (i.e., the baseline model), and once again for the specified alternate model. A new baseline sample is constructed, if necessary, for each specified alternate model.

ASSOC reports complete maximization information for each null and alternate model and, if desired, a file containing the residuals $Y_i - \beta^T x_i$ after standardization. These residuals are also available after linear transformation to make them quasi-independent (the results are calculated making the assumption that the estimate of β^T is equal to its true value). In addition, it performs a comparison test of the likelihood from each alternate model (H_1) that includes " test" covariates against that of the null (i.e., baseline) model (H_0). If L_1 and L_0 are the maximum likelihoods under H_1 and H_0 respectively, from the same (constructed) sample, then the likelihood ratio statistic is $2[ln(L_1) - ln(L_0)]$. Under the assumption of normality of the residuals and the null hypothesis that the additional covariate(s) have no effect, this statistic is asymptotically distributed as chi-square with the number of degrees of freedom equal to the number of additional 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 (i.e., the Wald test). These p-values are two-sided for the covariate coefficients β and the transformation parameter λ_1; they are one-sided for all variance components. In each case the test is for the null hypothesis that the parameter is 0, except for λ_1, where the null hypothesis is $\lambda_1 = 1$ (no transformation for the default George and Elston transformation). Note: If the transmitted allele indicator (see 3.2.7.6) is used as a covariate, the corresponding test is a test of linkage in the presence of association, or of association in the presence of linkage to the corresponding allele (the latter assumes a valid correlation structure). This represents a transmission disequilibrium type test (TDT) for quantitative traits in extended pedigrees.

5.3 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 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.

5.3.1 Running `assoc`

A typical run of the ASSOC program may use flags to identify the file types like the following:

```
>assoc -p par -d ped -l loc
```

or, rely on a set file order like the following:

```
>assoc par ped loc
```

where `par` is the name of the parameter file, `ped` is the name of the pedigree data file and `loc` is the name of the optional marker locus description file.

5.3.2 The `assoc` Block

An `assoc` block in the parameter file sets the options on how to perform an analysis using ASSOC.

The following table shows the syntax for an `assoc` parameter which starts the `assoc` block.

`parameter [, attribute]`	Explanation	
`assoc` `assoc_analysis,`	Starts an ASSOC 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	Valid file name in the form of a quoted character string.
	Default Value	assoc, if only one analysis is specified; assoc*n*-1 if multiple analysis are specified where *n* = 2, 3, ..., *k* for a given set of *k* specified ASSOC analyses
	Required	No
	Applicable Notes	None

The following table lists the parameters and attributes that may occur in an `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 or covariate from the data file or created by means of a function block.
	Default Value	None
	Required	Yes
	Applicable Notes	None
covariate cov	Specifies a variable in the regression model. It can be a trait name or covariate name.	
	Value Range	The name of a trait or, covariate.
	Default Value	None
	Required	No
	Applicable Notes	2
, models	Specifies in which alternate models the covariate should be included, in the form of a comma-separated list (ie: models="A1, A2, special").	
	Value Range	Comma-separated list of strings
	Default Value	None
	Required	No
	Applicable Notes	3

, val	Specifies the initial estimate for the covariate coefficient. Value Range: (-∞, ∞) Default Value: None Required: Yes if `fixed` equals "true". Otherwise, no Applicable Notes: 4
, fixed	Specifies that the coefficient for this covariate is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
batch	Instructs ASSOC to add a new alternate model for each covariate given in the pedigree block (excluding the primary trait and null model covariates). Value Range: N/A Default Value: N/A Required: No Applicable Notes: 5, 6
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: None
, val	Specifies the initial estimate for this variance component. Value Range: [.0000001,∞) Default Value: None Required: Yes if `fixed` equals "true". Otherwise, no Applicable Notes: 4
, 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: None

, val	Specifies the initial estimate for this variance component. Value Range: [.0000001,∞) Default Value: None Required: Yes if `fixed` equals "true". Otherwise, no Applicable Notes: 4
, fixed	Specifies that this effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
marital_effect se	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: None
, val	Specifies the initial estimate for this variance component. Value Range: 0, [.0000001,∞) Default Value: None Required: Yes if `fixed` equals "true". Otherwise, no Applicable Notes: 4
, fixed	Specifies that the effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
sibling_effect se	Specifies the inclusion of a sibling variance component in the model. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
, val	Specifies the initial estimate for this variance component. Value Range: 0, [.0000001,∞) Default Value: None Required: Yes if `fixed` equals "true". Otherwise, no Applicable Notes: 4

, fixed	Specifies that the effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
class_eff	Specifies the inclusion of a class effect variance component in the model. Value Range: Name of a categorical trait or covariate. Default Value: None Required: No Applicable Notes: 7
, val	Specifies the initial estimate for this variance component. Value Range: 0, [.0000001,∞) Default Value: None Required: Yes if `fixed` equals "true". Otherwise, no Applicable Notes: 4
, fixed	Specifies that the effect is fixed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 4
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: 8
transformation transform trans	Starts a transformation sub-block. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 9
omit_complete_summary	Suppresses a comprehensive list of tests performed, in the summary file. The list specified in the `summary_display` sub-block will still appear. in the summary file. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

`summary_display`	Starts a sub-block for configuring the .sum output file. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`residuals`	Starts a sub-block for specifying creation of .res output files. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

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. This parameter may be repeated multiple times, as needed. 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 `sex_code` has been specified as available to be used as a trait in the pedigree block, as indicated in Section 3.2.5.2.

3. If this attribute is not specified, ASSOC will include the covariate as part of the baseline in all models. There is no need to specify explicitly that a covariate be included in the baseline model. See section on "Models and Sampling Algorithm" in the beginning of this chapter for more details.

4. Treatment of `val` and `fixed` attributes for covariates and variance components is as follows:

   (a) If, for a covariate, `val` is set to 0 and `fixed` is set to **true**, the covariate will be "included" in the model; the effect on the analysis will be to remove all individuals for whom this covariate is missing.

   (b) If, for a variance component (polygenic, family, marital, class effect, etc.), `val` is set to 0 and `fixed` is set to **true**, that random effect will be excluded from the model. This is equivalent to setting the effect to **false** (eg.,`pe = false`,`fe = false, or me = false`).

   (c) If the `fixed` attribute for a covariate or variance component is set to **true**, the attribute `val` must be included.

   (d) If the `fixed` attribute for a covariate or variance component is set to **false** and the attribute `val` is included, this determines the initial value of the variable to be used in the maximization process. However, in the case of a variance component, `val` may not be set to 0 if `fixed` is set to **false**.

   (e) If the `fixed` attribute for a covariate or variance component is set to **false** and the attribute `val` is not included, then the program supplies initial values for the maximization process.

5. Each covariate belongs to one of three categories

(a) *baseline* : if it was listed in the analysis block but not explicitly named as a test covaraite for any model

(b) *test* : if it was listed in the anlysis block and explicitly named as a test covariate for at least one model

(c) *default* : if it was listed in the pedigree block but not listed within the analysis block

The `batch` parameter directs ASSOC to individually test each covariate in categories (b) and (c) while including those belonging to category (a) as non-test covariates within each model analyzed.

6. For an analysis in which one trait and ten covariates are listed in the pedigree block, for example, assuming the user does not explicitly include any single covariate but instead uses the 'batch' parameter, ASSOC will output ten pairs of maximizations (i.e, one baseline and one alternate for each of the covariates specified).

7. The names " Random", " Polygenic", " Family", " Sibling" and " Marital" are reserved for random effects built into the program, and may not be used here. All individuals having the same value for this categorical variable share a common effect. The parameter `class_eff` may be included more than once in an analysis block, in which case the total number of class effect variances estimated will equal the total number of categories in all the categorical variables.

8. 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 individuals fully informative for that analysis). This can greatly improve runtime performance because the baseline model needs to be evaluated only once.

9. Transformation of both sides is not allowed if the primary trait is binary. By default, for a quantitative trait ASSOC will estimate λ_1 using the George-Elston transformation of the standardized difference between the trait and its expected value.

5.3.2.1 The `transformation` Sub-Block, applicable for continuous traits only

The following table lists the parameters and attributes that may occur in a `transformation` sub-block.

`parameter` [, `attribute`]	**Explanation**
`option`	Specifies a particular transformation option. Value Range: {none, george_elston} Default Value: george_elston Required: No Applicable Notes: 1, 2
`lambda1`	Specifies the power parameter Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

, val	Value of the parameter Value Range: (-∞,∞) Default Value: 1.0 Required: No Applicable Notes:
, fixed	Specifies option to fix the value. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
, lower_bound	Specifies inclusive lower bound for power parameter. Value Range: (-∞, 1] Default Value: -∞ Required: No Applicable Notes: 3
, upper_bound	Specifies inclusive upper bound for power parameter. Value Range: [1,∞) Default Value: +∞ Required: No Applicable Notes: None
lambda2	Specifies the shift parameter Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4
, val	Specifies the value of the parameter. Value Range: (-∞,∞) Default Value: 0.0 Required: No Applicable Notes:
, fixed	Option to fix this value. Value Range: {true, false} Default Value: true Required: No Applicable Notes:
both_sides	Specifies that the transformation is to be performed on the dependent trait and its expected value separately. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 5

Notes

1. An `option` value of **none** disables transformation calculations for the analysis, and an `option` value of **george_elston** means that the George-Elston transformation is used. For the **George-**

Elston transformation, it is strongly advised to make all the primary trait values > 1.

2. The default values are $\lambda_1 = 1$ and $\lambda_2 = 0$ for the George and Elston transformation, which (if applied) would give the same result as no transformation PROVIDED that either (1) all the trait values are > 1, or (2) all the trait values are < 1.
3. Theoretically, $\lambda_1 < 0$ can never result in the trait being normally distributed, but in practice it may result in an approximate normal distribution if λ_1 is not too small. If $\lambda_1$is allowed to decrease without bound, it is not uncommon for the other parameter estimates to take on unrealistic values. If this happens, a lower bound (e.g. -1) should be specified.
4. The value of λ_2 is fixed at 0 unless the **both_sides** parameter is specified.
5. If the **both_sides** parameter is not specified, then the default program behavior is to transform the difference between the trait and its expected value.

5.3.2.2 The `summary_display` Sub-Block

The following table lists the parameters and attributes that may occur in a `summary_display` sub-block.

`parameter [, attribute]`	**Explanation**
`order`	Specifies how results are ordered in the .sum output file.
	Value Range: {as_input, lrt, wald, larger_pvalue, pvalue_ratio}
	Default Value: lrt
	Required: No
	Applicable Notes: 1
`filters`	Starts a sub-block for specifying filtering of results to be displayed in .sum output file.
	Value Range: N/A
	Default Value: N/A
	Required: No
	Applicable Notes: None

Notes

1. The meanings of the `order` values are as follows:

`as_input`	Display the results in the same order as the corresponding model names appear in the `assoc_analysis` block in the pedigree file or, if the batch option is chosen, the order in which traits are given in the `pedigree` block of the parameter file followed by any traits created by using `function` blocks, in the order they are specified.
`lrt`	Display in ascending order by likelihood ratio test p-value.
`wald`	Display in ascending order by Wald p value.
`larger_pvalue`	Display in ascending order of the larger of the lrt and Wald p values.
`pvalue_ratio`	Display in descending order by the value of $min(-log_{10}LRT, -log_{10}W) - max(-log_{10}LRT, -log_{10}W)$ where $LRT$is the likelihood ratio test p-value and $W$is the Wald p-value.

5.3.2.3 The `filters` Sub-Block

The following table lists the parameters and attributes that may occur in a `filters` sub-block.

parameter [, attribute]	Explanation
`all`	Specifies that all results be displayed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 1
`lrt`	Specifies that only results with likelihood ratio test p-value less than the `max` value be displayed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2
`, max`	Specifies bound for `lrt` filter. Value Range: (0, 1.0] Default Value: .01 Required: No Applicable Notes: None
`wald`	Specifies that only results with Wald p-value less than the `max` value be displayed. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2
`, max`	Specifies bound for `wald` filter. Value Range: (0, 1.0] Default Value: .01 Required: No Applicable Notes: None
`limit_number`	Specifies that the number of results displayed be limited to the value of `number`. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 2
`, number`	Specifies maximum number of results to display. Value Range: integer in range [0,∞) Default Value: 10 Required: No Applicable Notes: None

Notes:

1. If `all` is **true,** other `filters` parameters do not apply.

2. May be specified in conjunction with other `filters` parameters, in which case the results from the intersection of the filters are displayed .

5.3.2.4 The `residuals` Sub-Block

The following table lists the parameters and attributes that may occur in a `residuals` sub-block.

parameter [, attribute]	Explanation
`model`	Designates a regression model for which residuals and independent residuals are to be written to a file.
	Value Range: Name of a regression model as given in the `models` attribute of the `covariate` parameter.
	Default Value: None
	Required: No
	Applicable Notes: 1
`, null`	Specifies whether to write standardized residuals to a file for the null model.
	Value Range: { true, false }
	Default Value: false
	Required: No
	Applicable Notes: None
`, test`	Specifies whether to write standardized residuals to a file for the alternative model.
	Value Range: { true, false }
	Default Value: true
	Required: No
	Applicable Notes: None

Notes:

1. The value **Baseline** may be used to refer to the implicit model given by the dependent variable and covariates common to all models.

The following are all valid `assoc` statements:

```
assoc_analysis
{
   trait = TRAIT1     # TRAIT1 is the primary trait
   cov   = TRAIT2     # TRAIT2 is a (baseline) covariate
}

assoc_analysis
{
   title = "Analysis, Oct. 8, 2001 "
   trait = TRAIT3                    # TRAIT3 is the primary trait
   cov   = x1, models="A1"           # x1 is a test covariate in the model "A1"
}

assoc_analysis, out=Assoc_res
```

```
{
   trait = TRAIT3              # TRAIT3 is the primary trait
   cov   = X1, models="A1"     # X1 is a test covariate in model "A1"
   cov   = X2, models="A1, A2" # X2 is a test covariate in models A1 and A2
   cov   = X3                  # X3 is a baseline covariate, included in all models
   cov   = TRAIT1              # TRAIT1 is a baseline covariate, included in all models
}
```

5.4 Program Output

ASSOC produces several output files that contain results and diagnostic information:

File Name	File Type	Description
analysis.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
analysis.sum	ASSOC calculation summary output	Contains tables of the models analyzed with Wald and LRT p-values.
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.
analysis.tsv	ASSOC tab-delimited results	Each record in this file contains a model name, an LRT p-value and a Wald p-value.
analysis_model_null or analysis_model_test.res	ASSOC tab-delimited residuals	Each record in this file contains a pedigree id, individual id, parent1, parent2, sex and estimate of the residual and the quasi-independent residual for that individual.

5.4.1 Summary Output File

The Summary Output File contains a model comparison table in which models containing one and only one additional covariate are compared against the null model. The table is filtered and sorted as specified in the `summary_display` sub-block. A table showing all models analyzed is also shown if **omit_complete_summary** is not included in the `assoc` block. Each record in either table contains the model name, the intercept estimate, estimates of the covariate coefficients, standard error for the test covariate estimate, Wald p-value and LRT p-value.

Example:

```
=================
     Results
=================
==============================
Analysis description
==============================
Title                          assoc
Primary Trait                  sbpd_rand10 (Quantitative)
Allow averaging                Disabled
Omit complete summary table    Disabled
Summary table display order    By LRT p-value

Summary filter                 Show no more than 10 results



====================================
Transformation configuration
====================================

Note: No transformation applied.



======================
```

```
Summary table

======================

-------------------------------------------------------------------------------------------------------------------------------
   Model                Intercept    sex_code   Test cov.              Estimate    Std. err.   P-value (Wald)  P-value (LRT)
 -------------------------------------------------------------------------------------------------------------------------------
   foo_selection          131.180992  -5.400113  foo_selection          17.693939    1.486697   2.23e-308       1.72e-28
   foo_age                131.180995  -5.368967  foo_age                 0.427976    0.066267   1.06e-10        2.78e-10
   foo_k_u24c             131.133838  -3.722359  foo_k_u24c              0.093116    0.043523   0.032397        0.032863
   foo_na_u24c            131.133772  -3.781233  foo_na_u24c             0.032032    0.015451   0.038158        0.038646
   foo_bmi                131.180994  -5.012986  foo_bmi                 0.309041    0.163769   0.059153        0.059677
   PEDIGREE_SIZE          131.180995  -4.166253  PEDIGREE_SIZE          -0.312730    0.224757   0.164099        0.164559
   foo_PEDIGREE_SIZE      131.180995  -4.166253  foo_PEDIGREE_SIZE      -0.312730    0.224757   0.164099        0.164559
   foo_FOUNDER_INDICATOR  131.180995  -4.354616  foo_FOUNDER_INDICATOR   4.811372    3.783636   0.203506        0.203921
   FOUNDER_INDICATOR      131.180995  -4.354645  FOUNDER_INDICATOR       4.811575    3.783628   0.203486        0.203921
   FAMILIAL_INDICATOR     131.181105  -4.327102  FAMILIAL_INDICATOR      0.305280   12.312859   0.980220        0.980219
 -------------------------------------------------------------------------------------------------------------------------------
Note: Models with an asterisk (*) may not have converged.
.
.
.
```

5.4.2 Detailed Output File

The detailed output includes:

1. A sample description
2. Variance components on the transformed scale.
   - σ_G^2−Polygenic variance
   - σ_E^2−Random variance
   - σ_F^2−Familial variance
   - σ_M^2−Marital variance
   - σ_S^2−Sibship variance
   - ** - any class effect variances, where each label ** indicates a category name
3. Coefficients
   - β_0− Intercept
   - β_j−Covariate coefficients, $j > 0$
4. Total variance:$V[h(Y_i)]$
5. " Heritability":$\sigma_G^2/V[h(Y_i)]$
6. Residual familial correlations (based on non-zero variance components)
   - Full Sibs $(\sigma_F^2 + \sigma_S^2 + \frac{1}{2}\sigma_G^2)/\{V[h(Y_i)]\}$
   - Half Sibs $(\sigma_F^2 + \frac{1}{4}\sigma_G^2)/\{V[h(Y_i)]\}$

- Parent-Offspring $\sigma_F^2 + \frac{1}{2}\sigma_G^2 / \{V[h(Y_i)]\}$
- Marital (spouse): $(\sigma_F^2 + \sigma_M^2) / \{V[h(Y_i)]\}$

7. Environmental intraclass correlations (based on non-zero variance components)

   - Nuclear Family: $\sigma_F^2 / \left\{V[h(Y_i)] - \sigma_G^2\right\}$
   - Marital (spouse): $(\sigma_F^2 + \sigma_M^2) / \left\{V[h(Y_i)] - \sigma_G^2\right\}$
   - Full sibs (spouse): $(\sigma_F^2 + \sigma_S^2) / \left\{V[h(Y_i)] - \sigma_G^2\right\}$

8. Transformation paramters

   - λ_1 −Lambda 1
   - λ_2 −Lambda 2

9. The estimated variance-covariance matrix of all the estimated parameters

10. The partial first derivative of the natural logarithm of the likelihood with respect to each of the parameters estimated

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, σ_S^2 and class effect variance components use a 1-sided test. All other p-values use 2-sided tests.

Example:

```
.
.
.
============================
     Model 'foo_na_u24c'
============================
============================
Sample description
============================
Number of individuals in dataset                                604
Number of constituent pedigrees in dataset                       73

Number of singletons in dataset                                   3
Number of constituent pedigree members with complete information   423

Number of singletons with complete information                    2



Covariates
====================

----------------------------------------------------------------

Name         Mean         Std. dev.    Min        Max
----------------------------------------------------------------
sex_code        0.555294    0.496933   0.000000     1.000000
foo_na_u24c   106.973341   54.015420   4.130000   329.110000

================================================================
MAXIMIZATION RESULTS foo_na_u24c without test covariates
================================================================
----------------------------------------------------------------------

Parameter       Estimate     S.E.        P-value     Deriv
```

```
-------------------------------------------------------------------------
Variance components
       Random   297.732700   20.424277  < 1.00e-07  -0.0000000304
Other parameters
Total variance   297.732700   20.424277  < 1.00e-07   0.0000000309
       Intercept   131.133772    0.836987  < 1.00e-07   0.0000000000
Covariates
       sex_code    -4.010422    1.684304   0.017263    0.0000000000
 -------------------------------------------------------------------------
Final ln likelihood: -1813.490721
=========================================================================

VARIANCE-COVARIANCE MATRIX foo_na_u24c without test covariates
=========================================================================
-------------------------------------------------------------------------
Random       Total variance  Intercept  sex_code
-------------------------------------------------------------------------
Random           417.151076   417.151076     -0.000155   0.000000
Total variance   417.151076   417.151076     -0.000155   0.000000
Intercept         -0.000155    -0.000155      0.700548   0.000000
sex_code           0.000000     0.000000      0.000000   2.836879
.
.
.
```

Chapter 6

DECIPHER

DECIPHER obtains, for different types of analysis units, maximum likelihood estimates of frequencies of all possible haplotypes for autosomal or X-linked markers. The analysis units may be random individuals from the population, an individual representative of a constituent pedigree, the set of founders in a constituent pedigree, or a pooled DNA sample. In the case of members of constituent pedigrees, genotypes of other pedigree members are used to infer phase for ambiguous individuals, which improves the haplotype frequency estimates over those obtained using unrelated individuals. Haplotype frequencies can be estimated separately from different groups of analysis units that are specified by the user. A likelihood ratio test and a permutation test are provided to compare haplotype frequency distributions amoung groups.

Decipher automatically removes from analyisis markers with no data or only a single allele . In addition, the user may specify that markers with minor allele frequency less than a specified value be removed.

It can determine marker blocks by either the four gamete rule or linkage disequalibrium. The user may also specify that haplotype domains be defined by a sliding window

6.1 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 is missing, or when a large number of markers (more than 25) is haplotyped. Finally, markers in the haplotype region must be codominant, and family information and pools may not be used in the case of X-linked markers.

6.2 Theory

6.2.1 Haplotype Frequency Estimation

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. 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 et al. 2003). The key feature is 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 as pools of one individual; sets of founders in a constituent pedigree can be considered as pools of f individuals, where f is the number of founders in the pedigree. To allow combinations of the data types and to allow variation in the number of founders per pedigree, we have extended the usual EM algorithm to the situation where there are different numbers of individuals in each unit.

To estimate population haplotype frequencies, each analysis unit must come from a random set of individuals or pedigrees. For pedigree data the user can specify analysis units of single individuals, all founders or single representatives for each constituent pedigree. To choose single representatives as the analysis unit, the user may designate a single representative from each constituent pedigree or, if no individual is indicated for a particular constituent pedigree, the program will randomly select one individual out of those individuals in the pedigree with the most marker genotypes available (i.e., we are assuming 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) 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 sum of these two numbers is $2k$ because 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 the i-th pool and the $j-th$ column, $M_{.j}$, represents the j-th SNP, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Each entry, M_{ij}, is an integer representing the number of copies of a particular allele in pool i at SNP j. The value of each entry is thus an integer in $[0, 2k]$. For m diallelic markers, there is a 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 corresponding 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 $2k$ haplotypes for the k individuals in pool i. Under the assumption of Hardy-Weinberg proportions, and assuming that all the individuals are independent, the likelihood for the proportions given the data can be expressed as

$$P(M,F) = \prod_{i=1}^{n} \sum_{\Delta \in H_i} P(\Delta). \tag{6.1}$$

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 different starting points that will be used. DECIPHER will obtain maximum likelihood estimates for each of this number of randomly selected starting points, and the set of estimates corresponding to the largest 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 6.1 above.

There are several types of information that can be obtained. First, haplotype frequencies can be estimated for specified sets of individuals or pools. The user has the option of partitioning the individuals or pools into groups representing different subpopulations (e.g., case-control groups, ethnic groups, etc) and obtaining haplotype frequencies separately for each group. Second, we can obtain a list of all possible non-recombinant combinations of haplotypes for each individual or pool (with the constraint of < 30 markers). Third, we can obtain a list of the most likely combinations of haplotypes for each individual or pool, together with the posterior probability of each, based on population data. Because these lists can be quite large, particularly when there is a large number of markers and/or alleles, separate thresholds can be specified for displaying the lists of haplotypes and most likely haplotype combinations. Only haplotypes with an estimated frequency, or haplotype combinations with a posterior probability, greater than these respective thresholds will then be displayed. In the case where only the most likely haplotype combinations are requested, more than one haplotype combination will be returned if they have the same (maximum) posterior probability.

6.2.2 Likelihood Ratio Test

A likelihood ratio test is available to compare the distribution of haplotypes between groups (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 j'. 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 different haplotypes in 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 group assignment (e.g., case-control status), and recomputing the LR test statistic for each permutation. On the assumption of exchangability, the empirical p-value is determined from the sample permutations as the number of permutations for which the LR test statistic exceeds the observed LR test statistic, divided by the total number of permutations.

6.2.3 Haplotype Block Determination

6.2.3.1 Four Gamete Rule

Haplotype block determination may be done by the four gamete rule as described in Wang et. al. 2002. The four gamete rule applies only if all markers in the region of interest are diallelic. A recombination between two diallelic markers is inferred by the four gamete rule if the frequency estimates of all four possible haplotypes formed by those markers exceed some user supplied threshold. These recombinations can be used to determine haplotype blocks as follows.

Haplotype frequencies for the first two markers in the region are estimated using the EM algorithm. The four gamete rule is applied to these frequency estimates using the user supplied threshold. If a recombination is found, the first marker is discarded, and the process begins again with the second and subsequent markers. If, however, no recombination is found, the first and second markers form the beginning of a block, and the procedure continues by successively pairing the third marker with the first and second markers. If a recombination is found as a result of either of these pairings, the first and second markers constitute a block, and the search for the next block begins with the third and fourth markers. Otherwise, the third marker is added to the incipient block, and the fourth marker is paired with each of the first three. The process continues in this fashion until the end of the region is reached. Blocks consisting of only one marker are discarded.

6.2.3.2 Linkage Disequilibrium

Linkage disequilibrium may be used to determine blocks by sequentially computing D' (Lewontin's LD measure) between pairs of consecutive markers. If D' exceeds the threshold for the first two markers, they constitute the beginning of a block. If D' between the second and third exceeds the threshold, the third marker becomes part of the block. This process continues until a D' is found that does not exceed the threshold, at which point the current block ends, and the process continues. When a D' is found that exceeds the threshold, a new block is begun.

6.3 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.
Marker locus description file (optional)	Lists the allele frequencies and phenotype to genotype mapping for each marker locus. This file is irrelevant and need not be given if the parameter `analysis_unit` in the `data` sub-block of the parameter file equals **pool**. It is required only for the filtering of markers by minor allele frequency and marker block determination by linkage disequilibrium. See 6.3.2.
Genome description file (optional)	Contains a description of the linked marker regions This file is irrelevant and need not be given if the parameter `analysis_unit` in the `data` sub-block of the parameter file equals **pool**. Otherwise, it may be used to specify regions for analysis. See 6.3.2.

6.3.1 Running `decipher`

A typical run of the DECIPHER program may use flags to identify the file types like the following:

```
>decipher -p par -d ped -l loc -g gen
```

or, rely on a set file order like the following:

```
>decipher par ped loc
```

where `par` is the name of the parameter file, `ped` is the name of the pedigree data file, `loc` is the name of the locus description file and `gen` is the name of the genome description file. Note that the locus description and genome description files are optional.

6.3.2 The `decipher` Block

A `decipher` block in the parameter file sets the options on how to perform an analysis using DECIPHER.

The following table shows the syntax for a `decipher` parameter which starts the `decipher` block.

parameter [, attribute]	Explanation
decipher	Starts a DECIPHER 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: decipher_analysis*k* where $k = 1, 2 \ldots n$ for n analysis Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `decipher` block.

parameter [, attribute]	Explanation
title	Specifies the title of the analysis Value Range: Character string Default Value: Analysis*k* where $k = 1, 2 \ldots n$ for n analysis Required: No Applicable Notes: None
region	Specifies the name of the chromosomal region to be analyzed. Value Range: Character string naming a region. Default Value: None Required: Yes Applicable Notes: 1, 2, 3
, first	Names a marker in the data file that is to be taken as the first marker in the region. Value Range: Character string Default Value: None Required: See notes 2 and 3. Applicable Notes: 2, 3
, last	Names a marker in the data file that is to be taken as the last marker in the region. Value Range: Character string Default Value: None Required: See notes 2 and 3. Applicable Notes: 2, 3

`epsilon`	Specifies the maximum 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 the number of randomly chosen starting points for which the EM algorithm is to run. Value Range: {1, 2, 3, ...} Default Value: 10 Required: No Applicable Notes: None
`dump`	Option to write haplotype frequencies (and the log-likelihood) for each set of EM algorithm starting points to an output file. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 4
`, 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: 4
`filters`	Starts a sub-block to specify marker filtering options. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`blocks`	Starts a sub-block to specify how to determine haplotype blocks. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

`data`	Starts a sub-block to specify 1. how to treat relatedness of individuals, 2. individuals to represent families, 3. individuals to represent groups and 4. how pooled data are represented.
	Value Range: N/A
	Default Value: N/A
	Required: See note 5.
	Applicable Notes: 5
`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. May be specified more than once.
2. If the attributes `first` and `last` are not specified, the `region` parameter is used to name a region in the genome description file. Markers to be analyzed and their order are then as specified in the genome description file and the marker order in the data files is ignored. Distances between the markers are ignored by DECIPHER.
3. If the attributes `first` and `last` are specified, the `region` parameter is used to assign a name to the region whose first marker is given by the attribute `first`, and whose last marker is given by the attribute `last` (The genome description file is ignored in this case). Markers analyzed are given by the range of markers named by `first` and `last`. If the data file uses character delimited records, markers named by the `marker` parameter in the `pedigree` block of the parameter file come first, in the order they are given, followed by markers specified by the `marker_list` parameter, again in the order given, unless a marker locus description file is given, in which case the marker order in that file is the marker order used in the analysis. If the data file uses column delimited records, the marker order is as given in the parameter file `pedigree` block unless a marker locus description file is given, in which case the marker order in that file is the marker order used in the analysis. The `region` parameter does not apply if `data` sub-block parameter, `analysis_unit`, equals **pool**.
4. Applicable only if `pop_freq` or `most_likely_combinations` in the tasks sub-block is set to **true**. Starting points shown in the dump file output are generated by choosing random phase probabilities for each pool and then calculating the haplotype frequencies. To display all haplotype frequency estimates, specify `cutoff` to be 0.
5. This sub-block is required when the data are pooled.

6.3.2.1 The `filters` Sub-Block

The following table lists the parameters and attributes that may occur in a `filters` sub-block.

parameter [, attribute]	Explanation
maf_filter	Specifies removal of markers based on their minor allele frequency. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 1, 2, 3
, threshold	Specifies the frequency below which a marker is removed from the analysis. Value Range: [0, .5) Default Value: **.1** Required: No Applicable Notes: None

Notes:

1. This parameter does not apply if `the data` sub-block parameter `analysis_unit` equals **pool**.
2. This parameter only applies if markers are diallelic.
3. The user must supply a marker locus description file to use this feature.

6.3.2.2 The `blocks` Sub-Block

The following table lists the parameters and attributes that may occur in a `blocks` sub-block.

parameter [, attribute]	Explanation
sliding_window	Specifies use of a sliding window. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 1, 2
, width	Specifies the width in number of markers of the sliding window. Value Range: {2, 3, 4 ...} Default Value: 3 Required: No Applicable Notes: None
four_gamete_rule	Determine haplotype blocks using the four gamete rule, and perform the specified analysis for each block. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 2, 3

, threshold	Specifies the frequency threshold used in applying the four gamete rule. Value Range: [0, .25) Default Value: .01 Required: No Applicable Notes: 4
ld	Determine haplotype blocks by applying linkage disequilibrium threshold to pairs of adjacent markers and perform the specified analysis for each block. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 2, 3, 5
, threshold	Specifies the frequency threshold used in determining haplotype blocks via linkage disequilibrium. Value Range: (0, 1) Default Value: .8 Required: No Applicable Notes: None

Notes:

1. If value is **true**, for each region the specified analysis is performed on a haplotype domain consisting of the first marker in the region and the next n - 1 markers, where n is the value of the width attribute. The analysis is repeated with a haplotype domain consisting of the second marker and the next n - 1 markers in the region. This process is continlued until the window consists of the last n markers in the region.
2. This parameter does not apply if `the data` sub-block parameter `analysis_unit` equals **pool**.
3. This parameter only applies if markers are diallelic.
4. The `threshold` attribute of the `four_gamete_rule` parameter may not be less than ten times the EM algorithm convergence criteriun (specified with the `epsilon` parameter).
5. The user must supply a marker locus description file to use this feature.

6.3.2.3 The `data` Sub-Block

The following table lists the parameters and attributes that may occur in a `data` sub-block.

parameter [, attribute]	Explanation
analysis_unit	Specifies unit to be used in determining possible haplotype combinations in the estimation of haplotype frequencies. Value Range: { each_individual, family_rep, family_founders, pool} Default Value: family_founders Required: No Applicable Notes: 1
family_rep	Variable used to specify one genotyped individual per constituent subpedigree when analysis_unit equals family_rep. Value Range: Character string representing the name of a trait, covariate or string field listed in the data file. Default Value: None Required: No Applicable Notes: 2, 3
, 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 value to be matched in the designated family_rep field. Default Value: None Required: Yes if family_rep is specified. Otherwise, no. Applicable Notes: 4
partition	Starts a sub-block for specifying groups of the data set representing different subpopulations. Value Range: Character string representing the name of a trait, covariate or string field listed in the data file or created by means of a function block. The named variable will be used as the basis of classification for the created partition. Default Value: None Required: Yes if likelihood_ratio_test = true. Otherwise, no. Applicable Notes: 3, 5, 6

pools	Starts a sub-block for mapping traits to allele frequencies.	
	Value Range	N/A
	Default Value	N/A
	Required	See note 7.
	Applicable Notes	7

Notes:

1. For the analysis_unit parameter:
   (a) If this parameter is set to **each_individual**, all individuals will be used in the estimation of haplotype frequencies, and they will be assumed to be independent.
   (b) If this parameter is set to **family_rep**, one genotyped person per constituent pedigree will be used, and familial information will be considered in determining possible haplotype combinations (in this case diplotypes) for that person. Singletons will be treated as if the parameter were set to **each_individual**.
   (c) If this parameter is set to **family_founders**, the set of founders in a constituent pedigree will be considered as a group in determining possible haplotype combinations. If there are no genotyped founders in a constituent pedigree, or if partition information among founders is inconsistent (all founder partition values must be valid and all must have the same values to be considered consistent, see 6.3.2.3.1), the constituent pedigree is treated as if the analysis_unit parameter were set to **family_rep** without a family_rep parameter being specified. Singletons are treated as if the parameter were set to **each_individual**.
   (d) If this parameter is set to **pool**, each record in the data file is treated as a pool of genetic material as specified in the pools sub-block.
   (e) If the value of analysis_unit is **family_rep** or **family_founders** 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 from among those with the most genotyped loci in the haplotype region.
3. The same trait or covariate may not be used as a value for both the family_rep and partition parameters.
4. The program looks at the values of each individual in the constituent pedigree for the family_rep variable. If no individual matches the family_rep_value, then the constituent pedigree is not used in the analysis. If only one matches the family_rep_value, the person with that value is the family representative (person whose haplotype combinations are used in the analysis). If more than one individual in a constituent pedigree has this value, the program will arbitrarily pick from among the designated individuals a genotyped individual with the most genotyped loci in the haplotype region to be the family representative.
5. 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. If this sub-block is not specified, all

analysis units will be treated as coming from a single population. All analysis units having the same value for this variable belong to the same subpopulation (group). When the value of `analysis_unit` is **family_rep**, the family representative is determined first and that representative's `partition` values are used. If no representative with valid subpopulation (group) values can be found, the constituent pedigree is skipped.

6. If the value of `analysis_unit` is **family_founders**, partition values of the constituent pedigree founders must be consistent (as defined in note 1) or the constituent pedigree will be skipped. The order in which the partitions are listed is significant. See note 4 of the `tasks` sub-block for details.

7. This sub-block is required if `analysis_unit` is **pools**.

6.3.2.3.1 The `partition` Sub-Block

The following table lists the parameters and attributes that may occur in a `partition` sub-block.

parameter [, attribute]	Explanation	
`sub_pop`	Specifies name of group (subpopulation).	
	Value Range	Character string
	Default Value	None
	Required	Yes if `likelihood_ratio_test` = true. Otherwise, no.
	Applicable Notes	1, 3
`, sub_pop_value`	Specifies value of the partition variable common to all individuals in this group (subpopulation).	
	Value Range	Character string representing the value to be matched in the designated `partition` variable.
	Default Value	None
	Required	Yes if `sub_pop` is specified. Otherwise, no.
	Applicable Notes	2

Notes:

1. If a value is not given for this parameter, the group (subpopulation) name is the same as `sub_pop_value`. This parameter may be repeated as needed but `sub_pop` and `sub_pop_value` must be unique within a partition. If no valid groups (subpopulations) are specified, then every distinct value of the partition variable found in the data file (except the missing value), will designate a group (subpopulation).

2. The missing value code for the `partition` variable may not be specified as a `sub_pop_value`.

3. This parameter may be repeated as necessary.

6.3.2.3.2 The `pools` Sub-Block

The following table lists the parameters and attributes that may occur in a `pools` sub-block.

parameter [, attribute]	Explanation
`pool_size`	Specifies the ploidy multiplied by the number of individuals per pool . Value Range: {1, 2, 3, ...} Default Value: 2 Required: No Applicable Notes: 1
`pool_size_trait`	Variable used to specify pool sizes (ploidy X number of individuals per pool) on a per pool basis. Value Range: Character string representing the name of a quantitative trait or covariate field listed in the data file or created by means of a function block. Default Value: None Required: No Applicable Notes: 2
`locus`	Starts a sub-block for specifying loci and the variables that are used to specify their allele probabilities. Value Range: Character string representing the locus name. Default Value: None Required: Yes if `analysis_unit` = pools, there must be a locus sub-block for each locus in the haplotype region. Otherwise, no. Applicable Notes: 3

Notes:

1. This parameter designates the pool size for all pools for which `pool_size_trait` is not specified.
2. For a given record, if a value for `pool_size_trait` is given, it takes precedence over `pool_size`. Values are rounded to the nearest integer.
3. At least two `locus` sub-blocks must be given. A Genome Description File is not required or used when the value of the `data` sub-block parameter,`analysis_unit`, is **pool**. Map order is assumed to be the order in which the `locus` sub-blocks are given.

6.3.2.3.2.1 The `locus` Sub-Block

The following table lists the parameters and attributes that may occur in a `locus` sub-block..

parameter [, attribute]	Explanation	
allele	Specifies an allele name.	
	Value Range	Character string representing the name of an allele in the locus.
	Default Value	None
	Required	Yes if analysis_unit = pools. Otherwise, no.
	Applicable Notes	1
, trait	Variable used to specify allele probabilities.	
	Value Range	Character string representing the name of a quantitative trait or covariate field listed in the data file.
	Default Value	None
	Required	Yes if analysis_unit = pools. Otherwise, no.
	Applicable Notes	2
last_allele	Specifies the name of the last allele in a locus.	
	Value Range	Character string representing the last allele in the locus.
	Default Value	None
	Required	Yes if analysis_unit = pools. Otherwise, no.
	Applicable Notes	3

Notes:

1. At least one allele parameter must be included for each locus sub-block.
2. The value of this variable is interpreted as the pool allele probability for the allele named by the allele parameter.
3. last_allele must not have a trait associated with it. Its probability is one minus the sum of the probabilities of the other alleles in the locus.

6.3.2.4 The tasks Sub-Block

The following table lists the parameters and attributes that may occur in a tasks sub-block.

parameter [, attribute]	Explanation
pop_freq	Specifies option to estimate population 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_combinations_table	Specifies option to display haplotype combinations for each analysis unit in tabular form. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 2
most_likely_combinations	Specifies option to display the most likely haplotype combinations for each analysis unit. Value Range: { true, false} Default Value: false Required: No Applicable Notes: None
, cutoff	Specifies minimum haplotype combination posterior probability threshold value for display. If none of the posterior probabilities meet or exceed the specified value, then the haplotype combination with the greatest posterior probability is displayed. Value Range: [0, 1] Default Value: 0.05 Required: No Applicable Notes: 3
likelihood_ratio_test	Specifies option to perform likelihood ratio test. Value Range: { true, false} Default Value: false Required: No Applicable Notes: 4

`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 in this sub-block. 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 the 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 the width interval 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. A haplotype combination is a group of haplotypes that are consistent with the genotypes of the `analysis_unit` over the haplotyping region. If the value of `analysis_unit` is **each_individual** or **family_rep**, a haplotype combination is synonymous with diplotype, but if `analysis_unit` is **family_founders** or **pool**, there may be more than two haplotypes to a haplotype combination.

3. To display all haplotype combination posterior probabilities, specify a `cutoff` of 0.

4. If two `partition` sub-blocks are specified, designate the first partition listed as *Partition 1* , and the second as *Partition 2* . Likelihood ratio tests are performed across the subpopulations (groups) of Partition 1 for each of the subpopulations (groups) in Partition 2, i.e., *Partition 1* is the " *inner* " partition and *Partition 2* is the " *outer* " partition. At least two subpopulations (groups) must be specified in the first `partition` sub-block to do this test.

The following are all valid decipher analysis blocks:

```
decipher
{
  region = "chrom 1"
}

# In this next example the partition parameter is used to indicate case / control
# status.
decipher
{
   region = Chr12          # Quotes not required since the region name
                           # does not contain spaces
   data
   {
      analysis_unit = each_individual   # Do not use family information in determining
                                        # possible haplotypes.

      partition = affection_status
      {
          sub_pop = my_cases, sub_pop_value = 1
          sub_pop = my_controls, sub_pop_value = 0
      }
   }
}


# In following example, the partition parameter is used twice to partition on both
# case / control status and ethnic group.
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
   {
      analysis_unit = family_rep
      family_rep = T1,            # Values for this trait designate one genotyped individual
                                  # per family whose possible haplotype combinations are
                                  # use in the analysis.
      family_rep_value = 1        # Haplotypes to be determined for genotyped individuals
                                  # whose value for trait, T1, equals 1.
      partition = affection_status
      {
          sub_pop = my_cases, sub_pop_value = 1
          sub_pop = my_controls, sub_pop_value = 0
      }
```

```
      partition = ethnicity
      {
          sub_pop = "african american", sub_pop_value = 1
          sub_pop = caucasian, sub_pop_value = 0
      }
   }

   tasks
   {
      pop_freq = true, cutoff = .1    # Show only haplotype frequency estimates greater
                                      # than or equal to 0.1.
      likelihood_ratio_test = true
      compute_empirical_pvalue = true, permutations = 1000
   }
}

# This example illustrates the use of pooled DNA.
decipher, out=analysis1
{
  epsilon = .000001

  data
  {
    analysis_unit=pool

    pools
    {
      pool_size = 4   # 4 haplotypes (2 persons) per pool.

      locus = M1      # 1st locus of the haplotype region.
      {
        allele = A, trait = T1   # Variable T1 of pedigree data contains probabilities for allele A.
                                 # allele A.
        last_allele = a          # probability of allele a is 1 - value of T1.
      }

      locus = M2      # 2nd locus of the haplotype region.
      {
        allele = A, trait = T2
        last_allele = a
      }
    }
  }

  tasks
  {
    pop_freq = false
    all_possible_combinations_table = true
    most_likely_combinations = true, cutoff = .0001
    likelihood_ratio_test = false
  }
}
```

6.4 Program Output

DECIPHER produces four types of output files that contain results and diagnostic information:

File Name	File Type	Description
decipher.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
decipher.sum	Summary output file	Contains population haplotype frequency estimates.
decipher.det	Detailed output file	Contains possible haplotype combinations and most likely haplotype combinations for analysis units.
decipher.dmp	Data dump file	Contains haplotype frequency estimates and ln likelihoods for each set of starting points for which the EM algorithm is run.

6.4.1 Summary Output File

Contains results pertaining to the whole data set, specifically haplotype frequency estimates, likelihood ratio test results and empirical p-values.

Example:

```
==========
Analysis 1
==========
Options Selected
================
Haplotype region                                  Region 1
Minor allele frequency filter                     no
Sliding window                                    no
Four gamete rule blocks                           no
LD blocks                                         no
EM algorithm convergence criterion                1e-05
Number of EM algorithm starting states            10
  Dump                                            no
Subpopulations specified                          no
Analysis unit                                     each individual
Estimate haplotype frequencies                    yes
  Cutoff                                          0.0001
Show all possible haplotype combinations table    no
List most likely haplotype combinations           yes
  Cutoff                                          0.0001
Do likelihood ratio test                          no
Compute empirical p-value                         no
 Results
=======
Markers in order:
```

```
  M1 M2 M3 M4 M5 M6 M7 M8 M9
                        Haplotype Frequency Estimates
Note:  Haplotypes listed have estimated frequencies greater than or equal to
       the cutoff or have the greatest frequency estimate.
Haplotype                  Frequency
---------                  ---------
1-1-2-1-1-2-1-1-2          0.181216
1-1-1-2-1-2-2-1-2          0.178580
1-1-2-2-1-2-2-2-2          0.176715
1-2-1-1-1-2-2-2-1          0.176348
1-2-2-2-1-1-1-2-2          0.169943
1-1-2-1-1-2-1-1-1          0.00383315
1-2-1-1-2-1-2-2-1          0.00340491
1-1-1-2-1-1-2-2-2          0.00308505
1-2-2-2-2-1-1-1-2          0.00307154
2-1-1-2-2-2-1-2-2          0.00249970
1-1-1-2-2-1-2-2-1          0.00249657
2-1-1-1-2-2-2-2-1          0.00242523
2-1-2-1-1-2-1-1-2          0.00234241
1-1-2-2-1-1-1-1-1          0.00202179
.
.
.
2-1-2-1-2-1-2-2-1          0.000104205
                           ---------
Total                      0.999876
Ln likelihood    -3980.05
```

6.4.2 Detailed Output File

Contains results on an analysis unit basis, specifically possible and most likely haplotype combinations. Note: in the following output, "haplotype combination" refers to a group of haplotypes that are consistent with the genotype of an `analysis_unit`. In the following example, the term is synonymous with diplotype.

Example:

```
==========
Analysis 1
==========

Markers in order:
  M1 M2 M3 M4 M5 M6 M7 M8 M9


                    Most Likely Haplotype Combinations

Note:  Haplotype combinations listed have estimated probabilities greater than or equal
       to the cutoff or have the greatest probability estimate.
Pedigree     Member     Combination                                Probability
--------     ------     -----------                                -----------
1            1          1-1-2-1-1-2-1-1-2  1-2-2-2-1-1-1-2-2       1.00000
10           10         1-1-1-2-1-2-2-1-2  1-1-2-1-1-2-1-1-2       1.00000
100          100        1-1-2-2-1-2-2-2-2  1-2-1-1-1-2-2-2-1       1.00000
```

```
1000        1000      1-1-2-2-1-2-2-2-2  1-2-1-2-1-1-2-1-2      0.511122
                      1-1-1-2-1-2-2-1-2  1-2-2-2-1-1-2-2-2      0.488878
101         101       1-1-1-2-1-2-2-1-2  1-1-2-2-1-2-2-2-2      1.00000
102         102       1-1-1-2-1-2-2-1-2  1-2-1-1-1-2-2-2-1      1.00000
103         103       1-1-1-1-1-1-2-2-2  1-1-2-1-1-2-1-1-2      1.00000
104         104       1-1-1-2-1-2-2-1-2  1-2-1-1-1-2-2-2-1      1.00000
105         105       1-2-1-1-1-2-2-2-1  1-2-2-2-1-1-1-2-2      0.999993
106         106       1-1-2-1-1-2-1-1-2  1-1-2-2-1-2-2-2-2      1.00000
107         107       1-2-1-1-1-2-2-2-1  1-2-2-2-1-1-1-2-2      0.999993
108         108       1-2-1-1-1-2-2-2-1  2-1-1-2-1-1-2-1-1      1.000000
109         109       1-1-2-2-1-2-2-2-2  1-2-1-1-1-2-2-2-1      1.00000
11          11        1-1-2-1-1-2-1-1-2  2-1-1-1-2-2-2-2-1      0.997526
                      1-1-2-1-2-2-1-1-2  2-1-1-1-1-2-2-2-1      0.00247398
110         110       1-1-2-2-1-2-2-2-2  1-2-2-2-1-1-1-2-2      1.00000
111         111       1-1-2-1-1-2-1-1-2  1-1-2-1-1-2-1-1-2      1.00000
112         112       1-2-1-1-1-2-2-2-1  2-2-1-2-1-2-1-2-2      1.00000
.
.
.
```

Chapter 7

FCOR

FCOR can estimate multivariate familial correlations, and their asymptotic standard errors, without any distributional assumptions other than the existence of first and second moments. This can be done for all pair types available in a set of pedigrees with no marriage rings or loops. 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 (ignoring sex) 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.

7.1 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.

7.2 Theory

The theory underlying all the calculations performed by FCOR is given in Keen and Elston (2003) and Matthew et al (2011).

7.2.1 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. In other words, cross-correlations are not calculated for any pairs that do not have data for a common trait.

We call relative pair types that depend on individuals' sexes *subtypes*, and those that do not are called *main types*.

7.2.2 Correlations

Consider the N pairs of the observations of a particular type or subtype 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

$$r_{xy} = \frac{\sum_{i=1}^{N} w_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} w_i (x_i - \bar{x})^2 \sum_{i=1}^{N} w_i (y_i - \bar{y})^2}} \tag{7.1}$$

where $\bar{x} = \sum_i w_i x_i \big/ \sum_i w_i$ and $\bar{y} = \sum_i w_i y_i \big/ \sum_i w_i$ for arbitrary non-negative weights $\{w_i\}$. These weights are chosen to minimize the variance of the correlation being estimated, but with the restriction that using them to form confidence intervals leads to confidence intervals with the appropriate coverage (see Matthew et al 2011).

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, or trait. 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 unspecified (rather than being left at its default value of 2), FCOR will examine the pedigree structure and then decide for itself what pedigree

correlations can 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).

7.2.3 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.

7.2.4 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

$$EquivalentCount = \frac{1}{2}\left[N_0 + \sqrt{N_0^2 + \frac{22(1-r^2)}{s^2}r^2}\right], \tag{7.2}$$

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

7.2.5 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 recalulating the correlations ignoring sex. 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.

7.2.6 P-values for Correlations

Let r denote the estimate of the correlation ρ. P-values for testing $\rho = 0$ are based on Fisher's z-transformation $z(r) = \frac{1}{2} log_e \left(\frac{1+r}{1-r}\right)$ and the equivalent count n. For large samples, z is normally distributed with mean $\frac{1}{2} log_e \left(\frac{1+\rho}{1-\rho}\right)$, and variance $\frac{1}{n-3}$ for an interclass correlation and $\frac{1}{n-\frac{3}{2}}$ for intraclass correlation. For this purpose, all cross-correlations are taken to be interclass.

7.3 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, and trait data.

7.3.1 Running `fcor`

A typical run of the FCOR program may use flags to identify the file types like the following:

```
>fcor -p data.par -d data.ped
```

or, rely on a set file order like the following:

```
>fcor data.par data.ped
```

where `data.par` is the name of the parameter file and `data.ped` is the name of the pedigree data file.

7.3.2 The `fcor` Block

An `fcor` block in the parameter file sets the options on how to perform an analysis using FCOR.

The following table shows the syntax for a `fcor` parameter which starts the `fcor` block.

parameter [, attribute]	Explanation
`fcor`	Starts a FCOR 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: fcor Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in an `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
`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: 2
`conservative`	Specifies calculation of asymptotic standard errors conservatively. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 3
`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: 4
`output_options`	Specifies a sub-block of various output option parameters for the current analysis. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`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: 5

, all , each	Option to specify whether the test should be performed for *each* trait seperately, or for *all* traits jointly. Value Range: {all, each} Default Value: None Required: No Applicable Notes: None
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: N/A Required: No Applicable Notes: 6
, single	Prints a single matrix for each trait specified. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 6
, joint	Prints a joint matrix for each pair of traits specified. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 6

Notes

1. The value of a `trait` parameter should be set to the name of a trait or covariate field either read from the 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 then considers all specified traits jointly.

2. The `type` parameter is used to specify whether to calculate correlations for relative subtypes only, 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**.

3. 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 parameter `conservative` specifies that if any required correlation is nonestimable, then that standard error is not calculated. The default value for `conservative` is **false**.

4. 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.

5. 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 default value of `homogeneity_test` is **false**.

6. 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 the matrix for each trait one trait at a time, and the `joint` attribute is used to print a single joint matrix for two traits (see7.4.5). 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. The amount of computation and output can be limited by using a `var_cov` parameter block.

7.3.2.1 The `output_option` Sub-Block

The following table lists the parameters and attributes that may occur in an `output_option` sub-block.

`parameter [, attribute]`	**Explanation**
`detailed_out`	Specifies option to produce an additional output file of detailed information. Value Range: {true, false} Default Value: false Required: No Applicable Notes: None
`tabular_out`	Specifies option to produce an additional output file of alternate tabular structure for all correlation types. Value Range: {true, false} Default Value: false Required: No Applicable Notes: See 7.4.3
`pairs_out`	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: {true, false} Default Value: false Required: No Applicable Notes: See 7.4.4
`sex_name`	Option to print out the name of the relationship with the sequence of ancestors in the lineage. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 1

Notes

1. If the value of `sex_name` is set to **false**, then non-sex specific names will be printed in output tables. Non-sex specific name for a pair is 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 separated by a comma. 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 separated by two commas. The default value is **true**.

7.3.2.2 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 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 table of codes below. Default Value: None Required: No Applicable Notes: 2

Notes

1. The value of a `trait` parameter should be set to the name of 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
		fm,fm	female-cousin-through-father:male-cousin-through-mother

Main Types		Subtypes	
Code	Name	Code	Name
		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

The following are all valid `fcor` statements:

```
fcor
{
}

fcor, out=test
{
   trait=TRAIT1
   trait=TRAIT2
   trait=TRAIT3
   type=maintypes
   standard_error=true
   sex_name=false
   conservative=true
   homogeneity_test=true
   generation_limit=3

   output_options
   {
      detailed_output=true
      tabular_out=true
      pairs_out=true
   }

   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
   }
}
```

7.4 Program Output

FCOR produces several output files that contain results and diagnostic information:

File Name	File Type	Description
fcor.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
fcor.out	Analysis output file	Contains tables of correlations, their standard errors, used pair counts and equivalent pair counts for each pair of traits for each subtype and/or main type of relative (see 7.4.1).
fcor.det	Detailed output file	Contains detailed tables of correlations and standard errors with different weights, used pair counts and equivalent pair counts, for each pair of traits for each subtypes and/or main type of relative (see 7.4.2).
fcor.alt	Alternative tabular format file	Contains tables of correlations, their standard errors, used pair counts and equivalent pair counts in the alternate tabular form (see 7.4.3).
fcor.pair	The smallest number of pair count file	Contains tables of the smallest number of pairs used to calculate any of the required correlations for each standard error (see 7.4.4).
fcor.cov	Variance-covariance matrix output file	Contains the variance-covariance matrix or matrices of correlation estimates (see 7.4.5).

7.4.1 Analysis Output File

The FCOR main analysis output file contains tables of correlations, their standard errors, used pair counts and equivalent pair counts for each pair of traits for each subtype and/or main type of relative up to 2nd generation (by default) or the generation specified by `generation_limit`. Addtionally, it contains pooled cross-correlations of interclass relative types and homogeneity test result when there are more than 1 trait. When `homogeneity_test` value is true, it also contains chi-square values, degrees of freedom and p-values.

Example:

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

    Number of pedigrees : 207
    Number of traits    : 2

    Trait(s) : TRAIT1 TRAIT2
```

```
    Legend :
      ------  : Value is not estimable.
      &&&&&&  : Pair count 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 at least one nonestimable required
                correlation to a value of 0.

================================================================================
.
.
.
================================================================================
Main Relationship Type : parent:offspring
================================================================================
Subtypes Pooled   : father : son
                    mother : son
                    father : daughter
                    mother : daughter
Total Pairs Found = 913
--------------------------------------------------------------------------------
                          TRAIT1                          TRAIT2
 INTERCLASS  ---------------------------------------------------------------
              Count  Correlation  P-value     Count  Correlation  P-value
             EqvCnt   +/- StdErr             EqvCnt   +/- StdErr
--------------------------------------------------------------------------------
   TRAIT1      913         0.1851  0.0000 ***  913         0.1446  0.0003 ***
               645.2   +/- 0.0381              658.1   +/- 0.0382
--------------------------------------------------------------------------------
   TRAIT2      913         0.1402  0.0061 **   913         0.0906  0.0141 *
               380.0   +/- 0.0504              733.5   +/- 0.0366
--------------------------------------------------------------------------------
 Pooled Cross-Correlations
 -------------------------
   TRAIT1                                      913         0.1431  0.0003 ***
                                               636.8   +/- 0.0389
--------------------------------------------------------------------------------
 Test for Homogeneity of cross-correlations
 ------------------------------------------
  Chi-Square = 0.007990 with 1 degree(s) of freedom
  P-Value    = 0.928776
--------------------------------------------------------------------------------



Relationship Type : father(Row):son(Column)
      Pairs Found = 241
--------------------------------------------------------------------------------
                          TRAIT1                          TRAIT2
 INTERCLASS  ---------------------------------------------------------------
              Count  Correlation  P-value     Count  Correlation  P-value
             EqvCnt   +/- StdErr             EqvCnt   +/- StdErr
--------------------------------------------------------------------------------
   TRAIT1      241        -0.0011  0.9884      241         0.0355  0.6525
               166.4   +/- 0.0777              163.4   +/- 0.0784
--------------------------------------------------------------------------------
   TRAIT2      241        -0.0494  0.5299      241         0.0195  0.8047
               164.4   +/- 0.0780              163.1   +/- 0.0785
--------------------------------------------------------------------------------
Pooled Cross-Correlations
 -------------------------
   TRAIT1                                      241        -0.0089  0.8942
                                               224.2   +/- 0.0669
--------------------------------------------------------------------------------
 Test for Homogeneity of cross-correlations
 ------------------------------------------
  Chi-Square = 1.016663 with 1 degree(s) of freedom
  P-Value    = 0.313312
```

```
--------------------------------------------------------------------------
.
.
.
--------------------------------------------------------------------------
Test for Homogeneity of Correlations among Subtypes - All Traits
--------------------------------------------------------------------------
  Chi-Square = 23.48062 with 12 degree(s) of freedom
  P-Value    = 0.023912
.
.
.
```

7.4.2 Detailed Output File

The detailed output file contains detailed tables of correlations and standard errors with different weights, used pair counts and equivalent pair counts, for each pair of traits for each subtypes and/or main type of relative up to 2nd generation (by default) or up to the generation specified by `generation_limit`. Generated when `detailed_out` value in `output_option` sub-block is true.

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

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


    Number of pedigrees : 207
    Number of traits    : 2


    Trait(s) : TRAIT1 TRAIT2


    Legend :
      ------  : Value is not estimable.
      &&&&&&  : Pair count 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 at least one nonestimable required
                correlation to a value of 0.

==========================================================================
.
.
.
==========================================================================
Main Relationship Type : parent:offspring
==========================================================================
Subtypes Pooled   : father : son
                    mother : son
                    father : daughter
                    mother : daughter
Total Pairs Found = 913
--------------------------------------------------------------------------
                          TRAIT1                          TRAIT2
 INTERCLASS  ---------------------------------------------------------------
               Count  Correlation  P-value     Count  Correlation  P-value
              EqvCnt   +/- StdErr             EqvCnt   +/- StdErr
--------------------------------------------------------------------------
 PAIR_WISE WEIGHT
 ----------------
   TRAIT1      913         0.1851  0.0000 ***  913         0.1446  0.0003 ***
               593.1   +/- 0.0397              606.6   +/- 0.0398
--------------------------------------------------------------------------
```

```
   TRAIT2      913         0.1169  0.0026 **   913         0.0906  0.0187 *
               658.4   +/- 0.0385              673.1   +/- 0.0383
-------------------------------------------------------------------------------
 UNIFORM WEIGHT
 --------------
   TRAIT1      913         0.1991  0.0001 ***  913         0.1528  0.0021 ***
               369.6   +/- 0.0500              401.0   +/- 0.0488
-------------------------------------------------------------------------------
   TRAIT2      913         0.1402  0.0061 **   913         0.1011  0.0403 *
               380.0   +/- 0.0504              411.6   +/- 0.0488
-------------------------------------------------------------------------------
 AVERAGE of PAIR_WISE & UNIFORM
 ------------------------------
   TRAIT1      913         0.1874  0.0000 ***  913         0.1459  0.0003 ***
               597.8   +/- 0.0395              618.3   +/- 0.0394
-------------------------------------------------------------------------------
   TRAIT2      913         0.1209  0.0019 **   913         0.0924  0.0163 *
               653.9   +/- 0.0386              673.1   +/- 0.0382
-------------------------------------------------------------------------------
.
.
.
```

7.4.3 Output File of the Alternate Tabular Form

This file contains tables of correlations, their standard errors, used pair counts and equivalent pair counts in the alternate tabular form. It is generated when `tabular_out` value in `output_option` sub-block is set to true.

Here is a typical example of the optional additional tabular form of output:

```
Relationship Type : father:son
      Pairs Found = 241
-----------------------------------------------------------------------
                  Count  Correlation  EqvCnt  StdError  P-values
-----------------------------------------------------------------------
 TRAIT1 - TRAIT1    241   0.0496812    190.9  0.072387  0.495490 *
 TRAIT1 - TRAIT2    241   0.0443138    168.6  0.077100  0.568280
 TRAIT2 - TRAIT1    241   0.0258354    190.8  0.072536  0.723237
 TRAIT2 - TRAIT2    241   0.0295229    168.6  0.077172  0.703905
-----------------------------------------------------------------------
```

7.4.4 Output File of the Smallest Number of Pairs

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. It is generated when `pairs_out` value in `output_option` sub-block is set to true

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 Sub/Maintypes
=========================================================================

    Number of pedigrees : 207
    Number of traits    : 2

    Trait(s) : TRAIT1 TRAIT2
```

```
    [      ] : Excluded the number of pairs for nonestimable required
               correlations.

================================================================================
.
.
.
Relationship Type : father(Row):son(Column)
      Pairs Found = 241
-------------------------------------
 INTERCLASS     TRAIT1      TRAIT2
-------------------------------------
   TRAIT1           178         178
-------------------------------------
   TRAIT2           178         178
-------------------------------------
.
.
.
```

7.4.5 Variance-Covariance Matrix Output File

This file contains the variance-covariance matrix or matrices of correlation estimates. It is generated when there is a `var_cov` sub-block within the `fcor` block.

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 at least one nonestimable
         required correlation to a value of 0.
================================================

  Legend :
    [Row1] PARENT:OFFSPRING - TRAIT1:TRAIT1
    [Col1] PARENT:OFFSPRING - TRAIT1:TRAIT1
  ---------------------
     \       [Col1]
  ---------------------
  [Row1]    0.0052217
  ---------------------

  The Smallest Number of Pairs Used in
    Calculating Required Correlations
  ---------------------
     \       [Col1]
  ---------------------
  [Row1]           86
  ---------------------

  Legend :
    [Row1] PARENT:OFFSPRING - TRAIT2:TRAIT2
```

```
  [Col1] PARENT:OFFSPRING - TRAIT2:TRAIT2
---------------------
   \       [Col1]
---------------------
[Row1]  [ 0.0062607]
---------------------

The Smallest Number of Pairs Used in
  Calculating Required Correlations
---------------------
   \       [Col1]
---------------------
[Row1]           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 at least one nonestimable
         required correlation to a value of 0.
================================================


  Legend :
    [Row1] FATHER:SON - TRAIT1:TRAIT1
    [Row2] FATHER:SON - TRAIT1:TRAIT2
    [Row3] FATHER:SON - TRAIT2:TRAIT1
    [Row4] FATHER:SON - TRAIT2:TRAIT2
    [Col1] MOTHER:SON - TRAIT1:TRAIT1
    [Col2] MOTHER:SON - TRAIT1:TRAIT2
    [Col3] MOTHER:SON - TRAIT2:TRAIT1
    [Col4] MOTHER:SON - TRAIT2:TRAIT2
  ------------------------------------------------------------
     \       [Col1]       [Col2]       [Col3]       [Col4]
  ------------------------------------------------------------
  [Row1]   -0.0004526   -0.0000822   -0.0001792   -0.0000288
  [Row2]   -0.0000854   -0.0004590    0.0000230   -0.0001697
  [Row3]    0.0003632    0.0001635    0.0000327    0.0001361
  [Row4]    0.0000577    0.0002804    0.0001000    0.0001008
  ------------------------------------------------------------

  The Smallest Number of Pairs Used in
    Calculating Required Correlations
  ------------------------------------------------------------
     \       [Col1]       [Col2]       [Col3]       [Col4]
  ------------------------------------------------------------
  [Row1]          178          178          178          178
  [Row2]          178          178          178          178
  [Row3]          178          178          178          178
  [Row4]          178          178          178          178
  ------------------------------------------------------------

  Legend :
    [Row1] FATHER:SON - TRAIT1:TRAIT1
    [Row2] FATHER:SON - TRAIT1:TRAIT3
```

```
  [Row3] FATHER:SON - TRAIT3:TRAIT1
  [Row4] FATHER:SON - TRAIT3:TRAIT3
  [Col1] MOTHER:SON - TRAIT1:TRAIT1
  [Col2] MOTHER:SON - TRAIT1:TRAIT3
  [Col3] MOTHER:SON - TRAIT3:TRAIT1
  [Col4] MOTHER:SON - TRAIT3:TRAIT3
-------------------------------------------------------------
   \       [Col1]       [Col2]       [Col3]       [Col4]
-------------------------------------------------------------
[Row1]   -0.0004526    0.0000009   -0.0000811   -0.0000411
[Row2]    0.0000646   -0.0005089    0.0000746   -0.0000916
[Row3]    0.0003006    0.0001168   -0.0003247    0.0002001
[Row4]   -0.0000465    0.0003295   -0.0000248   -0.0002056
-------------------------------------------------------------

The Smallest Number of Pairs Used in
  Calculating Required Correlations
-------------------------------------------------------------
   \       [Col1]       [Col2]       [Col3]       [Col4]
-------------------------------------------------------------
[Row1]          178          178          178          178
[Row2]          178          178          178          178
[Row3]          178          178          178          178
[Row4]          178          178          178          178
-------------------------------------------------------------

Legend :
  [Row1] FATHER:SON - TRAIT2:TRAIT2
  [Row2] FATHER:SON - TRAIT2:TRAIT3
  [Row3] FATHER:SON - TRAIT3:TRAIT2
  [Row4] FATHER:SON - TRAIT3:TRAIT3
  [Col1] MOTHER:SON - TRAIT2:TRAIT2
  [Col2] MOTHER:SON - TRAIT2:TRAIT3
  [Col3] MOTHER:SON - TRAIT3:TRAIT2
  [Col4] MOTHER:SON - TRAIT3:TRAIT3
-------------------------------------------------------------
   \       [Col1]       [Col2]       [Col3]       [Col4]
-------------------------------------------------------------
[Row1]    0.0001008   -0.0003228   -0.0006320    0.0000243
[Row2]    0.0001108   -0.0000552    0.0005476   -0.0005515
[Row3]    0.0003562    0.0005819   -0.0003420   -0.0002770
[Row4]    0.0000523    0.0005094   -0.0000148   -0.0002056
-------------------------------------------------------------

The Smallest Number of Pairs Used in
  Calculating any Required Correlations
-------------------------------------------------------------
   \       [Col1]       [Col2]       [Col3]       [Col4]
-------------------------------------------------------------
[Row1]          178          178          178          178
[Row2]          178          178          178          178
[Row3]          178          178          178          178
[Row4]          178          178          178          178
-------------------------------------------------------------
```

Chapter 8

FREQ

FREQ is a program that estimates allele frequencies and marker-specific inbreeding coefficients from marker data among related individuals with known pedigree structure and, for codominant markers, generates marker locus description files, needed by GENIBD, MLOD, and other S.A.G.E. programs.

8.1 Limitations

Maximum likelihood estimates of allele frequencies and inbreeding coefficients 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 estimates increases greatly with the number of alleles at any locus.

8.2 Theory

8.2.1 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. 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 because much of the data are not used and, in many datasets, the 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.

8.2.2 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 (unless an inbreeding coefficient is estimated, see 8.2.3 below) 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 (Boehnke 1991). 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.

8.2.3 Inbreeding Coefficient

FREQ also estimates, optionally, an inbreeding coefficient f for each marker. It is calculated by assuming, for alleles i and j with frequencies x_i and x_j, the founder genotype frequencies $x_i^2 + x_i(1-x_i)f$ for genotype ii, $2x_ix_j(1-f)$ for genotype ij and $x_j^2 + x_j(1-x_j)f$ for genotype jj. When the inbreeding coefficient is estimated, only maximum likelihood estimates are calculated, for both the allelle frequenceies and the marker-specific inbreeding coefficient. Standard errors are calculated by numerical double differentiation of the log likelihood.

8.3 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.
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.

8.3.1 Running `freq`

A typical run of the FREQ program may use flags to identify the file types like the following:

```
>freq -p data.par -d data.ped -l m.loc
```

or, rely on a set file order like the following:

```
>freq data.par data.ped m.loc
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, and `m.loc` is the name of the marker locus description file.

8.3.2 The `freq` Block

A `freq` block in the parameter file sets the options on how to perform an analysis using FREQ.

The following table shows the syntax for a `freq` parameter which starts the `freq` block.

parameter [, attribute]	Explanation
`freq`	Starts a FREQ 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: freq Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `freq` 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
`inbreeding`	Enables estimation of inbreeding coefficient. Value Range: {true, false} Default Value: None Required: No Applicable Notes: 3

Notes

1. The value of this parameter is the weight given to estimates from only the founder and singleton data. It 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.

3. If inbreeding is enabled, FREQ will perform two maximum likelihood estimations. The first will exclude the inbreeding coefficient; the second will include it. In addition, the output will include a two-sided likelihood ratio test of the null hypothesis $f = 0$.

8.4 Program Output

FREQ produces several output files that contain results and diagnostic information:

File Name	File Type	Description
freq.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
freq.sum	Summary output file	Contains summary information of analysis options and results.
freq.det	Detailed output file	Contains detailed information of analysis options and results.
freq.loc freq.loc2	Locus description file	Two output files that present the allele frequencies estimated by FREQ in forms that may be read directly into any other S.A.G.E. program that requires it. While the original locus description file with the .loc extension just includes the allele frequency estimates under no-inbreeding, the newly formatted locus description file with the extension .loc2 also includes the allele and genotype frequency estimates under inbreeding if `inbreeding=true`. Therefore, by using the .loc2 file instead of the .loc file as input to S.A.G.E. linkage programs, linkage analysis is performed without assuming marker genotype frequencies under HWE proportions. Note the .loc2 file must be used for X-linkage analysis in LODLINK.

8.4.1 Summary Output File

The summary output file contains the following information:

- Analysis configuration
- Allele frequencies for each requested marker, including initial counts (codominant only)

Example:

```
======================

    Freq Analysis

======================
```

```
======================
     Configuration
======================
  Output file:                freq
  Founder weight:             Disabled
  Max. likelihood estimation: Enabled
  Inbreeding coefficient:     Not estimated

Note: A '*' by an MLE column indicates that the corresponding maximization may not
have converged. Please review the detailed output file for more information.

================================
     Allele frequencies 'm1'
================================

-----------------------------------------------------
  Allele  Founders only  Entire dataset  MLE
-----------------------------------------------------
  1        0.000000       0.687500        0.684802
  2        1.000000       0.312500        0.315198

================================
     Allele frequencies 'm2'
================================

-----------------------------------------------------
  Allele  Founders only  Entire dataset  MLE
-----------------------------------------------------
  1        0.500000       0.666667        0.607625
  2        0.500000       0.333333        0.392375
```

8.4.2 Detailed Output File

The detailed output file contains everything in the summary output file, plus complete maximization information (standard errors, final log likelihoods, derivatives) and, if an inbreeding coefficient f was estimated, a two-sided likelihood ratio test of the null hypothesis $f = 0$.

Example:

```
======================


     Freq Analysis


======================


======================
     Configuration
======================
  Output file:                freq
  Founder weight:             Disabled
  Max. likelihood estimation: Enabled
  Inbreeding coefficient:     Not estimated

Note: A '*' by an MLE column indicates that the corresponding maximization may not
have converged. Please review the detailed output file for more information.

================================
     Allele frequencies 'm1'
================================
```

```
-----------------------------------------------------
  Allele  Founders only  Entire dataset  MLE
-----------------------------------------------------
  1        0.000000       0.687500        0.684802
  2        1.000000       0.312500        0.315198

=================================
     MAXIMIZATION RESULTS m1
=================================


-------------------------------------------------------------
  Parameter  Estimate   S.E.       P-value  Deriv
-------------------------------------------------------------
  Alleles
          1   0.684802   0.187457           -0.0000004061
          2   0.315198   0.187457           -0.0000008824
-------------------------------------------------------------

Final Log likelihood: -4.677551

=================================
     Allele frequencies 'm2'
=================================


-----------------------------------------------------
  Allele  Founders only  Entire dataset  MLE
-----------------------------------------------------
  1        0.500000       0.666667        0.607625
  2        0.500000       0.333333        0.392375

=================================
     MAXIMIZATION RESULTS m2
=================================


-------------------------------------------------------------
  Parameter  Estimate   S.E.       P-value  Deriv
-------------------------------------------------------------
  Alleles
          1   0.607625   0.178317           -0.0000009799
          2   0.392375   0.178317           -0.0000004529
-------------------------------------------------------------

Final Log likelihood: -7.920528
```

8.4.3 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.

Chapter 9

GENIBD

GENIBD is a program for generating identity-by-descent (IBD) sharing distributions from genetic marker locus data on family structures 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 first cousin. 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. These files can then be read into other programs (e.g. SIBPAL) for analyses.

9.1 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.

Each constituent pedigree is treated as an independent pedigree. There are three methods currently implemented that generate IBD sharing distributions. Each method has distinct capabilities and limitations:

9.1.1 Single Marker IBD Analysis

The Single Marker IBD Analysis uses complete pedigree information at each marker individually 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.

9.1.2 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. 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.

9.1.3 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 by putting in 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 should be only used when pedigrees are too large for the exact IBD analysis.

9.2 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

$$\hat{f}_{imj} = \frac{P(I_m \,|\, pair\ j\ shares\ i\ alleles\ IBD)\, P(pair\ j\ shares\ i\ alleles\ IBD)}{L(I_m)} \tag{9.1}$$

or

$$\hat{f}_{imj} = \frac{P(I_m,\, pair\ j\ shares\ i\ alleles\ IBD)}{L(I_m)} \tag{9.2}$$

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 sibs 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 shared 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.

9.2.1 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 9.1, while for other pair types we use equation 9.2. For sib pairs, we use the *counting* method suggested by Amos, Dawson and Elston (1990). To evaluate equation 9.2 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

$$P(I_m, pair\ j\ shares\ i\ alleles\ IBD) = \sum_{g \in G} P(I_m, pair\ j\ shares\ i\ alleles\ IBD, g),$$

where G is the set of all possible genotype configurations of the individuals in the conditioned set. We therefore calculate $P(I_m, pair\ j\ shares\ i\ alleles\ IBD, g)$ for each possible genotype configuration g 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.

9.2.2 Exact IBD Analysis

The exact IBD analysis is used for both single- and multi-point analysis. It uses the exact multipoint 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.

9.2.2.1 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.

9.2.2.2 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 9.2 above), we obtain the probability of the pair sharing that number of alleles IBD.

9.2.2.3 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. Although distances between markers may be specified in Haldane or Kosambi cM units (see 3.4), once these have been translated into recombination fractions calculations proceed under the assumption of no interference.

9.2.3 Simulation IBD Analysis

The simulation IBD analysis uses a modified 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, again 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.

9.2.3.1 Calculating the Amount of Simulation

By default, GENIBD itself 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.

9.3 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.
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-marker (two-point) linkage analysis.

9.3.1 Running `genibd`

A typical run of the GENIBD program may use flags to identify the file types like the following:

```
>genibd -p data.par -d data.ped -l m.loc -g g.map
```

or, rely on a set file order like the following:

```
>genibd data.par data.ped m.loc g.map
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, `m.loc` is the name of the marker locus description file, and `g.map` is the name of the genome description file.

9.3.2 The `genibd` Block

A `genibd` block in the parameter file sets the options on how to perform an analysis using GENIBD.

The following table shows the syntax for a `genibd` parameter which starts the `genibd` block.

parameter [, attribute]	Explanation
genibd	Starts a GENIBD 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	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 GENIBD analyses.
	Required	No
	Applicable Notes	None

Notes

1. 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.

The following table lists the parameters and attributes that may occur in a genibd 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 GENIBD analyses.
	Required	No
	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 data file) are skipped.	
	Value Range	Character string.
	Default Value	None
	Required	No
	Applicable Notes	None
output_pair_types	Specifies pair types to be generate.	
	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 one 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 are used to calculate the IBD sharing at each point, assuming no interference.
3. The `simulation` sub-block allows simulation on pedigrees for which the value of $2n - f$ (see 9.1.2) exceedes the value of `max_peigree`. 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
   }
}
```

9.3.2.1 The `simulation` Sub-Block

The following table lists the parameters and attributes that may occur in a `simulation` sub-block.

parameter [, attribute]	Explanation
`random_seed`	Specifies the seed value for the random simulation. Value Range: {1, 2, 3, ...} Required: No Applicable Notes: 1
`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: 2
`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: 3

`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
`dememfactor`	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: 3
`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: 3
`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: 3
`sim_steps`	The number of simulation steps during each batch.
	Value Range: {1, 2, 3, ...}
	Default Value: 200000
	Required: No
	Applicable Notes: 3
`dememsteps`	The number of dememorization steps during each batch.
	Value Range: {1, 2, 3, ...}
	Default Value: 50000
	Required: No
	Applicable Notes: 3

batch_count	The number of batches of simulation to perform.	
	Value Range	{1, 2, 3, ...}
	Default Value	100
	Required	No
	Applicable Notes	3

Notes

1. If not specified, different seeds, and hence different results, will be obtained each time a given analysis is performed.

2. Transitions proposed during the simulation process comprise changes to the inheritance state at specific markers or sets of markers. The parameter `sim_local_marker` specifies the probability that the next marker proposed for alteration is adjacent to the marker chosen immediately prior to the current proposal; otherwise a marker is chosen at random. This increases the probability of a compatible set of alterations being proposed, decreasing the time to convergence. Setting this value too high can cause a reduction in the coverage of the space due to only local changes being proposed. A lower value can result in better coverage, at the expense of time to convergence, since transitions of lower probability transitions will be proposed. The default value of 0.75 has been chosen based upon extensive in-house testing and should be sufficient for most data sets.

3. 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 by default 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 `demem_factor`, `sim_factor` and `sim_batch_factor` as described in the syntax table) or may set the values of `demem_factor`, `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 `demem_steps`, `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.

9.4 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. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
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_*n.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 3.5).

9.4.1 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.

9.4.2 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.

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 3.5):

- 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) and 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 the following example, a multi-point IBD sharing file is generated. Although the numerical results are different, the single-point file is similar in structure. The following is a portion of the 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 10

LODLINK

LODLINK performs model-based lod score calculations for two-point linkage between a main trait/marker and each of the other markers in the pedigree file. The main trait/marker may be a marker or a trait that follows Mendelian transmission and has either two or three types. When a trait is used (as opposed to a marker) as the main trait/marker, 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 relatively fast exact linkage calculations.

10.1 Limitations

Pedigrees may not contain loops or marriage rings. **Note that for X-linkage both the main trait-marker (see 3.2.5.4) and the marker (see 3.2.5.4) must be specified to be X-linked and all hemizygous males must be coded as homozygous**. Be aware that for X-linkage the main result will be correct, but while we believe the results are appropriate for all the options available in LODLINK, this has not been carefully checked.

10.2 Theory

10.2.1 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.

1. For autosomal and pseudo-autosomal linkage, the probability of a phased joint genotype $\frac{T_b M_c}{T_d M_e}$ where *b, d = 1, ...k; c, e = 1, ..., m,* in the population is

$$\psi\left(\frac{T_b M_c}{T_d M_e}\right) = C\psi(T_b T_d)\psi(M_c M_e),$$

where

$$C = \begin{cases} 1 & if\ T_b = T_d\ and\ M_c = M_e, \\ \frac{1}{2} & if\ (T_b = T_d\ and\ M_c \neq M_e)\ or\ (T_b \neq T_d\ and\ M_c = M_e)\ , \\ \frac{1}{4} & otherwise \end{cases}$$

and

$\psi(T_bT_d)$ = probability of trait genotype in the population,

$\psi(M_cM_e)$ = probability of marker genotype in the population.

For X-linkage, females are as above, but males are hemizygous and for them the joint genotype probability in the population is

$$\psi(T_bM_c) = \psi(T_b)\psi(M_c),$$

where (in SEGREG) b = A or B.

2. The transmission probability for autosomal and pseudo-autosomal linkage is

$$\tau_s\left(\frac{T_bM_c}{T_dM_e} \to T_fM_g\right) = Pr\left(\ parent\ of\ sex\ s\ and\ genotype\ \tfrac{T_bM_c}{T_dM_e}\ transmits\ T_fM_g\ to\ child\ \right)$$

$$= \frac{(1-\theta_s)(\delta_{T_bT_f}\delta_{M_cM_g} + \delta_{T_dT_f}\delta_{M_eM_g})}{2} + \frac{\theta_s(\delta_{T_bT_f}\delta_{M_eM_g} + \delta_{T_dT_f}\delta_{M_cM_g})}{2},$$

where θ_s is the sex-dependent recombination fraction between the trait and marker loci ($\theta_s = \theta_{male}$ or θ_{female}) and

$$\delta_{xy} = \begin{cases} 1 & if\ x = y, \\ 0 & if\ x \neq y. \end{cases}$$

For X-linkage, the transmission probability depends on the sexes of both parent and offspring.

For female parents,

$$\tau_{female}\left(\frac{T_bM_c}{T_dM_e} \to T_fM_g\right) = Pr\left(\ female\ parent\ with\ genotype\ \tfrac{T_bM_c}{T_dM_e}\ transmits\ T_fM_g\ to\ child\ \right)$$

$$= \frac{(1-\theta_{female})(\delta_{T_bT_f}\delta_{M_cM_g} + \delta_{T_dT_f}\delta_{M_eM_g})}{2} + \frac{\theta_{female}(\delta_{T_bT_f}\delta_{M_eM_g} + \delta_{T_dT_f}\delta_{M_cM_g})}{2}$$

for all children, the male children being hemizygous.

For male parents,

$$\tau_{male}(T_bM_c \to T_fM_g) = Pr\left(\ male\ parent\ with\ genotype\ T_bM_c\ transmits\ T_fM_g\ to\ child\ \right)$$

$$= (1-\theta_{male})(\delta_{T_bT_f}\delta_{M_cM_g} + \delta_{T_dT_f}\delta_{M_eM_g}) + \theta_{male}(\delta_{T_bT_f}\delta_{M_eM_g} + \delta_{T_dT_f}\delta_{M_cM_g})$$

for female children and is irrelevant for the male children.

3. The transition probability is

$$Tr\left(\frac{T_bM_c}{T_dM_e}, \frac{T_rM_s}{T_uM_v}, \frac{T_fM_g}{T_hM_j}\right) = \Pr\left(\begin{array}{c} Mother\ with\ genotype\ \frac{T_bM_c}{T_dM_e}\ and\ Father\ with\ genotype\ \frac{T_rM_s}{T_uM_v} \\ have\ a\ child\ with\ genotype\ \frac{T_fM_g}{T_hM_j} \end{array}\right)$$

$$= \begin{cases} \tau_{female}\left(\frac{T_bM_c}{T_dM_e} \to T_fM_g\right)\tau_{male}\left(\frac{T_rM_s}{T_uM_v} \to T_hM_j\right) & if\ T_f = T_h\ and\ M_g = M_j, \\ \tau_{female}\left(\frac{T_bM_c}{T_dM_e} \to T_fM_g\right)\tau_{male}\left(\frac{T_rM_s}{T_uM_v} \to T_hM_j\right) + & \\ \tau_{female}\left(\frac{T_bM_c}{T_dM_e} \to T_hM_j\right)\tau_{male}\left(\frac{T_rM_s}{T_uM_v} \to T_fM_g\right) & otherwise. \end{cases}$$

4. For a phased joint genotype $\frac{T_bM_c}{T_dM_e}$ of pedigree member i, let the separate one-locus genotypes be denoted $u_i = T_bT_d$ for the trait and $v_i = M_cM_e$ for the marker. Let y_i be the trait and m_i be the marker phenotype (discrete). Let w_{male_i}, w_{female_i} and w_i indicate the phased two-locus (i.e., 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

$$L(\theta) = \sum_{w_1} \ldots \sum_{w_n} \prod_{i=1}^{n} H_i,$$

where

$$H_i = \begin{cases} p_i(w_{female_i}, w_{male_i}, w_i) & if\ i\ has\ any\ missing\ data, \\ p_i(w_{female_i}, w_{male_i}, w_i)g_{u_i}(y_i)g_{v_i}(m_i) & otherwise \end{cases}$$

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 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 $Z(\theta) = \text{Log}_{10}L(\theta)$ - $\text{Log}_{10}L(0.5)$.

If pedigree member i is a hemizygous male with joint genotype T_bM_c, then phase is irrelevant and it is only necessary to sum over the phases of females in $L(\theta)$. Let w_{female_i} and w_i indicate the phased two-locus genotypes of the mother of individual i, and individual i is male, respectively. Then for a male individual we have

$$H_i = \begin{cases} p_i(w_{female}, w_i) & if\ i\ has\ any\ missing\ data, \\ p_i(w_{female}, w_i) g_{u_i}(y_i) g_{v_i}(m_i) & otherwise \end{cases}$$

and for a female individual, we have the same H_i as given above.

10.2.2 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.

10.2.3 Hypothesis Tests

10.2.3.1 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 $0 \leq \hat{\theta}_{male}, \hat{\theta}_{female} < 0.5$.

10.2.3.2 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.

10.2.3.3 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

$$2[\sum_{i=1}^{n} \log_e L_i(\hat{\theta}_i) - \sum_{i=1}^{n} \log_e L_i(\hat{\theta})], \quad \begin{array}{l} with\ n-1\ degrees\ of\ freedom\ if\ \theta_{male} = \theta_{female} \\ 2(n-1)\ degrees\ of\ freedom\ if\ \theta_{male} \neq \theta_{female} \end{array},$$

The "asymptotic p-value" is the p-value based on the statistic following a chi-square distribution.

10.2.3.4 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 -α) $L_i(0.5)]$. 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

$$\chi_1^2 = 2[\sum_{i=1}^{n} \log_e L_i(\hat{\alpha}, \hat{\theta}) - \sum_{i=1}^{n} \log_e L_i(1, \hat{\theta})], \textit{ and the one sided } p-\textit{value is } 1-\Phi(\sqrt{\chi_1^2}),$$

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".

10.2.3.5 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".

10.2.3.6 Posterior Probability of Linkage

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.

10.2.4 Conditional Trait Genotype Probabilities

The table in the detailed file headed "Individual Genotype Probabilities" gives, for each pedigree member, the probabilities of having trait genotypes *bd* conditional on that member's output marker phenotype, assuming maximum likelihood estimates of the recombination fraction (or fractions, sex specific), and 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.

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.
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 trait genotype probability file are optional. One, but not both, may be used for LODLINK input.

10.3.1 Running `lodlink`

A typical run of the LODLINK program may use flags to identify the file types like the following:

```
>lodlink -p par -d ped -l loc -m mld (or typ)
```

or, rely on a set file order like the following:

```
>lodlink par ped loc tld (or typ)
```

where `par` is the name of the parameter file, `ped` is the name of the pedigree data file, `loc` is the name of the locus description file, `tld` is the name of the trait locus description file and `typ` is the name of the trait locus description file.

10.3.2 The `lodlink` Block

A `lodlink` block in the parameter file sets the options on how to perform an analysis using LODLINK.

The following table shows the syntax for a `lodlink` parameter which starts the `lodlink` block.

`parameter [, attribute]`	Explanation	
`lodlink`	Starts a LODLINK 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	lodlink_analysis*k* where *k* = 1, 2 ... *n* for a set of *n* analysis
	Required	No
	Applicable Notes	None

The following table lists the parameters and attributes that may occur in a `lodlink` block.

parameter [, attribute]	**Explanation**	
title	Specifies a title for the analysis	
	Value Range	Character string
	Default Value	LODLINK Analysis *k* DETAIL FILE (or SUMMARY FILE), where *k* = 1, 2, ..., *n* for a given set of *n* specified LODLINK analyses.
	Required	No
	Applicable Notes	None
model	Specifies the model for linkage calculations.	
	Value Range	N/A
	Default Value	N/A
	Required	Yes
	Applicable Notes	None
, trait	Model name in the ".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	See note 1.
	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	See note 1.
	Applicable Notes	1

, x_linked	Specifies the main trait/marker is x-linked. Value Range: {true, false} Default Value: false Required: Only for X-linked trait/markers. Applicable Notes: None
linkage_tests	Specifies option to perform linkage tests. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 2
, homog	Specifies option to assume linkage homogeneity. 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_tests	Starts a sub-block that specifies tests for linkage homogeneity. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 3
lods	Starts a sub-block that specifies lod score calculations. Value Range: N/A Default Value: N/A 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 not both.

2. Linkage tests are performed according to entries in 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.

10.3.2.1 The `homog_tests` Sub-Block

The following table lists the parameters and attributes that may occur in a `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

10.3.2.1.1 The `mortons_test` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 (which may contain multiple constituent pedigees, all assumed to be mutually independent) is its own group.
2. Each pedigree must be listed in one, and only one group in the `group` sub-block described below.

10.3.2.1.1.1 The `group` Sub-Block

The following table lists the parameters and attributes that may occur in a `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
   }
}
```

10.3.2.2 The `lods` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 the sex-specific recombination fractions at which lods will be calculated if `option` equals **specified** and `sex_specific` equals **true**. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 2

`average`	Starts a sub-block for specifying the sex-averaged recombination fractions at which lods will be calculated if `option` equals **specified** and `sex_specific` equals **false**.
	Value Range: N/A
	Default Value: N/A
	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**.

10.3.2.2.1 The `male_female` Sub-Block

The following table lists the parameters and attributes that may occur in a `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: N/A
	Required: No
	Applicable Notes: None
`, male`	Specifies the male recombination fraction.
	Value Range: [0, 1]
	Default Value: None
	Required: See note 1.
	Applicable Notes: 1
`, female`	Specifies the female recombination fraction.
	Value Range: [0, 1]
	Default Value: None
	Required: See note 1.
	Applicable Notes: 1

Notes:

1. Required if the `theta` parameter is specified.

10.3.2.2.2 The `average` Sub-Block

The following table lists the parameters and attributes that may occur in a `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: See note 1. 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
         }

         group = 2
         {
            pedigree_id = 5
            pedigree_id = 6
            pedigree_id = 7
         }
      }
   }

   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
      }
   }
}
```

10.4 Program Output

LODLINK produces four types of output files that contain results and diagnostic information:

File Name	File Type	Description
lodlink.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
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.sum	Summary output file	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.
analysis.det	Detailed output file	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

10.4.1 Genome Information Output File

This file lists for each marker the 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.

10.4.2 Summary Output File

Contains results pertaining to the whole data set. See 10.2 for details regarding interpretation of these results.

Example:

```
==============================
LODLINK Analysis 1 SUMMARY FILE
==============================


Options Selected
================
Main locus type                 trait
Main locus name                 T1

Lod scores                      yes

  Recombination Fractions Selected
  ================================
```

```
  Sex averaged      0.0000      0.0100      0.0500      0.1000      0.2000      0.3000

Linkage tests                           yes
  Sex-specific recombination fractions  no
  Assume homogeneity                    yes

Smith's test for homogeneity            no

Morton's test for homogeneity           no

Genotype probabilities                  no


Results
=======

                Lod Scores
                Non-Sex-Specific Recombination Fractions

Locus           0.0000         0.0100         0.0500         0.1000         0.2000         0.3000
--------------  -------------  -------------  -------------  -------------  -------------  -------------
M1                 -Infinity      -73.260693     -24.172918      -7.227167       3.151731       4.311157
M2                 -Infinity      -50.954605       6.758411      23.507636      27.514735      19.968754



                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                  0.273697           ---       4.422634      20.366984       3.20e-06       3.78e-05
M2                  0.166870           ---      28.236996     130.036172      < 1.0e-20      < 1.0e-20
```

10.4.3 Detailed Output File

Contains results on a per individual, per family or per group basis. See 10.2 for details regarding interpretation of these results.

Example:

```
==============================
LODLINK Analysis 1 DETAIL FILE
==============================

Options Selected
================
Main locus type                         trait
Main locus name                         T1

Lod scores                              yes

  Recombination Fractions Selected
  ================================
  Sex averaged      0.0000      0.0100      0.0500      0.1000      0.2000      0.3000
```

```
Linkage tests                           yes
  Sex-specific recombination fractions  no
  Assume homogeneity                    yes

Smith's test for homogeneity            no

Morton's test for homogeneity           no

Genotype probabilities                  no

Results
=======
                Lod Scores By Family
                Non-Sex-Specific Recombination Fractions


                Constituent Pedigree in Pedigree 1 Containing Member 1

Locus           0.0000        0.0100        0.0500        0.1000        0.2000        0.3000
--------------  ------------  ------------  ------------  ------------  ------------  ------------
M1                  0.000000      0.000000      0.000000      0.000000      0.000000      0.000000
M2                  0.000000      0.000000      0.000000      0.000000      0.000000      0.000000


                Constituent Pedigree in Pedigree 2 Containing Member 1

Locus           0.0000        0.0100        0.0500        0.1000        0.2000        0.3000
--------------  ------------  ------------  ------------  ------------  ------------  ------------
M1                  0.301030      0.292345      0.257679      0.214844      0.133539      0.064458
M2                  0.000000      0.000000      0.000000      0.000000      0.000000      0.000000

.
.
.
                Constituent Pedigree in Pedigree 199 Containing Member 1

Locus           0.0000        0.0100        0.0500        0.1000        0.2000        0.3000
--------------  ------------  ------------  ------------  ------------  ------------  ------------
M1                  0.346787      0.336379      0.295105      0.244678      0.150572      0.072111
M2                 -Infinity     -1.333237     -0.662295     -0.396473     -0.166627     -0.063424


                Constituent Pedigree in Pedigree 200 Containing Member 1

Locus           0.0000        0.0100        0.0500        0.1000        0.2000        0.3000
--------------  ------------  ------------  ------------  ------------  ------------  ------------
M1                  0.602060      0.588966      0.535294      0.465383      0.318063      0.170262
M2                  0.301030      0.288068      0.237795      0.179552      0.084934      0.026942



                Lod Score Linkage Test
                Variance-Covariance Matrices
                Parameter Order (Avg Recomb)

Locus           Estimates in [0, .5]          Estimates in [0, 1]
--------------  --------------------          -------------------
M1                  0.001231                            ---
M2                  0.000289                            ---
```

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 j index the relative pair: $j = 1, 2, \ldots, n$.

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

$\hat{f}_{0j}$ be the probability of sharing 0 alleles IBD at a given marker location, for the j -th pair,

$\hat{f}_{1j}$ be the probability of sharing 1 allele IBD at a given marker location, for the j -th pair, and

$\hat{f}_{2j}$ be the probability of sharing 2 alleles IBD at a given marker location, ffor the j -th 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 multi-marker or single-marker estimates. Marker is the 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 notation that has been used for the probability of sharing i alleles IBD between affected sib pairs at a particular locus, where λ_i is the locus-specific risk ratio or relative recurrence risk for a relative who shares i alleles identical by descent with an affected person:

# 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 independent ARPs is

$$z = \sum_{j=1}^{n} \log_{10}\left\{\frac{\hat{f}_{0j} + \hat{f}_{1j}e^{\beta_1} + \hat{f}_{2j}e^{\beta_2}}{f_{r0} + f_{r1}e^{\beta_1} + f_{r2}e^{\beta_2}}\right\}$$

$$= \sum_{j=1}^{n} \log_{10}\left\{\frac{\sum\limits_{i=0,1,2}\hat{f}_{ij}e^{\beta_i}}{\sum\limits_{i=0,1,2} f_{ri}e^{\beta_i}}\right\} = \sum_{j=1}^{n} \log_{10}\left\{\frac{\sum\limits_{i=0,1,2}\hat{f}_{ij}\lambda_i}{\sum\limits_{i=0,1,2} f_{ri}\lambda_i}\right\},$$

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 j -th full sib pair

$$\log_{10}\left\{4\frac{\hat{f}_{0j} + \hat{f}_{1j}e^{\beta_1} + \hat{f}_{2j}e^{\beta_2}}{1 + 2e^{\beta_1} + e^{\beta_2}}\right\}.$$

For half sibs, $f_{h0} = \frac{1}{2}$, $f_{h1} = \frac{1}{2}$, $f_{h2} = 0$, giving for the j -th half sib pair

$$\log_{10}\left\{2\frac{\hat{f}_{0i} + \hat{f}_{1j}e^{\beta_1}}{1 + e^{\beta_1}}\right\}.$$

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
First cousins	$\frac{3}{4}$	$\frac{1}{4}$	0

In the next sections, the subscript i indexing the pair and the summation over j 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

$$\log_{10}\left\{\frac{\hat{f}_0+\hat{f}_1 e^{\beta_1}+\hat{f}_2(3.634e^{\beta_1}-2.634)}{f_{r0}+f_{r1}e^{\beta_1}+f_{r2}(3.634e^{\beta_1}-2.634)}\right\}.$$

The constants in the above expression, $\alpha = 2.634$ and $\alpha+1 = 3.634$, fixes 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

$$\log_{10}\left\{\frac{\hat{f}_0+\hat{f}_1 e^{\beta_1}+\hat{f}_2[(\alpha+1)e^{\beta_1}-\alpha]}{f_{r0}+f_{r1}e^{\beta_1}+f_{r2}[(\alpha+1)e^{\beta_1}-\alpha]}\right\},$$

where $\alpha \geq 1$ is a mode of inheritance parameter: $\alpha = 1$ (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 $\lambda_2 = (\alpha + 1)\lambda_1 - \alpha$ in terms of relative recurrence risks and $\beta_1 \geq 0$ (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.

$$\log_{10}\left\{\frac{\hat{f}_0+\hat{f}_1 e^{\beta_1+z\delta}+\hat{f}_2[(\alpha+1)e^{\beta_1+z\delta}-\alpha]}{f_{r0}+f_{r1}e^{\beta_1+z\delta}+f_{r2}[(\alpha+1)e^{\beta_1+z\delta}-\alpha]}\right\},$$

where δ 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 parameter δ_c for each covariate.

- Constraints on the δ_c:

  Let the original (unadjusted) covariate value be denoted x. Two options are allowed:

  1. Genetic constraints on β_1 hold 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_c is unconstrained.
  2. Genetic constraints on β_1 hold 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 c, the following constraint is applied:

$$\min_{z>0}\sum_c z_c\delta_c \geq -\beta_1.$$

11.2.3 Adding Discordant Sib Pairs (DSPs) 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:

$$\lambda_1 = e^{\beta_1+z\delta},$$

where the covariate z 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 z to

- 0 if the pair is concordantly affected (concordantly unaffected pairs are not used in the analysis) and to
- 1 if the pair is discordant for affection status.

When either option is chosen, β_1 and δ are estimated subject to the constraints: $\beta_1 \geq 0$, $\delta \leq -\beta_1$.

No additional covariates may be included when discordant sib pairs are included in the analysis this way.

11.2.4 Contrasting Discordant Relative Pairs (DRPs) to Affected Relative Pairs (ARPs)

This model is the same as the ARP one- or two- parameter model except that the prior probabilities of sharing 0, 1, or 2 alleles IBD given a relative pair of type r (f_{r0},f_{r1},f_{r2},) are replaced by the probabilities, given the data, of sharing 0, 1, or 2 alleles IBD by the corresponding discordant relative pairs (Shih et. al. (2005)). With this option, any pair type for which there are no contrasting discordant relative pairs in the data is not included in the analysis. Additional covariates can be included in the case of a one-parameter model.

11.2.5 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

$$\log_{10}\left\{\frac{\sum_{i=0,1,2}\hat{f}_i\lambda_i}{\sum_{i=0,1,2}f_{ri}\lambda_i}\right\}.$$

For X-linked models, the LOD score is

$$\log_{10}\left\{\frac{\sum_{i=0,1,2}\hat{f}_{iuv}\lambda_{iuv}}{\sum_{i=0,1,2}f_{riuv}\lambda_{iuv}}\right\},$$

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. λ_{imf} =λ_{ifm}.

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 : $\lambda_{2ff} = (\alpha{+}1)\lambda_{1ff} - \alpha$
  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 separately estimable if the data contain 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 contain at least 15 male-male and male-female sib pairs (ASPs) under the default constraints on λ_1, or if the data contain 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].

11.2.5.1 Covariates

Inclusion of a single covariate (z) gives

$$\log_{10}\left\{\frac{\hat{f}_{0uv} + \hat{f}_{1uv}e^{\beta_{1uv}+z\delta_{uv}} + \hat{f}_{2uv}[(\alpha+1)e^{\beta_{1uv}+z\delta_{uv}} - \alpha]}{f_{r0uv} + f_{r1uv}e^{\beta_{1uv}+z\delta_{uv}} + f_{r2uv}[(\alpha+1)e^{\beta_{1uv}+z\delta_{uv}} - \alpha]}\right\},$$

where, δ_{uv} is an additional parameter 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 is one additional parameter for each covariate.

Under a generalization of this model, the LOD score is

$$\log_{10}\left\{\frac{\hat{f}_{0uv} + \hat{f}_{1uv}e^{\beta_{1uv}+\sum_c z_c\delta_{uv}} + \hat{f}_{2uv}[(\alpha+1)e^{\beta_{1uv}+\sum_c z_c\delta_{uv}} - \alpha]}{f_{r0uv} + f_{r1uv}e^{\beta_{1uv}+\sum_c \delta_{uv}z_l} + f_{r2uv}[(\alpha+1)e^{\beta_{1uv}+\sum_c z_c\delta_{uv}} - \alpha]}\right\},$$

where c indexes the covariate. Note that covariates can only be included in the one-parameter model, with the constraint $\lambda_{2ff} = (\alpha{+}1)\lambda_{1ff} - \alpha$.

Constraints on the δs are the same as for the autosomal models.

11.2.6 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

[2] As with the autosomal models, β in $\lambda_{iuv} = \exp(\beta_{iuv})$ is estimated instead of estimating λ itself.

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

$$\log_{10}\left\{\frac{\sum_{i=0,1m,1p,2}\hat{f}_i\lambda}{\sum_{i=0,1m,1p,2}f_{ri}\lambda_i}\right\},$$

where *m* denotes maternal and *p* denotes paternal, so that the sum is over *i = 0, 1m, 1p, 2* rather than over $i = 0, 1, 2$. As in previous models, λ_0= 1 and λ_i= exp(β_i), where β_i is the parameter estimated.

11.2.6.1 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., $\lambda_2 = (\alpha + 1)\lambda_1 - \alpha$.

11.2.6.2 Covariates

Covariates may be included only in the one-parameter model. Inclusion of a single covariate (z) gives $\lambda_{1m} = e^{\beta_{1m}+z\delta_m}$ and $\lambda_{1p} = e^{\beta_{1p}+z\delta_p}$, where δ_m and δ_p 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 $\lambda_{1m} = e^{\beta_{1m}+\sum_c z_c\delta_m}$ and $\lambda_{1p} = e^{\beta_{1p}+\sum_c z_c\delta_{lp}}$, where c 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. 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 the 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. (It should be recognized that, for other types of ARPs, parent-of-origin effects may be highly confounded with ascertainment.)

11.2.7 Asymptotic P-value and Empirical P-value

Two p-values, asymptotic and empirical, are reported under the conditions listed below:

- Autosomal locations
- One-parameter model
- Affected pairs only

- Pair count is between 20 and 350
- Number of covariate is less than 5
- Average location distance is between 1cM and 20 cM in case using multipoint ibd information.

Asymptotic p-values are computed using 50:50 mixture of χ^2 with c and $c+1$ degrees of freedom for c covariates. Empirical p-values are computed using the results described in Sinha et. al. (2006). Note that no p-values are computed when any of these conditions are not met!

11.3 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 (including individual-specific covariate values) and marker data.
IBD sharing file produced by GENIBD	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 and the pre-constructed pair-specific covariate and/or weight values.

Note:

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 Running `lodpal`

A typical run of the LODPAL program may use flags to identify the file types like the following:

```
>lodpal -p data.par -d data.ped -i ch1.ibd
```

or, rely on a set file order like the following:

```
>lodpal data.par data.ped ch1.ibd
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, and `ch1.ibd` is the name of the IBD sharing file.

11.3.2 The `lodpal` Block

A `lodpal` block in the parameter file sets the options on how to perform an analysis using LODPAL.

The following table shows the syntax for a `lodpal` parameter which starts the `lodpal` block.

`parameter [, attribute]`	Explanation
`lodpal`	Starts a LODPAL 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: lodpal
	Required: No
	Applicable Notes: None

The following table lists the parameters and attributes that may occur in 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 trait 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: N/A
	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: N/A
	Required: No
	Applicable Notes: 1
, noconunaff	Specifies option to analyze concordantly affected relative pairs and discordant sib pairs.
	Value Range: N/A
	Default Value: N/A
	Required: No
	Applicable Notes: 1

, contrast	Specifies option to analyze concordantly affected relative pairs (ARPs) in contrast with discordant relative pairs (DRPs).
	Value Range: N/A
	Default Value: N/A
	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 or location listed in the IBD sharing 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: N/A
	Required: No
	Applicable Notes: 4
, diff	Specifies option to include the covariate difference.
	Value Range: N/A
	Default Value: N/A
	Required: No
	Applicable Notes: 4

, both	Specifies option to include covariate terms: sum & difference. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4
, prod	Specifies option to include the covariate product. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4
, avg	Specifies option to include the covariate average. Value Range: N/A Default Value: N/A 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: N/A 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: N/A 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: N/A 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: N/A 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
`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
`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: 8
`autosomal` `autosomal_model`	Starts a sub-block for specification of an autosomal model on an existing autosomal marker. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 9
`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: N/A Required: No Applicable Notes: 10

Notes

1. The value of a `trait` parameter should be set to the name of a binary `trait` or `covariate` field read from the data file or created by means of a function block.

   (a) If no valid `trait` parameters are listed, then all `trait` fields read in from the 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 no attributes are listed, then by default the `conaff` attribute is assumed. This attribute causes the program to select only concordantly affected relative pairs (ARPs) and perform 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 a `trait` parameter has the `contrast` attribute specified then the program computes the probabilities of sharing 0, 1 or 2 alleles IBD for a given relative pair type from the discordant relative pairs in the data, and the prior probabilities are replaced by these values. Additional covariates can be included in the case of a one-parameter model.

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` or `covariate` field read from the data file or created by means of a function block. The `covariate` parameter may be included more than once. A `covariate` parameter may have two attributes, separated by a comma, to specify the function to compute a pair-specific value from two individual-specific values (`sum`, `diff`, `single`, `avg` or `prod`), and to adjust the covariate value to impose genetic constraints on them (`mean` or `minimum`).

(a) Individual covariate values are not mean-corrected, neither in LODPAL nor in SIBPAL. In the SIBPAL program, pair-specific covariate values are mean-corrected as stated in the manual. In LODPAL, however, pair-specific covariate values are either mean-corrected (by default) or minimum-adjusted (if `minimum` attribute is specified) as stated also in the manual.

(b) If `sum`, `diff`, `avg` or `prod` attributes are specified, then a single covariate sum, absolute difference, average or product term of two individual-specific values is included as a pair-specific covariate value.

(c) If the `both` attribute is specified, then both sum and difference terms are included.

(d) If the `single` attribute is specified, then the covariate value for the first member of the pair is included as a pair-specific value.

(e) If no attribute for the `covariate` parameter is specified, then the `sum` attribute is applied by default.

(f) If the `mean` attribute is specified, 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).

(g) 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 information on the number of pairs
   - partial first derivative of the maximum LOD score with respect to each of the parameters
   - the number of iterations it took for maximization

8. If the program finds the `pair_info_file` parameter with a valid file name, then the program uses the pre-constructed pair-specific covariate values from the file specified. The `pair_info_file` parameter may have its own sub-block to specify the name of the pair-specific covariate(s) to be used in the current analysis.

9. 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.

10. 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, the X-linked model will be used, in which all three λ_1 parameters are constrained to be equal, and λ_{2ff} is fixed.

11.3.2.1 The `pair_info_file` Sub-Block

The following table lists the parameters and attributes that may occur in 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 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	2
`, minimum`	Specifies option to set covariate values as offsets from the smallest observed value.	
	Value Range	N/A
	Default Value	N/A
	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 `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.

11.3.2.2 The `autosomal` Sub-Block

The following table lists the parameters and attributes that may occur in an `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 no genetic constraints on the parameters to be estimated. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 2
`, alpha`	Specifies an alpha value for Whittemore and Tu's one-parameter model. 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 a 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: 3

, all_pairs	Specifies the option to include non-sibs in the analysis Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4

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. If this parameter is not specified, the Holmans triangle constraints are applied.
3. A `fixed` value of **maternal** sets λ_{1m} equal to 1, and a `fixed` value of **paternal** sets λ_{1p} equal to 1.
4. By default, other types of affected relative pairs are excluded from the analysis to reduce the computational complexity of calculating IBD sharing probabilities for other types of affected relative pairs. If the `all_pairs` attribute is specified, then other types of affected relative pairs are included in the analysis, but the parent-of-origin effect test is applied only to the affected sib pairs. The other types of affected relative pairs are included with λ_1 be replaced by $\{(\lambda_{1m} + \lambda_{1p})/2\}$ to avoid fitting an extra parameter.

11.3.2.3 The `x_linkage` Sub-Block

The following table lists the parameters and attributes that may occur in a `x_linkage` sub-block.

parameter [, attribute]	**Explanation**
model	Specifies the type of model to use in the analysis. Value Range: {M-M, M-F, F-F, all} Default Value: all Required: No Applicable Notes: None
lambda1_equal	Specifies $\lambda_{1mm} = \lambda_{1mf} = \lambda_{1ff}$ regardless of the 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	$[1, \infty\}$
	Default Value	2.634
	Required	No
	Applicable Notes	None

The following are all valid LODPAL statements:

```
lodpal
{
   trait = T1
}


lodpal
{
   trait     = T1
   marker    = M1
   covariate = ageexam, minimum, diff
}


lodpal
{
   trait = T1
   autosomal_model
   {
      model = two_parameter
   }

   diagnostic = "20 44.0"  # The additional output file for location "20 44.0" is generated.
}


lodpal, out="t1condisc.out"
{
   trait = T1,condisc
   turn_off_default        # Turn off default maximization process
}


lodpal
{
   trait = T1, noconunaff       # Analysis is done with the one-parameter model.
}


lodpal
{
   trait = T1
   pair_info_file = "cov.in"
   {
      pair_weight    = probability
      pair_covariate = covariate1, mean
   }
}


lodpal
{
   trait = T1
   x_linkage
   {
      lambda2_fixed = true, alpha = 3.5  # The same as default model,
                                         # but the different alpha value is used.
```

```
   }
}

lodpal
{
   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.
   }
}
```

11.3.3 Pair Information File

The pair information file is a character delimited file that stores the pre-constructed pair-specific covariate 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 one line for each pair, with the required IDs and covariate fields. The pedigree ID (PEDID in the example below), first individual ID (ID1 in the example), and second individual ID (ID2 in the example) fields are required in that order, and covariate fields can be in any order. Each individual is expected to be found in the 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   weight1    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, weight1, 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 Output

LODPAL produces several output files that contain results and diagnostic information:

File Name	File Type	Description
lodpal.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
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.

11.4.1 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.

Example:

```
============================================================
  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
.
.
.
============================================================
```

11.4.2 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.

Example:

```
================================================================================
  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
...
109      3        4         0.0028969  0.1407563    -1.0821854        0.0003981387
-------- -------- -------- ---------- ----------  ---------------  ---------------
Total Pair Count = 202                             Total LOD Score =  3.3944467415
================================================================================
```

Chapter 12

MARKERINFO

MARKERINFO detects Mendelian inconsistencies in pedigree data. Each marker is individually checked for inconsistencies in every constituent pedigree. These inconsistencies are sorted by marker, by pedigree, and by whether one or more than one nuclear family is involved in the inconsistency.

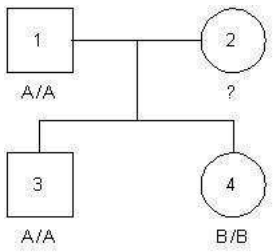
12.1 Limitations

MARKERINFO assumes codominant markers, 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).

12.2 Theory

The phenoset of an individual is the set of all genotypes consistent with that individual's marker phenotype. Individuals labeled as missing are considered to be consistent with all possible marker 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

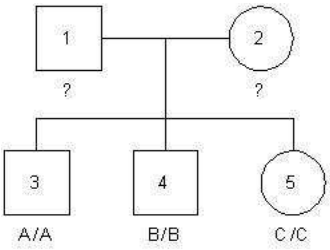


In this example pedigree, individuals 1 and 3 have a phenoset consisting of genotype A/A, while individual 4 has a phenoset consisting of genotype B/B. Individual 2 is unknown, so her phenoset includes all possible genotypes: {A/A, A/B, B/B, etc.}

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:

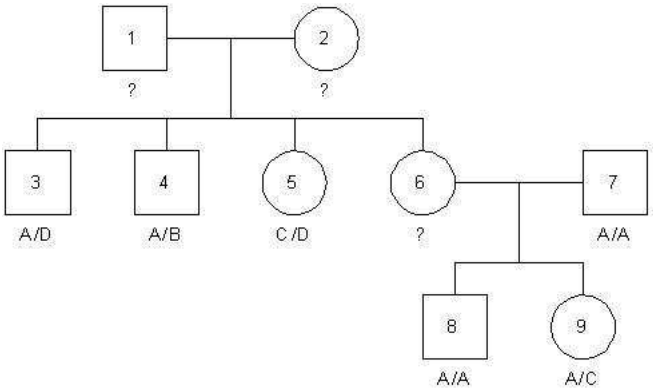


In this example, each of the children must receive a different set of alleles from each of their parents, but each parent has only two alleles. At least one child must be inconsistent with the parents, but it is impossible to determine which one.

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.

12.3 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.

12.3.1 Running `markerinfo`

A typical run of the MARKERINFO program may use flags to identify the file types like the following:

```
>markerinfo -p data.par -d data.ped
```

or, rely on a set file order like the following:

```
>markerinfo data.par data.ped
```

where `data.par` is the name of the parameter file and `data.ped` is the name of the pedigree data file.

12.3.2 The `markerinfo` Block

A `markerinfo` block in the parameter file sets the options on how to perform an analysis using MARKERINFO.

The following table shows the syntax for a `markerinfo` parameter which starts the `markerinfo` block.

parameter [, attribute]	Explanation
markerinfo	Starts a MARKERINFO 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: markerinfo Required: No Applicable Notes: None

The following table lists the parameters and attributes 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 string field in the 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
`pedigree_out`	Specifies option to generate a new pedigree file. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 3

Notes

1. The value of `sample_id` should be set equal to the name of a string 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.

3. If `pedigree_out` is set to **true**, a new pedigree file, along with a corresponding new parameter file, is produced in which, for all members of the pedigree with any inconsistency, those inconsistent markers are set to missing.

12.4 Program Output

SIBPAL produces several output files that contain results and diagnostic information:

File Name	File Type	Description
marker.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
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 *.

12.4.1 Analysis Output File

Here is a typical example of MARKERINFO output:

```
======================================================================
  MARKERINFO Analysis Output
======================================================================
----------------------------------------------------------------------
    Part 1.1: Number of Inconsistencies per pedigree
----------------------------------------------------------------------
======================================================================
                                        Number of Markers
Pedigree                          Incon.     Informative     Total
----------                       ----------  -------------  ----------
124                                     3            323         324
155                                     3            321         324
.
.
.
======================================================================
----------------------------------------------------------------------
    Part 1.2: Number of Inconsistencies per marker
----------------------------------------------------------------------
======================================================================
                                       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 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 across the genome.

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 quantitative traits). The time and space complexity of the algorithm is largely characterized by the number of genomic locations examined and 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. X-linked markers cannot be analyzed.

13.2 Theory

Given trait-marker (see 3.3, 3.2.5.4), T, and 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 location of interest along the chromosome by first generating exact multi-point likelihoods at each marker location 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 receiving 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 are translated 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

$$Z(p) = \log_{10}\left(\frac{P(M \,|\, T \, at \, p)}{P(M)P(T)}\right).$$

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

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-marker being analyzed.
Genome description file	Contains a description of the linked marker regions, including distances between markers.

13.3.1 Running `mlod`

A typical run of the MLOD program may use flags to identify the file types like the following:

```
>mlod -p data.par -d data.ped -g ch1.gen -m t1.trt -l ch1.loc
```

or, rely on a set file order like the following:

```
>mlod data.par data.ped t1.trt ch1.loc ch1.gen
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, `t1.trt` is the name of the trait locus description file or type probability file, `ch1.loc` is the name of the marker locus description file, and `ch1.gen` is the name of the genome description file.

13.3.2 The `mlod` Block

A `mlod` block in the parameter file sets the options on how to perform an analysis using MLOD.

The following table shows the syntax for a `mlod` parameter which starts the `mlod` block.

`parameter [, attribute]`	Explanation
`mlod`	Starts a MLOD 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: mlod_analysis
	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.

The following table lists the parameters and attributes that may occur in a `mlod` block.

parameter [, attribute]	**Explanation**
title	Specifies title of the run.
	Value Range: Character string
	Default Value: none
	Required: No
	Applicable Notes: None
trait_marker	Character string representing the name of a trait-marker to be analyzed. MLOD requires at least one trait-marker (see 3.3, 3.2.5.4) to be specified, but the user may list as many as desired.
	Value Range: Character string
	Default Value: None
	Required: Yes
	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: Yes
	Applicable Notes: 1
max_size	Maximum size (2n - f) of pedigree to analyze.
	Value Range: {0, 1, 2, ...}
	Default Value: 18
	Required: No
	Applicable Notes: None

`scan_type`	Specifies option to computes LOD scores at the observed markers, at the specified intervals or at the markers and intervals.
	Value Range: {marker, interval, both}
	Default Value: marker
	Required: No
	Applicable Notes: 2
`, distance`	Sets the interval used to compute LOD scores between observed markers in centiMorgans.
	Value Range: (0, ∞)
	Default Value: 2.0
	Required: No
	Applicable Notes: 2
`output_pedigrees`	Controls the amount of output generated on a per pedigree basis.
	Value Range: {none, marker, interval, both}
	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. 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 multiple `region` parameters are specified in the same analysis block, then the last region specified will be used.

2. The `scan_type` parameter defines the locations where LOD scores are to be computed. If the value of `scan_type` is set to **marker**, then LOD scores are computed only at observed marker loci. If set to **interval**, then LOD scores are computed only at intervals between markers defined by the `distance` attribute. If set to **both**, then LOD scores are computed both at the marker locations and at the intervals defined by the `distance` attribute.

3. If `output_pedigrees` is set equal to **marker**, the output for each pedigree is printed only for the markers. If `output_pedigrees` is set equal to **interval**, it is printed only for the intervals defined by the `distance` attribute for the `scan_type` parameter. If set equal to **both**, all points in the region are produced. Note that the `scan_type` parameter needs to be set properly for this option to work. For example, when `scan_type` is set to **marker**, then `output_pedigrees` can be set to only either **none** or **marker**. When `scan_type` is set to **both**, then `output_pedigrees` can be set to any value.

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**, then all individuals are included in the table with reason for removal or being kept.

13.4 Program Output

MLOD produces several output files that contain results and diagnostic information:

File Name	File Type	Description
mlod.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
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.
mlod_analysis.sum	LOD analysis summary output file(s)	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.
mlod_analysis.det	LOD analysis detailed output file	Contains a table for each pedigree analyzed, listing LOD scores and information content for each trait analyzed at each marker location.

13.4.1 Genome Information Output File

This file includes 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.

13.4.2 LOD Analysis 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.

Example:

```
# 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
.
.
.
```

13.4.3 LOD Analysis Detailed 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**.

Example:

```
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
.
.
.
```

Chapter 14

PEDINFO

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

14.1 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.

14.2 Theory

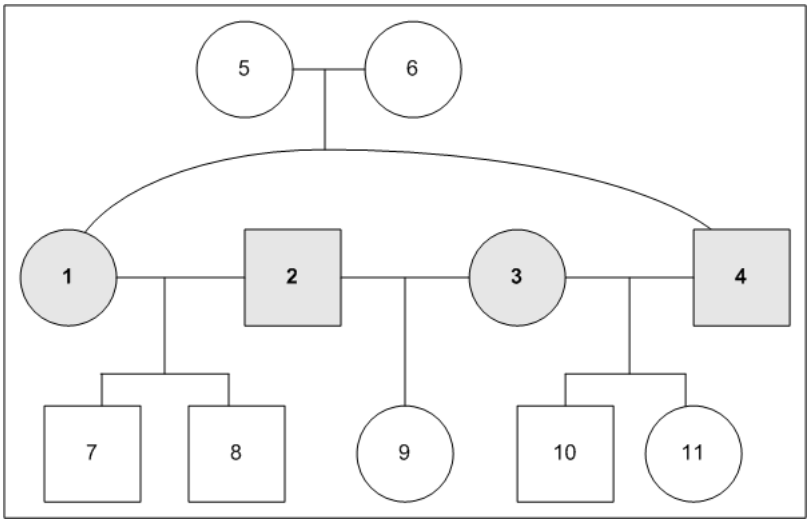
14.2.1 Terminology

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.

14.2.2 Problematic Family Structures

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

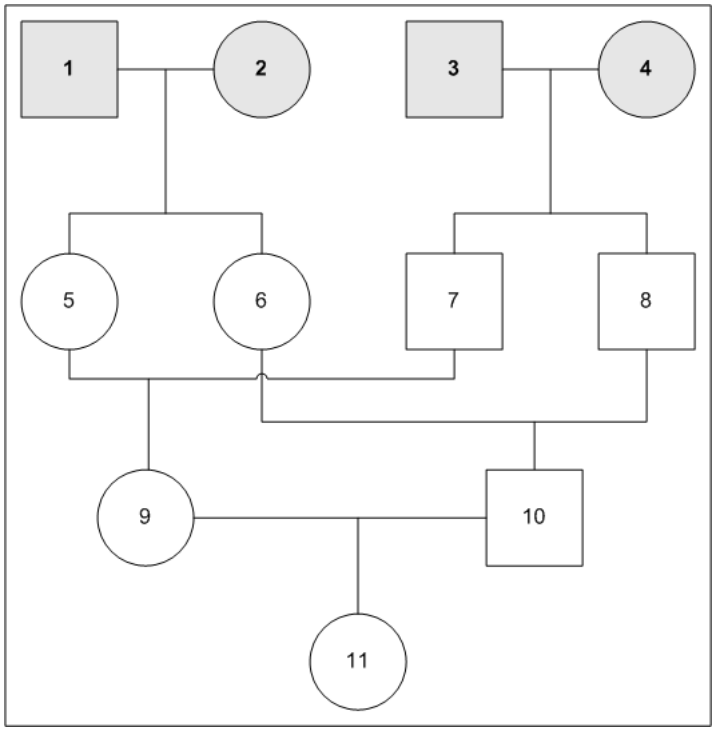


Individuals with multiple mates are enumerated in the output. For example, if the pedigree depicted above has Pedigree ID 1, the PEDINFO ouput will be:

```
|=============================================================================|
|                       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 cause computational difficulties for current programs using full pedigree structure information, and therefore individuals with multiple mates are listed so that users can find and 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. For example, if the above pedigree has Pedigree ID 1, the PEDINFO output will be:

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

When there are marriage rings or loops in the pedigree, some pairs are not distinct 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; e.g.

```
|=============================================================================|
|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 either by duplicating individuals or by removing certain connecting individuals.

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, sex, parents, trait and marker data.

14.3.1 Running `pedinfo`

A typical run of the PEDINFO program may use flags to identify the file types like the following:

```
>pedinfo -p parameters -d pedigree_data
```

or, rely on a set file order like the following:

```
>pedinfo parameters pedigree_data
```

where `parameters` is the name of the parameter file and `pedigree_data` is the name of the pedigree data file.

14.3.2 The `pedinfo` Block

A `pedinfo` block in the parameter file sets the options on how to perform an analysis using PEDINFO.

The following table shows the syntax for a `pedinfo` parameter which starts the `pedinfo` block.

parameter [, attribute]	Explanation
`pedinfo`	Starts a PEDINFO 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: pedinfo Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `pedinfo` block.

`parameter` [, `attribute`]	**Explanation**
`each_pedigree`	Specifies option to calculate statistics on a pedigree-by-pedigree basis Value Range: {true, false} Default Value: false Required: No Applicable Notes: 1
`suppress_general`	Specifies suppression of output for non-trait statistics. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 2
`print_table`	Specifies option to produce an additional output file of alternate tabular structure for non-trait statistics. Value Range: {true, false} Default Value: false Required: No Applicable Notes: none
`trait` `covariate`	Specifies a variable to be used in the analysis. Value Range: Character string representing the name of a trait or covariate listed in the data file. This parameter may be repeated. Default Value: None Required: No Applicable Notes: 3, 4

Notes

1. 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.

2. By default PEDINFO generates a report of general pedigree structure information *without* regard to any trait. If the `suppress_general` parameter is set to **true**, then this general output is suppressed, and reports are given only with respect to some specified trait.

3. The `trait` and `covariate` parameters are used to specify trait or covariate 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. This parameter can occur more than once. To be included in the statistics an individual must not have a missing value for variables included here as traits or covariates. See Note 4 for details about how missing data are treated.

   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 trait or covariate variables are specified, only non-trait or covariate information (i.e., based on pedigree structure alone) will be used to determine counts.

4. The following table details the way missing data for the variable (or variables) in question are treated for various statistics as a function of variable type. For the multiple variable case, an individual must have non-missing values for each of the specified variables to be considered informative (non-missing)

		Variable Type		
		Binary	Quantitative	Multiple
Statistic	Sibships by number of parents with data	a	a	a
	Sibships - count and size	b	b	b
	Pedigrees - count and size	c	c	c
	Pedigree counts by number of nuclear families	d	d	d
	Pairs	e	f	f
	Individuals	g	h	h

Description

(a) All sibships are counted regardless of whether the sibs have missing data.

(b) All sibships are counted, but sibship size refers to number of sibs with data.

(c) All pedigrees are counted, but pedigree size refers to number of pedigree members with data.

(d) All pedigrees are counted, but for a nuclear family to be counted it must have at least one parent and one child without missing data.

(e) Each pair is included in exactly one category.

(f) Only pairs where both individuals have non-missing data are included.

(g) Each individual is included in exactly one category.

(h) Only individuals without missing data are included.

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


pedinfo
{
}


# A pedinfo statement that specifies the name of an output file
# and requests a separate report for each pedigree
pedinfo,out=allpeds
{
   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               # if these two traits are binary
```

```
   trait=hematocrit
}

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

14.4 Program Output

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

File Name	File Type	Description
pedinfo.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
pedinfo.out	Analysis output file	Contains a table of summary statistics for all pedigrees combined, and optionally a table for each individual pedigree.

14.4.1 Analysis Output File

The PEDINFO analysis output file may contain the following types of tables (See notes following `pedinfo` parameter block for more information.):

- 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 variable or set of variables for the data as a whole.
- Tables of statistics pertaining to a specific variable or set of variables for a single pedigree.

Example:

```
=============================================================================
|                    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) |       |
===============================================================================
```

Chapter 15

RELPAL

This is a regression-based univariate or multivariate model-free two-level Haseman-Elston linkage program that models trait data from relative pairs as a function of marker allele sharing identity-by-descent (IBD) as proposed by Wang and Elston (2005, 2006). Available analyses can use both single- and multi- point IBD information, and models allow for both binary and quantitative traits caused by segregation at multiple genetic loci, including one epistatic interaction and covariate effects.

15.1 Limitations

This program is limited to pedigrees without loops and does not generate IBD sharing estimates itself. That must be done using GENIBD, which outputs an IBD sharing file as input for RELPAL. It is assumed that the only bilineal relatives in the data are full sibs.

15.2 Theory

15.2.1 Basic Notation

Let the number of pedigrees in the analysis be K.

Let i be the index of an individual: $i = 1, 2, ..., m_k$, where m_k is the total number of individuals in the k-th family.

Let the number of informative relative pairs in the k-th family be n_k, $k = 1, 2, ..., K$.

Let j be the index of a relative pair: $j = 1, 2, ..., \sum_k n_k = n$, where n is the total number of relative pairs.

Let the number of traits in the analysis be L, and let l be the index of a trait: $l = 1, 2, ..., L$.

Let $\otimes$ denote the Kronecker product.

Conditional on the marker information available, at a particular genomic location let $\hat{f}_{1j}$ be the probability of sharing 1 allele IBD, and $\hat{f}_{2j}$ be the probability of sharing 2 alleles IBD, for the j-th relative pair. Note that $\hat{f}_{2j} = 0$ in the case of non-full sib pairs.

Let $\pi = (1 + w_1)/4$ and $\hat{\pi}_j = \hat{f}_{2j} + w_1 \hat{f}_{1j}$ where $0 \leq w_1 \leq 0.5$ (Whittemore and Tu, 1998), for the j-th relative pair. The current default value of w_1 is 0.5.

15.2.2 Univariate Two-level Haseman-Elston Regression Model

With the assumption of randomly sampled pedigrees, a general two-level trait model is given by

$$y_{ik} = x_{ik}\beta + z_{ik}b + e_{ik}$$

where

- y_{ik} is the trait value of individual i in pedigree k,
- $x_{ik} = (1, x_{ik2}, ..., x_{ikr})$ is the design vector for fixed effects at the individual, or first level,
- $\beta = (\beta_1, \beta_2, ..., \beta_r)^{\mathrm{T}}$ is the coefficient vector of fixed effects at the first level,
- $z_{ik} = (z_{iku_1}, ..., z_{iku_s}, z_{ikp})$ is the design vector for random effects at the pedigree, or second level,
- b is the coefficient vector at the second level containing coefficients $u_1 ... u_s$ for up to s covariates at the pedigree level and a polygeneic effect p_{ik},
- e_{ik} are random individual effects assumed to be independently and identically distributed as N(0, σ_e^2).

The effect of a QTL is incorporated into either the first level or the second level, depending on the particular analysis. The polygenic effects are assumed to be independent across all founders in all pedigrees, and any common environmental effects are assumed to be confounded with the polygenic effects.

Under this trait model, we have r +1 coefficients of fixed effects β at the first level, s variances $\sigma_u^2 = (\sigma_{u_1}^2, ..., \sigma_{u_s}^2)$ and the variance σ_p^2 at the second level, and σ_e^2. The QTL parameter σ_g^2 may be included in β or σ_u^2. However, in a linkage analysis, the number of disease alleles is not observed directly (or the marker cannot be assumed to be in linkage disequilibrium with the disease locus), so we model the QTL effect at the second level, *i.e.* , $\sigma_{u_1}^2 = \sigma_g^2$.

15.2.2.1 Estimation

To estimate the above parameters, $(\sigma_u^2, \sigma_p^2, \sigma_e^2)$, an iterative generalized least squares (IGLS) algorithm (Goldstein, 2003) is used as follows:

1. First, an ordinary least squares estimate $\hat{\beta}$ of the fixed effect parameter β is obtained with the assumption that (σ_u^2, σ_p^2) are 0.
2. Let
$$\tilde{e}_k = y_k - X_k\hat{\beta}$$
where the length of $\tilde{e}_k$, y_k, and X_k is the number of individuals in pedigree k. Then, the expectation of the cross-product matrix $\tilde{e}_k\tilde{e}_k^T$ is simply the variance-covariance matrix for pedigree k, *i.e.* , $E(\tilde{e}_k\tilde{e}_k^T) = V_k = Z_k\Omega Z_k^T + \sigma_e^2 I$, where the way the entries in Z_k for pedigree

k are defined depends on the linkage analysis, Ω is a diagonal matrix with diagonal elements (σ_u^2, σ_p^2), and I is an identity matrix. Residuals, $\tilde{e}$, are transformed to follow a marginal normal distribution with the same varianes, and this normalization can be optionally turned off. We now rearrange the cross-product matrix $\tilde{e}_k\tilde{e}_k^T$ and the variance-covariance matrix V_k as vectors by stacking the columns one on top of the other so that we have $E[vec(\tilde{e}_k\tilde{e}_k^T)] = vec(V_k)$. This can be written as

$$\tilde{\eta}_k = \Delta_k\varpi + \varepsilon_k$$

where $\tilde{\eta}_k = vec(\tilde{e}_k\tilde{e}_k^T)$, $\varpi = (\sigma_e^2, \sigma_u^2, \sigma_p^2)$, $\Delta_k = \begin{bmatrix} 1 & z_{1ku}z_{1ku} & z_{1kp}z_{1kp} \\ 0 & z_{1ku}z_{2ku} & z_{1kp}z_{2kp} \\ \ldots & \ldots & \ldots \\ 1 & z_{mku}z_{mku} & z_{mkp}z_{mkp} \end{bmatrix}$ is the design matrix of cross-products for pedigree k, and ε_k is a residual vector. Then, the left-hand side of the equation can be treated as the response vector in a linear model, the right-hand side contains observed explanatory variables in the design matirx Δ_k, and the entries of ϖ are the regression coefficients of this linear model. A generalized least squares analysis is used to estimate the entries of the expectation of ϖ, namely

$$\tilde{\varpi} = (\Delta^T V^{*-1}\Delta)^{-1}\Delta^T V^{*-1}\tilde{\eta}$$

where is V^* a block-diagonal matrix made up of the matrices $V_k^* = 2(V_k \otimes V_k)$ and $\tilde{\varpi} = (\hat{\sigma}_e^2, \hat{\sigma}_u^2, \hat{\sigma}_p^2)^T$.

3. Based on $\tilde{\varpi}$, the variance-covariance matrix $\hat{V}$, an estimate of V, the block diagonal matrix made up of the matrices V_{k}, is obtained. Then, the weighted least squares estimate of the vector of fixed coefficients is given by

$$\tilde{\beta} = (X^T\hat{V}^{-1}X)^{-1}X^T\hat{V}^{-1}y.$$

Step 1 to 3 are iterated until the procedure converges.

4. Finally, with the assumption of multivariate normality, the model-based estimate of the variance-covariance matrix of $\tilde{\varpi}$ is given by $Var(\tilde{\varpi}) = (\Delta^T\hat{V}^{*-1}\Delta)^{-1}$.

15.2.3 Multivariate Two-level Haseman-Elston Regression Model

The multivariate model for general unlooped pedigrees is constructed under the same framework of two-level regression as the univariate case above. Denote the vector of trait values observed on family k, of length lm_k, y_k with transpose y_k^T. After the trait values have been adjusted for individual covariates, $E(y_k y_k^T)$ is the variance-covariance matrix of the traits for family k.

The variance-covariance matrix is given by

$$P \otimes \Phi_k + G \otimes \Pi_k + E \otimes I_k$$

where

- Φ_k is an $m_k \times m_k$ matrix of coefficients of relationship between relative pairs,
- Π_k is an $m_k \times m_k$ matrix of the observed proportions of alleles shared IBD at a particular locus,

- I_k is an identity matrix,
- P , G and E are $L \times L$ matrices of polygenic, quantitative trait locus (QTL) and random individual effects, respectively.

Let $E(y_k y_k^T) = E(A)$, let the variance-covariance matrix under the null hypothesis of no QTL effect be B^0, and let the variance-covariance matrix under the alternative hypothesis be B^1. Then, we have $vec[E(A)] = vec[B^1]$ at the second level regression, *i.e.* rearranging the cross-product matrix $E(A)$ and the variance-covariance matrix B^1 as vectors by stacking the columns on top of each other. The parametrers of this regression model are obtained by applying an IGLS algorithm as in the univariate case.

Under this regression model, a score is defined by

$$U = \sum_{k=1}^{K} U_k = \sum_{k=1}^{K} D_k^T W_k^{-1} S_k$$

where

- $S_k = vec(A_k)$, which measures the trait similarity among relative pairs,
- $D_k = \frac{\partial vec(B_k^1 - B_k^0)}{\partial \sigma_g}$, which measures the genetic (i.e. the proportion of alleles IBD) similarity among relative pairs and in which σ_g is a vector of QTL parameters of interest,
- W_k is an appropriate weight matrix to take account of the correlations between elements in S_k, which is defined as $W_k = 2(\hat{B}_k^0 \otimes \hat{B}_k^0)$ in our analysis.

U is further modified as a weighted least squares estimate of A that uses a weight matrix estimated under the null hypothesis, which is $U^* = (\sum_{k=1}^{K} D_k^T W_k^{-1} D_k)^{-1} U$, and we finally define a score statistic by

$$T_{unadjusted} = U^{*T} \Sigma_{U^*}^{-1} U^*$$

for some Σ_{U^*} defined below. In the case of a univariate trait, this statistic asymptotically follows a 50:50 mixture of a $\chi^2_{df=1}$ and a point mass at 1. However, for multivariate models, the asymptotic distribution of this statistic may be very complex and the p-value difficult to obtain. For this reason, we evaluate the p-value through a Monte Carlo procedure described in the next section.

15.2.3.1 One-sided Adjusted Score Statistic

Score tests for testing variance-covariance components under constrained parameterization requires replacing the classical score test statistic by an appropriate one-sided version as in Verbeke and Molenberghs (2003). Here we use the Cholesky decomposition to find feasible estimates, and the score statistic T is adjusted as

$$T_{adjusted} = U^{*T} \Sigma_{U^*}^{-1} U^* - \inf_{b \in \Theta} ((U^* - b)^T \Sigma_{U^*}^{-1} (U^* - b)),$$

where Θ is positive definite.

15.2.3.2 Variances

Four different estimators of the variance of U^* are given.

For the naive approach, assuming that the data generating process is multivariate normal, the estimator is given by

$$\Sigma_{naive} = (\sum_{k=1}^{K} D_k^T W_k^{-1} D_k)^{-1}.$$

The standard sandwich estimator under the null hypothesis is given by

$$\Sigma_{null} = \Sigma_{naive}(\sum_{k=1}^{K} U_k U_k^T)\Sigma_{naive}.$$

An alternative estimator is given by

$$\Sigma_{alt} = \Sigma_{naive}(\sum_{k=1}^{K} (U_k - \hat{E}(U_k))(U_k - \hat{E}(U_k))^T)\Sigma_{naive}$$

where $\hat{E}(U_k) = D_k^T W_k^{-1} D_k U^*$.

A new estimator using the variance of the IBD values conditional on Y is given by

$$\Sigma_{ibd} = \Sigma_{naive}(\sum_{k=1}^{K} B_k Var[Vec(\tilde{\Pi}_k)]B_k^T)\Sigma_{naive}$$

where there exists a matrix B_k such that $U_k = B_k Vec(\tilde{\Pi}_k)$.

15.2.4 Significance Tests

15.2.4.1 First level Wald Test

For both the univariate and multivariate cases, the significance of the effects at the first level are tested as follow. The weighted least squares estimate of the vector of fixed coefficients at the first level is given by

$$\underline{\tilde{\beta}} = (X^T \hat{V}^{-1} X)^{-1} X^T \hat{V}^{-1} y,$$

and

$$\underline{\tilde{\beta}} \sim MVN[\underline{\beta}, (X^T \hat{V}^{-1} X)^{-1}].$$

We wish to test $H_0 : A\underline{\beta} = 0$ vs. $H_1 : A\underline{\beta} \neq 0$. Then $\chi^2_{df=F} = (A\underline{\beta})^T [A(X^T \hat{V}^{-1} X)^{-1} A^T](A\underline{\beta})$, where A is a matrix whose elements are 0 and 1 to select only the effects F being tested.

15.2.4.2 Second level Score Test

For both the univariate and multivariate cases, the asymptotic p-value for the adjusted score statistic T is calculated as follows.

Let T_{org} be the adjusted score statistic from the data.

Let F_c be the cumulative distribution function of a $\chi^2_{df=c}$ where c is the number of variance-covariance components in the test.

1. Simulate a vector of size c from a multivariate normal distribution, then divide the vector by its norm to get U.
2. Calculate T, the adjusted score statistic, using U as a score.
3. Repeat steps 1 and 2 N times, resulting in T_1, T_2, ..., T_N. Then the estimated p-value is $\hat{p}_N = \frac{\sum_i (1 - F_p(\frac{T_{org}}{T_i}))}{N}$.
4. If not converged, then return to step 1. Otherwise, stop.

N is determined such that the estimated asymptotic p-value $\hat{p}$ is within a proportion ω (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 p, where the sample variance of $\hat{p}$ is $s^2_{\hat{p}} = \frac{s^2}{N}$, where s^2 is the variance of the N values of $1 - F_p(\frac{T_{org}}{T_i})$. So we choose N such that $Pr(|\hat{p} - p| \leq \omega\hat{p}) = \gamma$. Using a normal approximation for the distribution of $\hat{p}$, we obtain

$$N = (\frac{s^2}{\hat{p}\omega^2}[\Phi^{-1}(\frac{\gamma+1}{2})]^2)$$

where Φ is the standard normal cumulative distribution function.

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.
IBD sharing file	Stores identity-by-descent (IBD) distributions between pairs of related individuals at one or more marker loci.

15.3.1 Running `relpal`

A typical run of the RELPAL program may use flags to identify the file types like the following:

```
>relpal -p data.par -d data.ped -i ch1.ibd
```

or, rely on a set file order like the following:

```
>relpal data.par data.ped ch1.ibd
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, and `ch1.ibd` is the name of the IBD sharing file.

15.3.2 The `relpal` Block

A `relpal` block in the parameter file sets the options on how to perform an analysis using SIBPAL.

The following table shows the syntax for a `relpal` parameter which starts the `relpal` block.

`parameter` `[, attribute]`	Explanation
`relpal`	Starts a RELPAL 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: relpal Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `relpal` block.

parameter [, attribute]	Explanation
`trait`	Specifies a trait to be used as the dependant variable in the current analysis. Value Range: Character string representing the name of a trait listed in the pedigree data file or created by means of a function block. Default Value: None Required: Yes Applicable Notes: 1
`model`	Specifies the second level analysis model type in regard to the number of markers included in the current analysis. Value Range: zero_marker, single_marker, multiple_marker Default Value: single_marker Required: No Applicable Notes: 2
`first_level`	Specifies a sub-block of first level parameters included in the current analysis. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`second_level`	Specifies a sub-block of second level parameters included in the current analysis. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`data_options`	Specifies a sub-block of data filtering option parameters for the current analysis. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`output_options`	Specifies a sub-block of various output option parameters for the current analysis. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None

pvalue_options	Specifies a sub-block of various parameters for p-value calculation for the score test in the current analysis.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None

Notes

1. The value of a `trait` parameter should be set to the name of a `trait` or `covariate` field read from the data file or created by means of a function block. This parameter may be included more than once. To have a valid analysis block, at least one `trait` parameter must be specified. Depending on the number of `trait` parameters in the analysis block, either a univariate or multivariate analysis will be performed.

2. The value for a `model` parameter should be set to one of the following values:

   - **zero_marker :** Haseman-Elston regression model with covariate(s) only. Note that no ibd data are used with this option. In this model, at least one `covariate` parameter has to be specified as a test covariate in the `second_level` sub-block to perform a score test at the second level. If more than one `covariate` parameter is specified in the `second_level` sub-block, one `covariate` parameter has to be specified as a test covariate. Then the test of this covariate effect in the presence of other(s) will be performed. When no test covariates are listed in the `second_level` sub-block, only the tests in the first level are performed, without any score test in the second level.
   - **single_marker :** Haseman-Elston regression model with one marker and covariate(s). In this model, all markers/locations in the IBD sharing file are test markers by default. So, no valid `marker` parameters are required for this model since the analyses are done using all markers/locations in the IBD sharing file one by one when no valid `marker` parameters are listed in the `second_level` sub-block. If a subset of the markers from the IBD file is to be analyzed, one or more `marker` parameters should be specified. Any number of optional non-test `covariate` parameters may be included in the `second_level` sub-block for this model.
   - **multiple_marker :** Haseman-Elston regression analysis with multiple markers and covariate(s). At least one `marker` parameter has to be specified in the `second_level` sub-block for this model. If one `marker` parameter is specified as a test marker, then the tests for linkage to this marker are done in the presence of other marker using all markers in the IBD sharing file one by one. If one `marker` parameter is specified as a non-test marker, then the tests for linkage to all markers in the IBD sharing file are done one by one in the presence of this marker. If more than one `marker` parameters is specified, one `marker` parameter has to be specified as a test marker. Then the test for linkage to this test marker in the presence of other(s) will be performed. Any number of optional non-test `covariate` parameters may be included in the `second_level` sub-block for this model.

15.3.2.1 The `first_level` Sub-Block

The following table lists the parameters and attributes that may occur in a `first_level` sub-block.

parameter [, attribute]	Explanation
covariate	Specifies a covariate to be included at the first (individual) level. Value Range: Character string representing the name of a covariate listed in the pedigree data file. Default Value: None Required: No Applicable Notes: 1
, adj_trait	Specifies a trait to be adjusted. Only valid in a multivariate analysis. Value Range: Character string representing the name of a trait listed in the trait parameter. Default Value: None Required: No Applicable Notes: 1
interaction	Specifies an interaction to be included at the first (individual) level. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 2
normalize_residual	Specifies an option to insure that variance components are calculated on the basis of normally distributed residuals. Value Range: {true, false} Default Value: true Required: No Applicable Notes: 3
batch	Specifies automated covariate term inclusion. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4

Notes

1. The value of a `covariate` parameter should be set to the name of a `trait` or `covariate` field read from the data file or created by means of a function block. The `covariate` parameter may be included more than once. If no valid `covariate` parameters are listed, then by default no covariates are included in the first level. If the `adj_trait` attribute is specified with a valid name of a trait, then only the specified trait among all traits in the model is adjusted for this covariate. Otherwise all traits in the current analysis are adjusted for the covariate in the first level regression. Note that this `adj_trait` attribute is only applicable in a multivariate analysis.

2. The `interaction` parameter should contain a sub-block of two first level `covariate` parameters to specify a multiplicative interaction term in the model.

```
interaction
{
   covariate = AGE
   covariate = BMI
}
```

3. If the `normalize_residual` parameter is set to false, then the residuals from the first_level will not be normalized before calculation of variance components at the second_level. If no valid `normalize_residual` parameter is listed, or if the parameter is set to true, then by default the residuals from the first_level will be transformed to follow a marginal normal distribution with the same variance before being used at the second_level.

4. If the `batch` parameter is specified in the sub-block, then all first level covariate fields read from the data file or created by means of a function block will be automatically included one by one. Note that this option is valid only with the `zero_marker` model and when no second level test covariates are specified.

15.3.2.2 The `second_level` Sub-Block

The following table lists the parameters and attributes that may occur in a `second_level` sub-block.

parameter [, attribute]	Explanation	
`marker`	Specifies a marker to be included at the second (pedigree) level.	
	Value Range	Character string representing the name of a marker/location listed in the IBD file.
	Default Value	None
	Required	No
	Applicable Notes	1
`, test`	Specifies a test marker. Only valid with `multiple_marker` model.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	1
`covariate`	Specifies a covariate to be included at the second (pedigree) level.	
	Value Range	Character string representing the name of a covariate listed in the pedigree data file.
	Default Value	None
	Required	No
	Applicable Notes	2

, test	Specifies a test covariate. Only valid with zero_marker model. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 2
interaction	Specifies an interaction sub-block to be included at the second (pedigree) level. Only valid in a univariate model. Value Range: None Default Value: None Required: No Applicable Notes: 3
, batch	Specifies the automated covariate term inclusion. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4
naive_variance	Specifies the score test done using the naive variance. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
sandwich_variance	Specifies the score test done using the sandwich variance. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
alternative_variance	Specifies the score test done using the alternative variance. Value Range: {true, false} Default Value: true Required: No Applicable Notes: None
ibd_variance	Specifies the score test done using the ibd variance. Value Range: {true, false} Default Value: true 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 the test attribute is specified in the case of multiple_marker model, then this marker is used to test for linkage in the presence of other marker(s). If no valid marker parameters are listed then all markers in the IBD sharing file are used one by one.

2. The value of a `covariate` parameter should be set to the name of a `trait` or `covariate` field read from the data file or created by means of a function block. The `covariate` parameter may be included more than once. If the `test` attribute is specified in case of `zero_marker` model, then this covariate is used to test the effect in the presence of other covariate(s) . If no valid `covariate` parameters are listed, then by default no covariates are included.

3. The `interaction` parameter should contain a sub-block of two main effect parameters; two `marker` parameters, or one second level `covariate` parameter and one `marker` parameter that specify a multiplicative interaction term in the model. Note that this option is only valid in a univariate analysis with both corresponding main effects included in the analysis, so marker by marker interaction can be included only in `multiple_marker` model. The following interaction sub-block specifies a gene-environment interaction term between D1S344 and BMI:

```
interaction
{
   marker    = D1S344
   covariate = BMI
}
```

4. If the `batch` parameter is specified with only one `covariate` or `marker` parameter in the sub-block, then the interaction term between the given `covariate` or `marker` and a marker in the IBD sharing file will be automatically included one by one. Note that this option is valid only in a univariate analysis. The following interaction sub-block specifies a gene-environment interaction term between the dominance component of D1S344 and the squared BMI difference:

```
# Include marker by covariate interaction for all markers.
interaction
{
   covariate = BMI
}
# Include marker by marker interaction for all markers.
interaction
{
   marker = D1S344
}
```

15.3.2.3 The `data_options` Sub-Block

The following table lists the parameters and attributes that may occur in a `data_options` sub-block.

parameter [, attribute]	Explanation
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 or covariate listed in the pedigree file or created by means of a function block.
	Default Value: None
	Required: No
	Applicable Notes: 1
use_pairs	Specifies the types of relative pairs to use for the current analysis.
	Value Range: {all, fsib, sib}
	Default Value: all
	Required: No
	Applicable Notes: 2

Notes

1. 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.

2. The value of a `use_pairs` parameter should be set to one of fsib, `sib` or `all` depending on whether the analysis include full sib pairs only, full and half sib pairs, or all relative pairs from the IBD file.

15.3.2.4 The `output_options` Sub-Block

The following table lists the parameters and attributes that may occur in a `output_options` sub-block.

parameter [, attribute]	Explanation
detailed_out	Prints a separate detailed output file for the analysis.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: None
export_out	Prints a separate output file in comma-delimitted format.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: None

15.3.2.5 The `pvalue_options` Sub-Block

The following table lists the parameters and attributes that may occur in a `pvalue_options` sub-block.

parameter [, attribute]	Explanation
`seed`	Specifies the seed. If no seed value is specified, or the value "0" is used, then a random number is used as the seed by default. Value Range: {1, 2, 3, ...} Default Value: 0 Required: No Applicable Notes: 1
`replicates`	Specifies an exact number of replicates that should always be simulated to compute the empirical p-value.Use of this option effectively overrides all of the following parameters. Value Range: {1, 2, 3, ...} Default Value: None Required: No Applicable Notes: None
`min_replicates`	Specifies an minimum number of replicates that should be simulated. Value Range: {1, 2, 3, ...} Default Value: 50 Required: No Applicable Notes: None
`max_replicates`	Specifies an maximum number of replicates that should be simulated. Value Range: {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 the width interval of its true value.	
	Value Range	[0, 1]
	Default Value	0.95
	Required	No
	Applicable Notes	None

Notes

1. If not specified, different seeds, and hence different results, will be obtained each time a given analysis is performed.

15.4 Program Output

RELPAL produces several output files that contain results and diagnostic information:

File Name	File Type	Description
relpal.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
relpal.out	Relpal analysis summary output file	Contains the summary results of each test.
relpal.det	Relpal analysis detailed output file	Contains the detailed results of each test.

15.4.1 Summary Output File

One main analysis output file, named "`relpal.out`", is generated per run of RELPAL. It contains the results of all tests.

Example:

```
================================================================================
  Two-Level Haseman-Elston Regression Analysis for General Pedigree
  - Second (Pedigree) Level Score Test Summary Output
================================================================================

 Traits : (1) - EF, Quantitative
          (2) - Q2, Quantitative

 Legend :
    *   - significance  .05 level
    **  - significance  .01 level
    *** - significance .001 level

    nai - naive variance
    sdw - robust sandwich variance
    alt - alternative variance
    ibd - allele sharing variance

    #   - The number of pedigrees is too small for this result to be
          reliable when analyzing this number of traits.

 Empirical p-value options used :
    seed = 0
    min replicates = 20
    max replicates = 10000
```

```
================================================================================
Test                                 Unadjusted   Adjusted Empirical     Number of
No   Variable             Count Var    T-value    T-value   P-value     Replicates
------------------------ ----- --- ---------- ---------- ---------     ----------
1 #  D5G1                   771 nai     0.1740     0.0085 0.8281762             22
                                 sdw     0.4172     0.0184 0.9467879             26
                                 alt     0.1740     0.0085 0.8281762             43
                                 ibd     0.2268     0.0301 0.9851194             24
2    D5G2                   771 nai     1.5330     0.0120 0.9484626             22
                                 sdw     4.4773     0.0850 0.8739591             22
                                 alt    36.8187     5.2020 0.0706064             84
                                 ibd     0.8478     0.0572 0.9413330             22
.
.
.
```

15.4.2 Detailed Output File

One detailed output file, named "`relpal.det`", is generated per run of RELPAL optionally. It contains more detailed information of the results of all tests.

Example:

```
================================================================================
  Two-Level Haseman-Elston Regression Analysis for General Pedigree
  - Second (Pedigree) Level Score Test Summary Output
================================================================================

 Traits : (1) - EF, Quantitative
          (2) - Q2, Quantitative

 Legend :
    *   - significance  .05 level
    **  - significance  .01 level
    *** - significance .001 level

    nai - naive variance
    sdw - robust sandwich variance
    alt - alternative variance
    ibd - allele sharing variance

    #   - The number of pedigrees is too small for this result to be
          reliable when analyzing this number of traits.

 Empirical p-value options used :
    seed = 0
    min replicates = 20
```

```
    max replicates = 10000

================================================================================
Test 1 #
================================================================================
-----
Model
-----
H0: EF, Q2 ~ [Intercept + Q3] + POLYGENIC_EFF + RANDOM_EFF
H1: EF, Q2 ~ [Intercept + Q3] + D5G1 + POLYGENIC_EFF + RANDOM_EFF

------
Sample
------

Number of individuals used at first level     = 686
Number of relative pairs used at second level = 771

-----------------
Estimates from H0
-----------------

Test                          Nominal
Variable                      Estimate Variance-Covariance   Chi-sq.   P-value
------------------------- ------------ -------------------- --------- ---------
Q3(1)                           0.0772    0.0001    0.0000  242.5299 0.0000000 ***
Q3(2)                           0.2275    0.0000    0.0002

------------------
Score Test Results
------------------
Test                          Unadjusted   Adjusted Empirical     Number of
Variable                 Var    T-value    T-value   P-value     Replicates
------------------------- --- ---------- ---------- ---------     ----------
D5G1                     nai     0.1740     0.0085 0.8281762             22
                         sdw     0.4172     0.0184 0.9467879             26
                         alt     0.1740     0.0085 0.8281762             43
                         ibd     0.2268     0.0301 0.9851194             24
================================================================================
Test 2
================================================================================
.
.
.
```

Chapter 16

RELTEST

RELTEST helps classify pairs in a sib pair linkage study according to their true relationship using autosomal 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 and parent-offspring classification statistics; for each pair, missing data rates over the markers used; 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., putative half-sib, parent-offspring and unrelated pairs).

16.1 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.

16.2 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, such as other sibs and parents. Pair-wise allele-sharing is computed using a multipoint algorithm.

16.2.1 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 in 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(- β $|t|$), 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

$$Var(Y_{jk}) = \frac{1}{L_k^2} \int_0^{L_k} \int_0^{L_k} Cov(Z_{js}, Z_{jr}) dr ds = \frac{2}{\beta L_k} - \frac{2}{(\beta L_k)^2}(1 - e^{-\beta L_k}) \tag{16.1}$$

(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

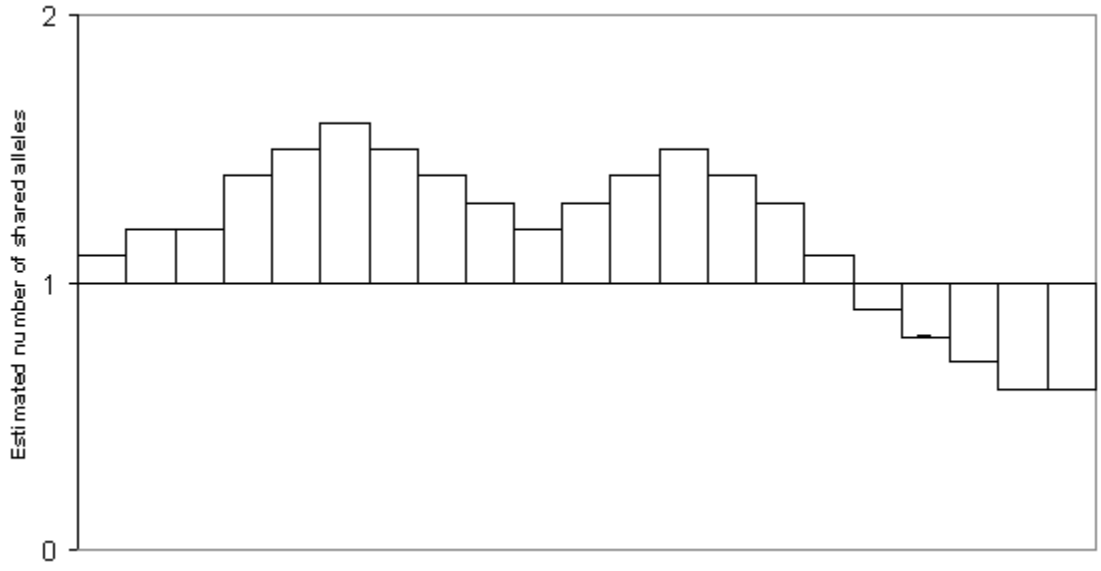


Figure 16.1: Approximate Mean-Corrected Allele Sharing

$$Y_j = (\sum_{k=1}^{22} Y_{jk}) \Big/ [\sum_{k=1}^{22} Var(Y_{jk})]^{1/2},$$

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 16.1),

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 (i.e., the width of the rectangles in Figure 16.1).

16.2.2 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(-β $|t|$), 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 16.1 with β= 0.08.

16.2.3 Incomplete Marker Information

When markers are not infinitely dense and fully informative, the variances 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

$$AMIC = \sum_{p=1}^{M} r^2(s)/M,$$

where M is the total number of points over which the genome IBD probabilities are calculated and

$$r^2(s) = 1 - \frac{\sum_{i=1}^{N} \sigma_{i,residual}^2(s)}{\sum_{i=1}^{N} \sigma_{i,initial}^2} = 1 - \frac{2\sum_{i=1}^{N} \sigma_{i,residual}^2(s)}{N},$$

N is total number of sib pairs in the sample and $\sigma_{i,residual}^2(s)$ is the variance of the IBD distribution at point s for sib pair i.

16.2.4 Classification Cut Points

The best-fit regression equations for obtaining classification 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.

16.2.5 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 $\leq C_u <$ 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 $\leq C_p < Sib$

16.2.6 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 for Y_j:
   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 $\phi(.)$ 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 steps 1 and 2 for the Y_j^* obtained from all putative full sib pairs.

4. We perform the classification as described in 16.2.5 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/or

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).

16.3 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.

16.3.1 Running `reltest`

A typical run of the RELTEST program may use flags to identify the file types like the following:

```
>reltest -p data.par -d data.ped -l m.loc -g g.map
```

or, rely on a set file order like the following:

```
>reltest data.par data.ped m.loc g.map
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, `m.loc` is the name of the marker locus description file, and `g.map` is the name of the genome description file.

16.3.2 The multiple `pedigree` Block

RELTEST can read multiple pedigree data files in cases where each pedigree data file contains the markers for a single chromosome. For each pedigree data 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 data files used.

Example:

```
pedigree, file=ped1
{
.
.
.
marker=''ch1m1''
.
.
.
}
```

```
pedigree, file=ped2
{
.
.
.
marker=''ch2m1''
.
.
.
}

pedigree, file=ped3
{
.
.
.
marker=''ch3m1''
.
.
.
}
```

16.3.3 The `reltest` Block

A `reltest` block in the parameter file sets the options on how to perform an analysis using RELTEST.

The following table shows the syntax for a `reltest` parameter which starts the `reltest` block.

`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

The following table lists the parameters and attributes that may occur in a `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 suggest reclassification of pairs.
	Value Range: N/A
	Default Value: N/A
	Required: No
	Applicable Notes: 2
`, unrelated`	Cut point for between unrelated and half-sib pairs.
	Value Range: (- ∞, ∞)
	Default Value: None
	Required: No
	Applicable Notes: 2
`, hsib`	Cut point for between half-sib and full-sib pairs.
	Value Range: (- ∞, ∞)
	Default Value: None
	Required: No
	Applicable Notes: 2
`, MZtwin`	Cut point for between full-sib and MZtwins pairs.
	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 Nuclear Family" information 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, false} 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 (see 16.2.4). The program will use the generated cut points if the cut_points parameter is not specified here. If it is specified with valid attributes and values, these preset cut points will be used instead of the cut points values as given in the theory section (see 16.2.4).

   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 16.4.2

16.4 Program Output

RELTEST produces several output files that contain results and diagnostic information:

File Name	File Type	Description
reltest.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
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 Y_j and Y_j^* statistics for all pairs used in the analysis.

16.4.1 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 data file, new class, sibling classification statistic, parent offspring classification statistic and missing genotype rate. Note: incorrect reclassification 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.

Example:

```
================================================================================
  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 | -3.43113       -3.93383
      ------------------------------------------------------------------------
    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
--------------------------------------------------------------------------------
    4      4/3      HALF SIB        -2.8302     -1.6862          1% /  2%
   45      4/3      HALF SIB        -3.0415     -0.6532          5% /  3%
   58      6/5      HALF SIB        -3.1239     -1.6834          4% /  3%
   60      5/3      HALF SIB        -2.7622     -1.0231          4% /  3%
   66     31/30     HALF SIB        -2.7980     -1.1299          5% /  5%
   66     16/12     MZTWINS          8.3388      7.1082          4% / 30%
  118      4/3      HALF SIB        -2.2236     -1.6296          5% /  4%
  159      5/3      HALF SIB        -4.2079      0.0942          8% /  8%
--------------------------------------------------------------------------------
 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
.
.
.
```

16.4.2 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.

Example:

```
================================================================================
  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 should be
                    reclassified.
    Analysis Name : default_analysis
================================================================================
                    reclassified
  pid      pair     pair type         Yj          Yj*          missing data
--------------------------------------------------------------------------------
    4       4/3      HALF SIB        -2.8302     -1.6862          1% /  2%
   45       4/3      HALF SIB        -3.0415     -0.6532          5% /  3%
   58       6/5      HALF SIB        -3.1239     -1.6834          4% /  3%
   58      15/14     FULL SIB         2.0909     -0.6428          2% /  2%
   60       5/3      HALF SIB        -2.7622     -1.0231          4% /  3%
   66      25/24     FULL SIB        -0.0537     -0.1758         15% /  7%
   66      29/28     FULL SIB         0.2178     -0.1804          5% /  6%
   66      29/27     FULL SIB         1.4343      1.4365          5% /  4%
   66      28/27     FULL SIB         0.2663      1.2670          6% /  4%
   66      31/30     HALF SIB        -2.7980     -1.1299          5% /  5%
   66      16/15     FULL SIB         1.0768     -0.7570          4% / 30%
   66      16/13     FULL SIB         2.5707      1.3036          4% /  5%
   66      16/12     MZTWINS          8.3388      7.1082          4% / 30%
   66      16/11     FULL SIB         2.5809      0.7421          4% / 30%
   66      16/10     FULL SIB         1.0902     -0.0462          4% / 31%
   66      15/13     FULL SIB         0.2244     -0.5510         30% /  5%
   66      15/12     FULL SIB         1.0914     -0.6805         30% / 30%
   66      15/11     FULL SIB         0.6307     -1.0433         30% / 30%
   66      15/10     FULL SIB         1.2501      0.3350         30% / 31%
   66      13/12     FULL SIB         1.9781      0.6856          5% / 30%
   66      13/11     FULL SIB         2.0036      0.5171          5% / 30%
.
.
.
================================================================================
```

16.4.3 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.

Example:

```
================================================================================
  RELATIONSHIP TEST PROGRAM DETAILED OUTPUT
================================================================================
    Analysis Name                      : default_analysis
.
.
.
================================================================================
PUTATIVE FULL SIB PAIRS TO BE RECLASSIFIED :
                  reclassified
  pid      pair     pair type         Yj          Yj*          missing data
--------------------------------------------------------------------------------
    1      4/3      FULL SIB    FULL SIB       2.6628    0.3874    3% /  3%
    2      4/3      FULL SIB    FULL SIB       1.8188   -0.3243    1% /  3%
    2      7/6      FULL SIB    FULL SIB       2.9981    1.1039    2% /  1%
    3      5/4      FULL SIB    FULL SIB       0.8656   -0.5446    3% /  3%
    4      4/3      FULL SIB    HALF SIB      -2.8302   -1.6862    1% /  2%
    5      5/4      FULL SIB    FULL SIB       1.4154   -2.0924    3% /  3%
    5      5/3      FULL SIB    FULL SIB       1.2640    0.1789    3% /  3%
    5      4/3      FULL SIB    FULL SIB       0.3967   -0.5439    3% /  3%
    6      5/3      FULL SIB    FULL SIB       0.4909   -0.8342    3% /  2%
    6      7/6      FULL SIB    FULL SIB       1.7456   -1.3888    2% /  2%
    7      6/4      FULL SIB    FULL SIB       0.5844   -1.1940    1% /  2%
    7      6/3      FULL SIB    FULL SIB       1.5012   -0.4603    1% /  1%
    7      4/3      FULL SIB    FULL SIB       1.4191   -1.3774    2% /  1%
    7      8/7      FULL SIB    FULL SIB       1.8528    0.5990    1% /  1%
    8      8/7      FULL SIB    FULL SIB       1.5790   -0.4496    3% /  1%
    8      8/6      FULL SIB    FULL SIB       0.7129   -0.2272    3% /  2%
    8      8/5      FULL SIB    FULL SIB      -0.0047   -0.3939    3% /  1%
    8      8/4      FULL SIB    FULL SIB       0.5248   -1.7989    3% /  3%
.
.
.
--------------------------------------------------------------------------------
 total putative  pairs          : 342
 total pairs to be reclassified : 8
.
.
.
```

Chapter 17

SEGREG

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 latter for autosomal linkage only). The most significant improvements over the programs REGC, REGD and REGTL of the early versions of S.A.G.E., all of which are now subsumed in SEGREG, 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.

17.1 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, though this is very likely. Also, as with the previous segregation analysis programs, situations can occur in which it is not numerically possible to calculate the variance-covariance matrix of the estimates.

All covariates are centered prior to entering any analysis and the means used to do this are displayed at the beginning of the output. However, these means are based on all the data available in the pedigree data file, whereas any particular analysis uses only those records informative for all variables

relevant to the analysis. Thus the centering will only be exact when none of the covariates used in the analysis have missing values.

Whenever a model is maximized, the corresponding ln likelihood and -(twice the ln likelihood) are given for the estimated model. However, these values differ from the true values by a constant that is the same for all analyses performed in the same SEGREG run, but might differ, for the same data, in separate SEREG runs.

Although X-linked traits can now be analyzed under the assumption of equal allele frequencies in the two sexes, be aware that the main purpose of including X-linkage in SEGREG is to produce an X-linkage model to be used in LODLINK; for this purpose SEGREG will appropriately estimate all the parameters of a given X-linked model, but when X-linkage is specified in the transmission sub-block. the distribution of the test statistics produced at the end of a SEGREG run are not all as stated in the theory section for autosomal inheritance.

17.2 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:

$$\sum_u \psi_u = 1.$$

However, for hemozygous males, there are only two types, u = A, B, and $\psi_A + \psi_B = 1$.

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:

$$\psi_{AA} = q_A^2;\ \psi_{AB} = 2q_A(1-q_A);\ \psi_{BB} = (1-q_A)^2,$$

$$\psi_A = q_A;\ \psi_B = (1-q_A).$$

[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).

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 $\tau_{AA} = 1$, $\tau_{AB} = 0.5$, and $\tau_{BB} = 0$; or $\tau_A = 1$ and $\tau_B = 0$ to only female offspring. 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 (for the father) and u_M (for the mother) produce offspring of type u. Transition probabilities assume random mating and hence are determined by the transmission probabilities as follows:

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

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, A, B. 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
2. τ_{AB} must be a specific function of τ_{AA}, τ_{BB} and the allele frequency q_A for autosomal transmission, and other constraints are necessary for X-linked transmission (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 for autosomal transmission (some of which are not appropriate for X-linked transmission) as shown in the table below (Self and Liang, 1987). In this table, the following abbreviations are used to describe the models:

[3] ditto

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

Abbreviation	Meaning
no_trans	$\tau_{AA} = \tau_{AB} = \tau_{BB}$, $\tau_A = \tau_B$
homog_no_trans	$\tau_{AA} = \tau_{AB} = \tau_{BB} = \tau_A = \tau_B = q_A$
homog_mendelian	$\tau_{AA} = 1$, $\tau_{AB} = .5$, $\tau_{BB} = 0$, $\tau_A = 1$, $\tau_B = 0$
homog_general	$0 \leq \tau_{AA}$, τ_{BB}, τ_A, $\tau_B \leq 1$ τ_{AB}= $(q_A - q_A^2\tau_{AA} - (1-q_A)^2\tau_{BB})/2q_A(1-q_A)$
tau_ab_free	$\tau_{AA} = 1$, $0 \leq \tau_{AB} \leq 1$, $\tau_{BB} = 0$, $\tau_A = 1$, $\tau_B = 0$
general	$0 \leq \tau_{AA} = \tau_A$, τ_{AB}, $\tau_{BB} = \tau_B \leq 1$

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

See also note 2 of the transmission sub-block (17.3.2.12).

In the description of the models from here on, X-linkage is largely ignored. It is assumed that $\beta_{AA} = \beta_A$ and $\beta_{BB} = \beta_B$ and, when the trait is indicated as being X-linked, the segreg.typ file (see 17.4) is approperiate for a trait/marker in LODLINK.

17.2.1 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 quantitative traits, regressive multivariate logistic models for binary traits, the finite polygenic mixed model, and models for binary traits with variable age of onset. Because SEGREG analyzes all constituent pedigrees as being independent, in the rest of this chapter we simply use the word " pedigree" to mean "consituent pedigree".

17.2.1.1 Ascertainment: 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 desireable 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

[5]The pedigree sampling frame can include pedigree members for whom the trait value is missing, in which case calculation will proceed as in Ginsberg et al. (2003). This is not advisable, however. A better strategy is to replace these missing values by the average of the observed (i.e., non-missing) trait values.

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 special cases. In the case of simple single ascertainment, the pedigree proband sampling frame for each pedigree comprises only the proband. See Ginsberg et al (2006) 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.

17.2.1.2 Type Probabilities and Penetrance Functions

Given a model with established parameter values, we estimate the posterior 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 and
- u_i is the type of individual i.

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 also prepare files of penetrance functions that can be used as input into LODLINK and MLOD (the latter for autosomal linkage only) using maximum likelihood estimates of all unknown parameters. These are of the form $\Pr(t_i|u_i)$, where t_i is the analysis trait (see 17.2.2.2).

17.2.2 Regressive Models for Quantitative 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 a quantitative 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 quantitative traits are implemented in SEGREG. Class A models assume that sibling subtypes 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 quantitative 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 correlations among the residuals from the type means can be 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 any two siblings (in 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), with the result that, for example, the residual grandparent-grandchild correlation is assumed to be zero if the intervening parent has a missing phenotype.

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.

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.

17.2.2.1 Composite Trait

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

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

where the parameters κ_i may be estimated.

17.2.2.2 Transformation of the Trait

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

The first possible transformation is:

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

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 trait and covariate values (nothing missing). This is the standardized Box and Cox (1964) transformation with power parameter λ_1 and shift parameter λ_2.

The second possible transformation is is the standardized generalized modulus power transformation (George and Elston, 1988) with power parameter λ_1 and shift parameter λ_2 (see 5.2.2).

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

17.2.2.3 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's analysis trait 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

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

Note we assume $\sigma_A^2 = \sigma_{AA}^2$ and $\sigma_B^2 = \sigma_{BB}^2$; any sex difference must be allowed for by adding a sex covariate.

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

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

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

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

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

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

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, let

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

and

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

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

Then under random mating the likelihood for a randomly sampled pedigree is

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

17.2.2.4 Allowing for Ascertainment

Ascertainment is allowed for as indicated in 17.2.1.1. In order to condition on traits 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 17.2.1.1 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

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

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.

17.2.3 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 trait 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 trait 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 trait transformation is 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 quantitative traits, are incorporated into the model. These are denoted in 17.2.4 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 order correlations are zero. The actual correlations are calculated from these associations measures for specific logit values [see Karunaratne and Elston (1988)]. Note that, because all covariate values are centered, the logit values are at the sample average value of all covariates.

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.

17.2.3.1 Likelihood for a Randomly Sampled Nuclear Family

Let t_F, t_M and t_i be the traits 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

$$\prod_{i=F,M,1}^{n} \frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}} \{1+ \delta_{FO}\left(1-\frac{e^{\theta_u(F)t_F}}{1+e^{\theta_u(F)}}\right)\sum_{i=1}^{n}(-1)^{t_F+t_i}\left(1-\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\right)$$

$$+\delta_{MO}\left(1-\frac{e^{\theta_u(M)t_M}}{1+e^{\theta_M(u)}}\right)\sum_{i=1}^{n}(-1)^{t_M+t_i}\left(1-\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\right)$$

$$+\delta_{SS}\sum_{1\le i<j\le n}(-1)^{t_i+t_j}\left(1-\frac{e^{\theta_u(i)t_i}}{1+e^{\theta_u(i)}}\right)\left(1-\frac{e^{\theta_u(j)t_j}}{1+e^{\theta_u(j)}}\right)$$

$$+(-1)^{t_M+t_F}\delta_{MF}\left(1-\tfrac{e^{\theta_u(M)t_M}}{1+e^{\theta_u(M)}}\right)\left(1-\tfrac{e^{\theta_u(F)t_F}}{1+e^{\theta_u(F)}}\right)\Big\}.$$

17.2.4 Finite Polygenic Mixed Model

The finite polygenic mixed model (Fernando et al, 1994; Lange, 1997) can be used for either quantitative 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 trait in the case of a quantitative trait. As for regressive models for quantitative traits, the type mean and/or variances can depend on covariates. For a quantitative 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 in the finite polygenic mixed model we define the penetrance function for a quantitative 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

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

17.2.4.1 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

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

17.2.5 Binary Traits with Variable Age of Onset

When using the command line, SEGREG requires specification of the fpmm for a binary trait with variable age of onset, but it is then possible to specify 0 polygenic loci. When using the GUI it is possible to chose "binary trait whith variable age of onset" without specifying the fpmm.

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 the susceptibility γ, can each be made dependent on covariates and/or type, in the same way as for a quantitative 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. 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 = $\frac{\pi^2}{3\alpha^2}$.

The mean and variance of the age of onset distribution 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 $\gamma = 1$). However, the variance of the logistic distribution depends on the mean, and so it is not permissible for the mean and variance to depend on the same covariate.

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 in the case of the finite polygenic mixed model. 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, age on onset 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.

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.

17.3.1 Running `segreg`

A typical run of the SEGREG program may use flags to identify the file types like the following:

```
>segreg -p data.par -d data.ped
```

or, rely on a set file order like the following:

```
>segreg data.par data.ped
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file.

17.3.2 The `segreg` Block

A `segreg` block in the parameter file sets the options on how to perform an analysis using SEGREG. The following table shows the syntax for a `segreg` parameter which starts the `segreg` block.

parameter [, attribute]	Explanation	
segreg	Starts a SEGREG 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	segreg_analysis*k* where *k* = 1, 2 ... *n* for a set of *n* analysis
	Required	No
	Applicable Notes	None

The following table lists the parameters and attributes 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: SEGREG Analysis *k*, where *k* = 1, 2, ..., *n* for a given set of *n* specified SEGREG analyses. Required: No Applicable Notes: 1
`trait`	Specifies the name of a primary trait. Must be the name of a trait or covariate in the 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 quantitative binary, if trait is binary Required: Sometimes. See note 2. Applicable Notes: 2
`composite_trait`	Starts a sub-block for specifying composite trait covariates. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 3
`type_mean`	Starts a sub-block for specifying type means. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 4, 19
`type_var`	Starts a sub-block for specifying type variances. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 5, 19
`type_suscept`	Starts a sub-block for specifying type susceptibilities or penetrances. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 6, 20

`mean_cov`	Starts a sub-block for specifying covariates for the mean. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 7, 19
`var_cov`	Starts a sub-block for specifying covariates for the variance. Value Range: N/A Default Value: N/A 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: N/A Required: No Applicable Notes: 9, 20
`class`	Specifies the model class Value Range: {A, D, FPMM, MLM} Default Value: D, for quantitative traits MLM, for binary traits FPMM, for age-of-onset traits Required: No Applicable Notes: 10
`fpmm`	Starts a sub-block for specifying a FPMM model. Value Range: N/A Default Value: N/A Required: No Applicable Notes: 10
`resid`	Starts a sub-block for specifying residual correlations (or associations). Value Range: N/A Default Value: N/A Required: No Applicable Notes: 11
`transformation`	Starts a sub-block for specifying transformation options. Value Range: N/A Default Value: N/A 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: N/A Required: No Applicable Notes: 13
`transmission`	Starts a sub-block for the specifying the transmission model Value Range: N/A Default Value: N/A Required: No Applicable Notes: 14
`ascertainment`	Starts a sub-block for specifying the pedigree ascertainment options. Value Range: N/A Default Value: N/A 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: N/A 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: N/A Required: No Applicable Notes: 17
`output_options`	Starts a sub-block for specifying output options. Value Range: N/A Default Value: N/A 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 required if a trait with variable age of onset is being analyzed, or a binary trait is to be analyzed as a quantitative 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 or trait (other than the primary trait) listed in the data file or created by means of a function block. Note: This sub-block is not applicable to binary traits.

4. This sub-block refers to means of quantitative 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 quantitative 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 (i.e., only one type susceptibility is estimable) when either the **no_trans** or **homog_no_trans** option for transmission is used unless the model includes non-zero residual associations.

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 quantitative traits and a multivariate logistic model is used for binary traits. **FPMM** is the option to choose for binary traits with variable age of onset when using the command line, even if zero polygenic loci are desired.

11. This sub-block is not relevant for the FPMM (finite polygenic mixed model). Residual correlations are relevant for quantitative 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 quantitative traits.

17.3.2.1 The `composite_trait` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 or covariate in the pedigree data file or 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	Yes, if `fixed` is set to true.
	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). A particular trait may not be specified as both a mean covariate and as a composite trait covariate.

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.

17.3.2.2 The `type_mean` Sub-Block

The following table lists the parameters and attributes that may occur in a `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, 3,4
mean	Specifies the mean 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	Yes, if `fixed` is set to true.
	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 quantitative 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), and the output will include results for all three types. 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 quantitative 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	$\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$

Note: If any of the options **three_add**, **three_inc** or **three_dec** is specified, the output will also include the results for options **one** and **two_dom** or **two_rec**, depending on which has the higher likelihood.

For example,

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

sets initial estimates $\beta_{AA} = \beta_{AB}$= -1.0 and β_{BB}= -2.0 when estimating $\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$.

4. If X-linkage is specified in the transmission sub-block (17.3.2.12), β_{AA} and β_{BB} are also the means of the hemizygous males.

17.3.2.3 The `type_var` Sub-Block

The following table lists the parameters and attributes that may occur in a `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: 1,2,3
var	Specifies the variance 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: 2,3
, val	Specifies value of the variance Value Range: $[0, \infty)$ Default Value: None Required: Yes, if `fixed` is set to true. 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 quantitative 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. When used in conjunction with the logistic density function for an age of onset

distribution, only one variance is possible regardless of the number of types 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 fixes values $\sigma^2_{BB} = \sigma^2_{AB}$=30.0, when estimating only σ^2_{AA}.

3. If X-linkage is specified in the transmission sub-block (17.3.2.12), σ^2_{AA} and σ^2_{BB} are also the valiances of the hemizygous males.

17.3.2.4 The `type_suscept` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 logit 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, 4,5
`, val`	Specifies value of a given mean.	
	Value Range	(- ∞, + ∞)
	Default Value	None
	Required	Yes, if `fixed` is set to true.
	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, θ, of penetrances, γ, (of susceptibilities for a binary trait with variable age of onset).

$$\gamma = \frac{e^{\theta}}{1+e^{\theta}}$$

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	$\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$

If option is for more than one type effect (eg., **two**, **three**, **two_dom**, etc.) and the value of one of them is fixed, then initial values must be specified for all susceptibilities.

For example,

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

sets initial estimates $\beta_{AA} = \beta_{AB}$= -2.0 and β_{BB} = -1.0 when estimating $\beta_{AA} \leq \beta_{AB} \leq \beta_{BB}$ in the logit $\theta_u(i)$ as described in 17.2.3.

4. If X-linkage is specified in the transmission sub-block (17.3.2.12), β_{AA} and β_{BB} are also the means of the hemizygous males.

17.3.2.5 The `mean_cov` Sub-Block

The following table lists the parameters and attributes that may occur in a `mean_cov` sub-block.

parameter [, attribute]	Explanation
`covariate`	Covariate to modify the mean of a quantitative 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 or covariate from the data file or a name created by means of a `function` block.
	Default Value: None
	Required: No
	Applicable Notes: 1, 2
`, val`	Specifies the value of the covariate coefficient.
	Value Range: (- ∞, + ∞)
	Default Value: None
	Required: Yes, if `fixed` is set to true.
	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 are to be estimated.
	Value Range: {true, false}
	Default Value: false
	Required: No
	Applicable Notes: 3

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. A covariate specified here cannot be also specified in the `composite_trait` sub-block. All covariates are centered, the centering (average) value being included as part of the output. This sub-block is only relevant for quantitative traits.

2. 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 age of onset a logistic density function

is used, and so the same covariate cannot be specified to modify both the mean and the variance; nor can the same covariate be used to modify both the mean and the susceptibility.

3. 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 "mean" 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.

17.3.2.6 The `var_cov` Sub-Block

The following table lists the parameters and attributes that may occur in a `var_cov` sub-block.

parameter [, attribute]	Explanation
`covariate`	Covariate to modify the variance (conditional on type) of a quantitative trait. This parameter may be specified multiple times.
	Value Range: Character string representing the name of a trait or covariate in the pedigree data file or 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: Yes, if `fixed` is set to true.
	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 quantitative 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, because a logistic age of onset distribution is assumed, 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 "variance" coefficient for each covariate. If there is interaction, then for this "variance" 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.

17.3.2.7 The `suscept_cov` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 or covariate in the pedigree data file or 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	Yes, if `fixed` is set to true.
	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 a covariate for both `type_mean` and the `type_suscept`.

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 "susceptibility/penetrance" covariate coefficient for each covariate. If there is interaction, then for this "susceptibility/penetrance" 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.

17.3.2.8 The `fpmm` Sub-Block

The following table lists the parameters and attributes that may occur in a `fpmm` sub-block.

parameter [, attribute]	Explanation
`loci`	Specifies number of polygenic loci. Value Range: {0,1, 2, 3, 4, 5} Default Value: 3 Required: No Applicable Notes: 1
`freq`	Specifies allele frequency of the polygenic loci. Value Range: (0, 1) Default Value: 0.5 Required: No Applicable Notes: 1, 4
`var`	Specifies the polygenic variance. 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: Yes, if `fixed` is set to true. 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 3 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, 5

Notes

1. These parameters cannot be estimated, only specified.
2. The following is an example of an `fpmm` sub-block

```
fpmm
{
   loci=6
   freq=.4
   var, val=10.3, fixed=false
}
```

3. This sub-block, nested within the `fpmm` sub-block, may be used 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.

4. This option is not applicable if the value of `loci` is set to zero.

5. The `onset` parameter is required for a binary trait with specified (or estimated) age of onset, and is not relevant otherwise.

17.3.2.8.1 The `onset` Sub-Block

The following table lists the parameters and attributes that may occur in a `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
multi_dependent	Specifies what is dependent on a polygenic component when using the FPMM. Value Range: {N, A, S} Default Value: N Required: No, inapplicable if loci=0. Applicable Notes: 2
age_onset	Specifies the age of onset. Value Range: Name of a quantitative trait or covariate that is either listed in the pedigree data file or created by means of a function block. Default Value: None Required: Yes Applicable Notes: 3
age_exam	Specifies the age at exam. Value Range: Name of a quantitative trait or covariate that is either listed in the data file or created by means of a function block. Default Value: None Required: Yes Applicable Notes: 3

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. The multi_dependent values have the following meanings (must be skipped if loci=0):

   N – There is no polygenic component.

   A – Age of onset has a polygenic component.

   S – Susceptibility has a polygenic component.

   If you choose the default option **N**, neither the age of onset nor the susceptibility depend on the number of polygenic loci.

3. It is permissible for the `age_onset` and `age_exam` parameters to specify the same quantitative 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. This should only be done if the age given for an affected person is the age of onset or unknown, i.e. this disallows the possibility of an affected person having an age at examination when age of onset is unknown. If an affected person has an age of exam but no age of onset, this information cannot be used when age of onset and age of exam are in the same field, so in that situation the result will be different from the result obtained if two separate fields are used.

   Example of an onset sub-block nested within 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=sidease
        age_onset=age
        age_exam=age
    }
}
```

17.3.2.9 The `resid` Sub-Block

The following table lists the parameters and attributes that may occur in a `resid` sub-block (see note 1).

parameter [, attribute]	Explanation
`option`	Specifies residual familial correlations/associations. Value Range: equal_po_ss, equal_po, arb Default Value: equal_po_ss Required: No Applicable Notes: 2, 3
`fm`	Specifies the correlation/association 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 quantitative. traits; N/A, for binary traits Default Value: 0 Required: Yes, if `fixed` is set to true. 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/association 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 quantitative. traits N/A, for binary traits Default Value: 0 Required: Yes, if `fixed` is set to true. 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/associations 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 quantitative. traits N/A, for binary traits Default Value: 0 Required: Yes, if `fixed` is set to true. 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/associations 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 quantitative. traits N/A, for binary traits
	Default Value	0
	Required	Yes, if `fixed` is set to true.
	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 quantitative traits and residual associations are relevant for binary traits.

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 quantitative data. In the multivariate logistic model used for non-quantitative data with residuals, the range is a calculated value that changes based on parameter estimates.

17.3.2.10 The `transformation` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 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: Yes, if `fixed` is set to true. 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	Yes, if `fixed` is set to true.
	Applicable Notes	See note 2 of the `composite_trait` sub-block.
, fixed	Option to fix λ_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. The Box and Cox transformation is given in 17.2.2.2 and the George and Elston transformation is given in 5.2.2.. For the Box-Cox transformation, all values of the trait to which it is applied must be$> -\lambda_2$, and should preferably be $> 1 - \lambda_2$.

2. The default values $\lambda_1 = \lambda_2 = 1$ result in no transformation when the Box-Cox transformation is applied, and The default values $\lambda_1 = 1$; $\lambda_2 = 0$ result in no transformation when the George-Elston transformation is applied, provided the trait values are all < 1 or all > 1.

17.3.2.11 The `geno_freq` Sub-Block

The following table lists the parameters and attributes that may occur in a `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: None

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.

17.3.2.12 The `transmission` Sub-Block

The following table lists the parameters and attributes that may occur in a `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
`model`	Option to specify the inheritance model.	
	Value Range	{autosomal, X-liked}
	Default Value	autosomal
	Required	No
	Applicable Notes	None
`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	5
`, val`	Specifies a value for the parameter.	
	Value Range	[0, 1]
	Default Value	None
	Required	Yes, if `fixed` is set to true.
	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: N/A 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_u to be the probability that a person of type u transmits A, and q_A to be the relative frequency of A, these options correspond to:

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

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. Note that the hemizygous genotypes A and B are not in the value range. AA and BB are also the values of hemizygous males for X-linked model.

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].

17.3.2.13 The `ascertainment` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 or covariate, binary or quantitative, in the 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 quantitative trait or covariate 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 – the same as not including this sub-block).
   - 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. Because founders do not include singletons (see 3.1), this option should not be used if there are any singletons in the data.
   - 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. Because founders do not include singletons (see 3.1), this option should not be used if there are any singletons in the data.

2. The `cond_val` parameter is relevant for quantitative traits only, and is ignored for binary traits, composite traits, and for age of onset models (for which the onset_option parameter in this sub-block 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 analysis trait value.
gte_thresh	Indicates that conditioning is on the analysis trait value being greater than or equal to a threshold value.
lte_thresh	Indicates that conditioning is on the analysis 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 trait 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 trait 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.

17.3.2.14 The `prev_constraints` Sub-Block

The following table lists the parameters and attributes that may occur in a `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: N/A Required: No Applicable Notes: 17.3.2.14.1

17.3.2.14.1 The `constraint` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 or covariates in the pedigree data file or created by means of a function block. This parameter may be specified multiple times but need not be included if the prevalence is to be constrained at the mean value. 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: No Applicable Notes: 2
`R`	Specifies the number of affected persons in a random sample. Value Range: (0, N) Default Value: None Required: Yes Applicable Notes: 3, 5

N	Specifies the sample size.	
	Value Range	(R, $+\infty$)
	Default Value	None
	Required	Yes
	Applicable Notes	3, 5
age	Specifies the age covariate (age of onset or age at exam) at which cumulative prevalence (probability of having been affected since birth) should be computed.	
	Value Range	(0, $+\infty$)
	Default Value	None
	Required	Required for age of onset traits, if mean age at onset is not to be used, and disallowed otherwise.
	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. If for a particular specified value of the covariate we have an independent estimate of the prevalence, p, with standard error s.e., then appropriate values of N and R are

$$N = \frac{p(1-p)}{(s.e.)^2}$$

and

$$R = Np.$$

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
         }
         .
         .
         .
      }
   }
```

17.3.2.15 The `prev_estimate` Sub-Block

The following table lists the parameters and attributes that may occur in a `prev_estimate` sub-block.

`parameter [, attribute]`	**Explanation**
`covariate`	Specifies a covariate on which prevalence depends. Allowable values are the names of traits or covariates in the pedigree data file or created by means of a function block. This parameter may be specified multiple times, but need not be included if the prevalence is to be constrained at the mean value. 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, if mean age is not to be used, and disallowed otherwise. Value Range: (0, +∞) 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 as indicated in the output. This mean is the average of all the values of the covariate in the sample. Note that if the particular SEGREG run uses other covariates with missing values, causing some individuals to be excluded from the likelihood for that run, the average of the covariate values used for that run may be different from its average in the whole sample.

3. The literal string `infinity` may be entered to indicate an " infinite" age.

17.3.2.16 The `output_options` Sub-Block

The following table lists the parameters and attributes that may occur in a `output_options` sub-block.

`parameter` [, `attribute`]	**Explanation**
`type_prob`	Specifies option to calculate type probabilities and penetrance functions. Value Range: {true, false} Default Value: false Required: No Applicable Notes: 1

Notes

1. Type probabilities can only be calculated if two or three types are specified in either the `type_mean` sub-block or the `type_susceptibility` sub-block. In either case (1) 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; and (2) for each individual, penetrance functions that can be used as input to LODLINK or MLOD (for autosomal linkage only). Because (2) usually only makes sense if the `transmission` option **homog_mendelian** has been chosen, these penetrances will only be produced if that option is among those chosen in the `transmission` sub-block.

17.4 Program Output

SEGREGproduces several output files that contain results and diagnostic information:

File Name	File Type	Description
segreg.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
segreg.sum	SEGREGsummary output file	Contains the table of final estimates of the parameters and their standard errors, and other results.
segreg.det	SEGREGdetailed output file	Contains the table of final estimates and variance-covariance matrix of the parameter estimates.
segreg.typ segreg.typX	Trait genotype probability and penetrance function output file	Contains the individual specific type probabilities conditional on the model and all the pedigree information available, and individual specific penetrance information. The latter will only be produced if the **homog_mendelian** option of the `transmission` parameter has been enabled and is suitable for input into model-based linkage programs such as MLOD (for autosomal models only) and LODLINK (for both autosomal and X-linked models). Note that .typX file must be used for X-linkage analysis in LODLINK along with .loc2 file from FREQ (8.4).

17.4.1 Summary Output File

The SEGREG summary output file stores the table of final estimates of the parameters with model information.

Example:

```
================================================================================================
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
===========================================================================================
```

17.4.2 Detailed Output File

The SEGREG Detailed output file stores the variance-covariance matrix as well as what the Summary File has. At the end of the analysis the ln likelihood and -(twice the ln likelihood) are given for the estimated model. Note that these values differ from the true values by a constant that is the same for all analyses performed in the same SEGREG run, but might differ, for the same data, in separate SEGREG runs.

Example:

```
===========================================================================================
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
===========================================================================================
```

The Detailed Output file also includes a table of p-values for the various transmission models estimated as shown in Figure17.1, below. ***This table is not included as part of the output for X-linked models***.

```
=============================================================================================
    Likelihood Ratio Criteria(above diagonal) and Asymptotic P-values(below diagonal)
               Rows: null hypothesis   Columns: model (alternate hypothesis)
               homo_no_trans   homo_mendelian  homo_general    tau_AB_free     general
---------------------------------------------------------------------------------------------
homo_no_trans  -----           -----           119.8[2]        -----           120.6[3]
homo_mendelian -----           -----           0.000[2]        0.812[1]        0.812[3]
homo_general   0.000           0.750           -----           -----           0.812[1]
tau_AB_free    -----           0.184           -----           -----           0.000[2]
general        0.000           0.637           0.368           1.000           -----
=============================================================================================
Note: The quoted P-values assume large samples and that the difference in the number of
functionally independent parameters estimated in the two models is as indicated in
bracket []. Because bounds placed on other parameters in the model used may result in a
number that is different, you are cautioned to check this before quoting the corresponding
large-sample P-value.
```

Figure 17.1: Transmission Model P-Values

Chapter 18

SIBPAL

This linkage program models trait data from sib pairs as a function of marker allele sharing identity-by-descent (IBD). Although this program has been extended to allow for the inclusion of half-sibs in a manner similar to that suggested by Schaid et al. (2000), users are advised to use RELPAL for this situation. Available analyses can use both single- and multi- point IBD information, and models allow for both binary and quantitative traits due to multiple genetic loci, including epistatic interaction and covariate effects. Many options are available for binary traits, including a generalization of the mean test and the proportion test. Like the original SIBPAL, it uses linear regression and hence is extremely fast.

18.1 Limitations

The Haseman-Elston linkage test in this release only supports the univariate analysis of full and half sibling pairs. Full support for multivariate analysis and using other relative pairs is available in the RELPAL program.

Unlike early 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.

18.2 Theory

18.2.1 Basic Notation

A nuclear family is a set of two individuals who have a mating relationship and their natural children; these children form a full sibship. For this chapter, we define a family to be a connected set of both full sibs and half sibs in a single generation (referred as cluster/string in Schaid et al. (2000)).

Let the number of families in the analysis be K.

Let the number of sib pairs (full or half) in the k -th family be n_k, $k = 1, 2, ..., K$.

Let j be the index of a sib pair: $j = 1, 2, ..., \sum_k n_k = n$, where n is the total number of sib pairs.

Conditional on the marker information available, at a particular genomic location let $\hat{f}_{1j}$ be the probability of sharing 1 allele IBD, and $\hat{f}_{2j}$ be the probability of sharing 2 alleles IBD for the j -th sib pair. Note that $\hat{f}_{2j} = 0$ in the case of half sib pairs.

Let $\pi = (1+2w_1)/4$ and $\hat{\pi}_j = \hat{f}_{2j} + w_1\hat{f}_{1j}$ where $0 \le w_1 \le 0.5$ (Whittemore and Tu, 1998), for the j -th sib pair. The current default value of w_1 is 0.5, corresponding to the mean test for a binary trait.

18.2.2 Test of Mean Allele Sharing

Currently all tests of mean allele sharing are done separately for full and half sib pairs. We first obtain estimates of the means of the $\hat{\pi}_j$ and $\hat{f}_{ij}\,(i = 0, 1, 2)$, which we 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, when $w_i = 0.5$, $\pi = 0.5$ and $(f_0, f_1, f_2) = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ *for a random sample* of full sib pairs and $\pi = 0.25$ and $(f_0, f_1, f_2) = (\frac{1}{2}, \frac{1}{2}, 0)$ *for a random sample* of half sib pairs. These means and their variances are estimated by calculating:

$$\bar{\hat{\pi}} = \frac{1}{\mathrm{n}}\sum_j \hat{\pi}_j \qquad s^2_{\bar{\hat{\pi}}} = \frac{1}{\mathrm{n(n-1)}}\sum_j \left(\hat{\pi}_j - \bar{\hat{\pi}}\right)^2,$$

$$\bar{\hat{f}}_i = \frac{1}{\mathrm{n}}\sum_j \hat{f}_{ij} \qquad s^2_{\bar{\hat{f}}_i} = \frac{1}{\mathrm{n(n-1)}}\sum_j \left(\hat{f}_{ij} - \bar{\hat{f}}_i\right)^2.$$

From each mean, a t statistic is computed and referred to the t distribution with n -1 d.f. for a two-sided test. The p-values are

$$2\mathrm{P}\left(t_{n-1} \ge \frac{\left|\bar{\hat{\pi}} - \pi\right|}{s_{\bar{\hat{\pi}}}}\right) \quad \text{and} \quad 2\mathrm{P}\left(t_{n-1} \ge \frac{\left|\bar{\hat{f}}_i - f_i\right|}{s_{\bar{\hat{f}}_i}}\right),$$

and where t_{n-1} is a random variable that is distributed as t with *n-1* d.f.

18.2.2.1 Test of Mean Allele Sharing for Binary Traits in Selected Pairs

The above tests are also 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

$$\mathrm{P}\left(t_{n-1} \ge \frac{\delta\left[\bar{\hat{\pi}} - \pi\right]}{s_{\bar{\hat{\pi}}}}\right) \quad \text{and} \quad P\left(t_{n-1} \ge \frac{\delta\left[\bar{\hat{f}}_i - f_i\right]}{s_{\bar{\hat{f}}_i}}\right),$$

where $\delta = 1$ for concordantly affected pairs (2 affected members) and unaffected pairs (0 affected members), and $\delta = -1$ for discordant pairs (1 affected member). No such tests are performed for full sib pairs if i = 1.

18.2.3 Generalized Haseman and Elston Linkage Test

18.2.3.1 Regression model for autosomal markers

The basic model we fit for autosomal markers is of the form:

$$y = \beta_0 + \sum_m a_m \hat{\pi}_m + \sum_m d_m \hat{f}_{2m} + \sum_c b_c f(z_c) + e$$

where

- y is a dependent variable (see 18.2.3.2),
- β_0 is the intercept,
- $\hat{\pi}_m = \hat{f}_{2m} + w_1 \hat{f}_{1m}$ where the current default value of w_1 is 0.5,
- b_c is a nuisance parameter accounting for the effect of some function f of the c -th covariate term z_c,
- e is the residual error.

In a random sample, when w_1 is 0.5, a_m is the additive genetic variance and d_m is the dominant genetic variance due to the m -th autosomal marker. In the case of an autosomal locus, the variances a_m and d_m are the trait locus-specific variances, attenuated by the recombination fraction between the trait and marker loci. Let β be the parameter vector for such a linear model. Then the generalized least squares estimator of β based on the above model is

$$b = (A'W^{-1}A)^{-1}A'W^{-1}y$$

and that of the residual variance is

$$s^2 = \frac{y'W^{-1}(y - Ab)}{n - m}$$

where m is the number of parameters estimated, n is the number of sib pairs, y is an $n \times 1$ vector of dependent variables with transpose $y' = (y_1, y_2, ..., y_n)$, W is the $n \times n$ weight matrix for y , and A is an $n \times m$ design matrix - each parameter corresponds to a particular column of A . Note that the weight matrix W is either a correlation matrix R or the residual variance matrix Σ of y , depending on the method used to generate the dependent variable y (see18.2.3.5).

18.2.3.2 Dependent variable y

Let the number of all sibs in the analysis be N and the trait values be $x_1, x_2, ..., x_N$.

In the case of a binary trait, $x_i = 1$ for an affected individual and 0 for an unaffected individual, and it is then treated the same way as for any other quantitative trait to obtain the dependent variable.

Let the number of sibs in the k-th full sibship be N_k, $k = 1, 2, ..., K'$ (Note: $K' \geq K$).

Let $\bar{x}$ be the mean of the trait x, calculated in one of four possible ways. It can be:

1. the sample mean estimated from all the data as $\tilde{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$,
2. the population mean or any other value set by the user,
3. the sibship specific mean $\tilde{x}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} x_i$. For an individual who has no full sibs, we set $\bar{x} = \tilde{x}$,
4. the best linear unbiased predictor (BLUP) of the sibship mean, $w\tilde{x}_k + (1-w)\tilde{x}$, where $w = \frac{\sigma_b^2}{\sigma_b^2 + \frac{\sigma_r^2}{N_k}}$. Here, σ_b^2 is the variance among full sibships and σ_r^2 is the variance within full sibships. For an individual who has no full sibs, we set $w = 0$.

The default mean is the BLUP of the sibship mean.

In the following, i is the index of a sib and j is the index of a sib pair. Then the dependent variable for the j-th sib pair can be:

$$y_j = \begin{cases} (x_{j1} - \bar{x})(x_{j2} - \bar{x}) & \text{mean-corrected cross-product} \quad \text{(PROD)} \\ -\frac{1}{2}\left[(x_{j1} - \bar{x}) - (x_{j2} - \bar{x})\right]^2 = -\frac{1}{2}\left[x_{j1} - x_{j2}\right]^2 & -\frac{1}{2}(\text{squared pair trait difference}) \quad \text{(DIFF)} \\ \frac{1}{2}\left[(x_{j1} - \bar{x}) + (x_{j2} - \bar{x})\right]^2 & \frac{1}{2}(\text{squared mean-corrected trait sum}) \quad \text{(SUM)} \\ \textit{weighted combination of squared pair trait difference \& squared mean - corrected trait sum} & \text{used for options W2, W3, W4} \end{cases}$$

In the case of the weighted option W2, the dependent variable for the j-th sib pair is computed in the following way:

$$y_j = \frac{\frac{1}{2}\{s_s^2[(x_{j1} - \bar{x}) + (x_{j2} - \bar{x})]^2 - s_d^2[x_{j1} - x_{j2}]^2\}}{\left(s_s^2 + s_d^2\right)},$$

where

- s_d^2 is the residual variance value from using the dependent variable DIFF,
- s_s^2 is the residual variance value from using the dependent variable SUM.

In the case of the weighted options W3 and W4, the dependent variable vector for a family is computed in the following way:

$$y = \left(\frac{W_s^{-1}}{s_s^2} + \frac{W_d^{-1}}{s_d^2} \right)^{-1} \left(\frac{W_s^{-1}}{s_s^2} y_s + \frac{W_d^{-1}}{s_d^2} y_d \right),$$

where

- s_d^2 is the residual variance value from using the dependent variable DIFF,
- s_s^2 is the residual variance value from using the dependent variable SUM,
- W_d is the weight matrix from using the dependent variable DIFF,
- W_s is the weight matrix from using the dependent variable SUM,
- y_d is the dependent variable vector from using the dependent variable DIFF,
- y_s is the dependent variable vector from using the dependent variable SUM.

18.2.3.3 Covariate terms

Let the number of covariates be C and the covariate values for i -th sib be $z_{i1}, z_{i2}, ..., z_{iC}$; $i = 1, 2, ..., N$.

The c -th covariate term for the j -th sib pair can be:

$$z_{jc} = \begin{cases} (z_{j1c} + z_{j2c}) & \text{the covariate sum (default)} \\ \mid z_{j1c} - z_{j2c} \mid & \text{the covariate absolute difference} \\ (z_{j1c} z_{j2c}) & \text{the covariate product} \end{cases}$$

where

- z_{j1c} is the raw value for the first sib in the j -th sibpair for the c -th covariate,
- z_{j2c} is the raw value for the second sib in the j -th sibpair for the c -th covariate.

Then, the c -th covariate term included in the model design matrix A for the j -th sib pair is $z_{jc} - \bar{z}_c$ where $\bar{z}_c = \frac{\sum_{j=1}^{n} z_{jc}}{n}$.

18.2.3.4 Design matrix A

A is an $n \times m$ design matrix, where m is the number of regression parameters estimated - each parameter corresponds to a particular column of **A**.

Columns of **A**:

1. The first one or two columns in the design matrix correspond to intercept(s):
   (a) In the case when only one type of siblings is allowed, either full or half, the first column is a column of 1s.
   (b) In the case when both types of siblings are allowed, both full and half, the first column contains 1 for full sib pairs and 0 for half sib pairs, and the second column contains 0 for full sib pairs and 1 for half sib pairs.
2. Following this come one or two columns for each marker locus entered in the model:
   (a) the first of each of these is a column whose elements are $\hat{\pi}_j$, centered on all n pairs;
   (b) the second of each (if present) is a column whose elements are f_2, centered. For each marker, the user can choose whether or not to include dominance (f_2) in the model; if it is not included, in a random sample a_m is an attenuated locus-specific measure of the total (additive and dominant) genetic variance.
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 (marker interactions).
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 values previously defined in the covariate terms, but centered.
5. Additional columns may be entered that are powers of previous covariates or products of previous covariates (covariate interactions). Note that including too many covariate terms may cause **A** to be singular due to linear dependencies in the data.

Thus, in the case of full sib pairs only or half sib pairs only, **A** will be of this form (all columns centered except for the first):

$$
\begin{array}{cccccccccc}
1 & \hat{\pi}_{11} & \hat{f}_{211} & \hat{\pi}_{12} & \hat{f}_{212} & \cdots & \hat{\pi}_{11}\,\hat{\pi}_{12}\cdots[z_{11}]^p\,[z_{12}]^p\cdots[z_{11}z_{12}]^p \\
1 & \hat{\pi}_{21} & \hat{f}_{221} & \hat{\pi}_{22} & \hat{f}_{222} & \cdots & \hat{\pi}_{21}\,\hat{\pi}_{22}\cdots[z_{21}]^p\,[z_{22}]^p\cdots[z_{21}z_{22}]^p \\
1 & \hat{\pi}_{31} & \hat{f}_{231} & \hat{\pi}_{32} & \hat{f}_{232} & \cdots & \hat{\pi}_{31}\,\hat{\pi}_{32}\cdots[z_{31}]^p\,[z_{32}]^p\cdots[z_{31}z_{32}]^p \\
. & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . \\
(1a) & (2a) & (2b) & (2a) & (2b) & \cdots & (3) & (4) & (5)
\end{array}
$$

In the case when both types of siblings are allowed, full and half, **A** will be of this form (all columns centered except for the first two):

1	0	$\hat{\pi}_{11}$	$\hat{f}_{211}$	$\hat{\pi}_{12}$	$\hat{f}_{212}$	…	$\hat{\pi}_{11}\,\hat{\pi}_{12}\ldots[z_{11}]^p\,[z_{12}]^p\ldots[z_{11}z_{12}]^p$		
1	0	$\hat{\pi}_{21}$	$\hat{f}_{221}$	$\hat{\pi}_{22}$	$\hat{f}_{222}$	…	$\hat{\pi}_{21}\,\hat{\pi}_{22}\ldots[z_{21}]^p\,[z_{22}]^p\ldots[z_{21}z_{22}]^p$		
1	0	$\hat{\pi}_{31}$	$\hat{f}_{231}$	$\hat{\pi}_{32}$	$\hat{f}_{232}$	…	$\hat{\pi}_{31}\,\hat{\pi}_{32}\ldots[z_{31}]^p\,[z_{32}]^p\ldots[z_{31}z_{32}]^p$		
.	.	.	.	.	.		.	.	
.	.	.	.	.	.		.	.	
.	.	.	.	.	.		.	.	
0	1	.	0	.	0		.	.	
0	1	.	0	.	0		.	.	
(1b)	(1b)	(2a)	(2b)	(2a)	(2b)	…	(3)	(4)	(5)

18.2.3.5 Weight matrix W

18.2.3.5.1 Weight matrix for DIFF, SUM, PROD and W2

The weight matrix for DIFF (W_d), SUM (W_s), PROD (W_p) and W2 (W_2) is the correlation matrix $\boldsymbol{R}$ of the corresponding residuals: $W_d = R_d$, $W_s = R_s$, $W_p = R_p$ and $W_2 = R_2$. The correlation matrix $\boldsymbol{R}$ is constructed as described in the following section and depends on the type of sib pair used.

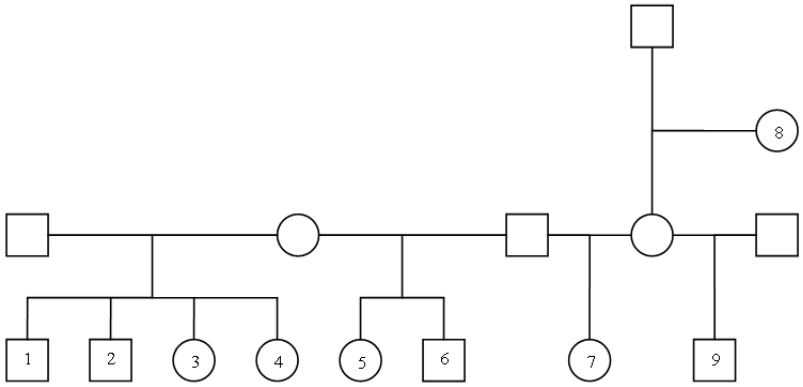
18.2.3.5.1.1 Correlation matrix R for full and half sib pairs separately

Let $r_F = (r_{F1}, r_{F0})$ be a vector of residual correlations for full sib pairs, where r_{F0} is the correlation between pairs of full sib pairs sharing 0 sibs in common and r_{F1} is the correlation between pairs of full sib pairs sharing 1 sib in common. *All correlations are between residuals of the regression model* .

Similarly, let $r_H = (r_{H1}, r_{H0})$ be a vector of correlations for related half sib pairs, where r_{H0} is the correlation between pairs of half sib pairs sharing 0 sibs in common and r_{H1} is the correlation between pairs of half sib pairs sharing 1 sib in common.

The vectors r_F and r_H are either estimated from the data for the chosen dependent variable: r_{Fs} and r_{Hs} for SUM, r_{Fd} and r_{Hd} for DIFF, and so on, with the restriction that the correlations are constrained to be greater than 0 to avoid numerical instability, or all set equal to 0 by the user. Additionally, r_{F1} is constrained to b e $\geq r_{F0}$, and $r_{H1} \geq r_{H0}$.

Consider the following family of 5 nuclear families with 7 full sib pairs and 13 half sib pairs.



The correlation matrix **R** for the chosen dependent variable between pairs of full sib pairs in the above family is:

Sib1 Sib2	**1** **2**	**1** **3**	**1** **4**	**2** **3**	**2** **4**	**3** **4**	**5** **6**
1,2	1	r_{F1}	r_{F1}	r_{F1}	r_{F1}	r_{F0}	0
1,3	r_{F1}	1	r_{F1}	r_{F1}	r_{F0}	r_{F1}	0
1,4	r_{F1}	r_{F1}	1	r_{F0}	r_{F1}	r_{F1}	0
2,3	r_{F1}	r_{F1}	r_{F0}	1	r_{F1}	r_{F1}	0
2,4	r_{F1}	r_{F0}	r_{F1}	r_{F1}	1	r_{F1}	0
3,4	r_{F0}	r_{F1}	r_{F1}	r_{F1}	r_{F1}	1	0
5,6	0	0	0	0	0	0	1

The correlation matrix **R** for the chosen dependent variable between pairs of half sib pairs in the above family is:

Sib1 Sib2	**1** **5**	**1** **6**	**2** **5**	**2** **6**	**3** **5**	**3** **6**	**4** **5**	**4** **6**	**5** **7**	**6** **7**	**7** **8**	**7** **9**	**8** **9**
1,5	1	r_{H1}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
1,6	r_{H1}	1	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
2,5	r_{H1}	r_{H0}	1	r_{H1}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
2,6	r_{H0}	r_{H1}	r_{H1}	1	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
3,5	r_{H1}	r_{H0}	r_{H1}	r_{H0}	1	r_{H1}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
3,6	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H1}	1	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
4,5	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	1	r_{H1}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
4,6	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H1}	1	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
5,7	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	1	r_{H1}	r_{H1}	r_{H1}	r_{H0}
6,7	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H1}	1	r_{H1}	r_{H1}	r_{H0}
7,8	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H1}	r_{H1}	1	r_{H1}	r_{H1}
7,9	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H1}	r_{H1}	r_{H1}	1	r_{H1}
8,9	0	0	0	0	0	0	0	0	r_{H0}	r_{H0}	r_{H1}	r_{H1}	1

18.2.3.5.1.2 Correlation matrix R for full and half sib pairs combined

To obtain a correlation matrix[1] for full and half sib pairs combined, the same correlation values as indicated in the two separate matrices above are used for the pairs of full sib pairs and the pairs of half sib pairs, except that the ratio of the residual variances $c = s_H^2/s_F^2$ is used in the diagonal for the half sib pairs, where

- s_F^2 is the residual variance from the regression using full sib pairs only and
- s_H^2 is the residual variance from the regression using half sib pairs only.

[1]The structure is referred to as " correlation" matrix even though the diagonal values are not all 1.

The correlations between full sib pairs and related half sib pairs (and related full sib pairs from other sibships within a family) are computed as follows. Analogous to the previous correlation vectors r_F and r_H, let $r = (r_1, r_0)$ be a vector of correlations between full sib pairs and related half sib pairs, where r_0 is the correlation between these sib pairs sharing 0 sibs in common when at least two of the four sibs are related as half-sibs and r_1 is the correlation between these sib pairs sharing 1 sib in common. These correlations are also constrained to be $r_1 \geq r_0 \geq 0$. Note that if none of the four sibs are related as half sibs, the correlation is set to 0.

Thus, the correlation matrix $\boldsymbol{R}$ of the chosen dependent variable for combined sib pairs in the above family is:

Sib1 Sib2	1 2	1 3	1 4	2 3	2 4	3 4	5 6	1 5	1 6	2 5	2 6	3 5	3 6	4 5	4 6	5 7	6 7	7 8	7 9	8 9
1,2	1	r_{F1}	r_{F1}	r_{F1}	r_{F1}	r_{F0}	r_0	r_1	r_1	r_1	r_1	r_0	r_0	r_0	r_0	r_0	r_0	0	0	0
1,3	r_{F1}	1	r_{F1}	r_{F1}	r_{F0}	r_{F1}	r_0	r_1	r_1	r_0	r_0	r_1	r_1	r_0	r_0	r_0	r_0	0	0	0
1,4	r_{F1}	r_{F1}	1	r_{F0}	r_{F1}	r_{F1}	r_0	r_1	r_1	r_0	r_0	r_0	r_0	r_1	r_1	r_0	r_0	0	0	0
2,3	r_{F1}	r_{F1}	r_{F0}	1	r_{F1}	r_{F1}	r_0	r_0	r_0	r_1	r_1	r_1	r_1	r_0	r_0	r_0	r_0	0	0	0
2,4	r_{F1}	r_{F0}	r_{F1}	r_{F1}	1	r_{F1}	r_0	r_0	r_0	r_1	r_1	r_0	r_0	r_1	r_1	r_0	r_0	0	0	0
3,4	r_{F0}	r_{F1}	r_{F1}	r_{F1}	r_{F1}	1	r_0	r_0	r_0	r_0	r_0	r_1	r_1	r_1	r_1	r_0	r_0	0	0	0
5,6	r_0	r_0	r_0	r_0	r_0	r_0	1	r_1	r_1	r_1	r_1	r_1	r_1	r_1	r_1	r_1	r_1	r_0	r_0	0
1,5	r_1	r_1	r_1	r_0	r_0	r_0	r_1	c	r_{H1}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
1,6	r_1	r_1	r_1	r_0	r_0	r_0	r_1	r_{H1}	c	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
2,5	r_1	r_0	r_0	r_1	r_1	r_0	r_1	r_{H1}	r_{H0}	c	r_{H1}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
2,6	r_1	r_0	r_0	r_1	r_1	r_0	r_1	r_{H0}	r_{H1}	r_{H1}	c	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
3,5	r_0	r_1	r_0	r_1	r_0	r_1	r_1	r_{H1}	r_{H0}	r_{H1}	r_{H0}	c	r_{H1}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
3,6	r_0	r_1	r_0	r_1	r_0	r_1	r_1	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H1}	c	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
4,5	r_0	r_0	r_1	r_0	r_1	r_1	r_1	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	c	r_{H1}	r_{H1}	r_{H0}	r_{H0}	r_{H0}	0
4,6	r_0	r_0	r_1	r_0	r_1	r_1	r_1	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H1}	c	r_{H0}	r_{H1}	r_{H0}	r_{H0}	0
5,7	r_0	r_0	r_0	r_0	r_0	r_0	r_1	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	c	r_{H1}	r_{H1}	r_{H1}	r_{H0}
6,7	r_0	r_0	r_0	r_0	r_0	r_0	r_1	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H0}	r_{H1}	r_{H1}	c	r_{H1}	r_{H1}	r_{H0}
7,8	0	0	0	0	0	0	r_0	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H1}	r_{H1}	c	r_{H1}	r_{H1}
7,9	0	0	0	0	0	0	r_0	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H0}	r_{H1}	r_{H1}	r_{H1}	c	r_{H1}
8,9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	r_{H0}	r_{H0}	r_{H1}	r_{H1}	c

As for r_F and r_H, all of the non-diagonal elements in **R** can be set to 0 by the user.

18.2.3.5.2 Weight matrix for W3

The weight matrix W_3 for option W3 is the estimated residual variance matrix Σ of the weighted dependent vector y , so that $W_3 = \Sigma = \left(\frac{1}{s_d^2}R_d^{-1} + \frac{1}{s_s^2}R_s^{-1}\right)^{-1}$, where

- s_d^2 is the residual variance value from using the dependent variable DIFF,
- s_s^2 is the residual variance value from using the dependent variable SUM,
- R_d is the correlation matrix from using the dependent variable DIFF,
- R_s is the correlation matrix from using the dependent variable SUM.

18.2.3.5.3 Weight matrix for W4

The weight matrix W4 for option W4 is further adjusted for possible non-independence of the squared trait sums and differences.

Let $r_{Fsd} = (r_{Fds2}, r_{Fds1}, r_{Fds0})$ be a vector of correlations between the residuals of the sums and differences for full sib pairs with 2, 1, 0 sibs in common. Similarly, let $r_{Hsd} = (r_{Hds2}, r_{Hds1}, r_{Hds0})$ be a vector of correlations between the residuals of the sums and differences for half sib pairs with 2, 1, 0 sib in common. These correlations are estimated from the data with the restrictions $r_{Fds2} \geq r_{Fds1} \geq r_{Fds0} \geq 0$ and $r_{Hds2} \geq r_{Hds1} \geq r_{Hds0} \geq 0$.

The correlation matrix between y_s and y_d, R_{sd}, is constructed in the in the same way as the correlation matrices above except that the diagonal elements are r_{Fsd2} or r_{Hsd2} instead of being all 1. For example, R_{sd} for the full sib pairs and half sib pairs in above family separately will look like the following:

Sib1 Sib2	**1** **2**	**1** **3**	**1** **4**	**2** **3**	**2** **4**	**3** **4**	**5** **6**
1,2	r_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	0
1,3	r_{Fsd1}	r_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	0
1,4	r_{Fsd1}	r_{Fsd1}	r_{Fsd2}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	0
2,3	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd2}	r_{Fsd1}	r_{Fsd1}	0
2,4	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd2}	r_{Fsd1}	0
3,4	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd2}	0
5,6	0	0	0	0	0	0	r_{Fsd2}

Sib1 Sib2	**1** **5**	**1** **6**	**2** **5**	**2** **6**	**3** **5**	**3** **6**	**4** **5**	**4** **6**	**5** **7**	**6** **7**	**7** **8**	**7** **9**	**8** **9**
1,5	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	0
1,6	r_{Hsd1}	r_{Hsd2}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	0
2,5	r_{Hsd1}	r_{Hsd0}	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	0
2,6	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	0
3,5	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	0
3,6	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	0
4,5	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	0
4,6	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd0}	0
5,7	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}	r_{Hsd1}	r_{Hsd0}
6,7	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}	r_{Hsd0}
7,8	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}	r_{Hsd1}	r_{Hsd1}
7,9	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}	r_{Hsd1}
8,9	0	0	0	0	0	0	0	0	r_{Hsd0}	r_{Hsd0}	r_{Hsd1}	r_{Hsd1}	r_{Hsd2}

To obtain a correlation matrix for full and half sib pairs combined, the same correlation values as indicated in the above two matrices are again used for the full sib pairs and the half sib pairs, except that the diagonal elements for the half sib pairs are multiplied by the ratio of the residual variance $c = s_H^2/s_F^2$.

Correlations between sums and differences of full sib pairs and related half sib pairs (and related full sib pairs from sibships within a family) are also computed the same as before. Let $r_{sd} = (r_{sd1}, r_{sd0})$ be a vector of correlations between the residuals of the sums and differences for full sib pairs and

related half sib pairs, where r_{sd0} is the correlation between these sib pairs sharing 0 sibs in common when at least two of the four sibs are related as half-sibs and r_{sd1} is the correlation between these sib pairs sharing 1 sib in common. These correlations are also constrained to be $r_{sd1} \geq r_{sd0} \geq 0$. Note that if none of the four sibs are related as half sibs, the correlation is set to 0.

Thus, the correlation matrix R_{sd} for combined sib pairs in the above family is:

Sib1 Sib2	1 2	1 3	1 4	2 3	2 4	3 4	5 6	1 5	1 6	2 5	2 6	3 5	3 6	4 5	4 6	5 7	6 7	7 8	7 9	8 9
1,2	r_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	0	0	0
1,3	r_{Fsd1}	r_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	0	0	0
1,4	r_{Fsd1}	r_{Fsd1}	r_{Fsd2}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	0	0	0
2,3	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	0	0	0
2,4	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd2}	r_{Fsd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	0	0	0
3,4	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd2}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	0	0	0
5,6	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{Fsd2}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}		0
1,5	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	0
1,6	r_{sd1}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	0
2,5	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd1}	r_{Fsd1}	r_{Fsd0}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	0
2,6	r_{sd1}	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd0}	r_{sd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	0
3,5	r_{sd0}	r_{sd1}	r_{sd0}	r_{sd1}	r_{sd0}	r_{sd1}	r_{sd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	0
3,6	r_{sd0}	r_{sd1}	r_{sd0}	r_{sd1}	r_{sd0}	r_{sd1}	r_{sd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	0
4,5	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	0
4,6	r_{sd0}	r_{sd0}	r_{sd1}	r_{sd0}	r_{sd1}	r_{sd1}	r_{sd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd0}	0
5,7	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}
6,7	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd0}	r_{sd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}	r_{Fsd0}
7,8	0	0	0	0	0	0	r_{sd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd1}	r_{Fsd1}
7,9	0	0	0	0	0	0	r_{sd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}	r_{Fsd1}
8,9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	r_{Fsd0}	r_{Fsd0}	r_{Fsd1}	r_{Fsd1}	CT_{Fsd2}

Then, the weight matrix W_4 for the option W4 method is the variance matrix Σ, adjusted as follows:

$$W_4 = \Sigma + \frac{1}{s_s s_d} \Sigma \left(R_s^{-1} R_{sd} R_d^{-1} + R_d^{-1} R_{sd} R_s^{-1} \right) \Sigma,$$

where

- s_d^2 is the residual variance value from using the dependent variable DIFF,
- s_s^2 is the residual variance value from using the dependent variable SUM,
- R_d is the correlation matrix from using the dependent variable DIFF,
- R_s is the correlation matrix from using the dependent variable SUM.

18.2.3.6 Generalized estimating equations (GEE)

An iterative method using the generalized estimating equations (GEE) of Liang and Ziegler(1986) is used in each model to allow for the non-independence of sibling pairs. Initially, all correlations are set to 0 to obtain the residuals, then the correlations of residuals from the previous iteration are used

to update the weight matrix W, and new values of the parameter estimates b_i and s_i^2 are generated, $i = 1, 2, ..., m$.

Let $\tilde{r}_0$, $\tilde{r}_1$ be residual correlations from the previous iteration and be these calculated in the current iteration. The iteration is stopped when the value $\delta = \left|\frac{\tilde{r}_0 - r_0}{\tilde{r}_0}\right| + \left|\frac{\tilde{r}_0 - r_0}{\tilde{r}_0}\right| \leq 0.2$ or the maximum number of iterations is achieved.

18.2.3.7 Significance tests

To assess significance, we use the t test statistics $\frac{b_i}{v_i}$, for $i = 2, 3, ..., m$, where b_i is the i-th element of the parameter estimates $b = (A^T W^{-1} A)^{-1} A^T W^{-1} y$ and v_i^2 is the variance estimate, which is the product of the i-th diagonal element of $(A^T W^{-1} A)^{-1}$ and $\frac{y^T W^{-1}(y - Ab)}{n-m}$.

For each test statistic we calculate a p-value which is either

$$p_t = \mathrm{P}\left(t_{n-m} \geq \frac{b_i}{v_i}\right) \quad \text{(one-sided test)}$$

or

$$p_t = 2\mathrm{P}\left(t_{n-m} \geq \frac{|b_i|}{v_i}\right) \quad \text{(two-sided test),}$$

where t_{n-m} is a random variable that is distributed as t with n-m d.f.

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 or $\hat{f}_2$s) use one-sided tests. A two-sided test is used for all remaining columns that contain any covariate terms.

Furthermore, 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

$$(A'W^{-1}A)^{-1}\left[A'W^{-1}(y - Ab)\right]\left[A'W^{-1}(y - Ab)\right]'(A'W^{-1}A)^{-1}$$

and use the same t test statistics $\frac{b_i}{v_i}$ as above, except v_i^2 is now the i-th diagonal element of the *sandwich* variance matrix estimate. These variance estimates can be extremely conservative and caution should be exercised when using this option. These need only be used when the data contain full sibs.

18.2.3.8 Empirical estimates of significance (full sibs only)

We can also estimate an empirical mid 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 have have p-values less than, plus half the proportion that have p-values equal to the p-value calculated from the original observations. We choose N, the number of replicates, such that the estimated empirical mid p-value, $\hat{p}$, is within a proportion w (the width parameter) of its true p-value, p , with predetermined confidence probability (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 the pre-specified proportion w of its true value with confidence coefficient γ. For example, if we wish to estimate an empirical p-value that is 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.

18.2.3.9 Regression model for X-linked markers

Note that this is implemented for full-sibs only.

In the case of X-linkage, there are three types of sib pairs, and hence three different πs (Wiener *et al* . 2003):

$$\left.\begin{array}{l} \hat{\pi}_{BB} = \hat{f}_0 \text{ for brother - brother pairs} \\ \hat{\pi}_{BS} = \hat{f}_0 \text{ for brother - sister pairs} \\ \hat{\pi}_{SS} = \hat{f}_1 \text{ for sister - sister pairs} \end{array}\right\} \text{ all are in the model.}$$

At the X-location, the three coefficients correspond to three different variance components, just as in the autosomal case the coefficient of $\hat{\pi}$ is σ^2_g.

Let the three coefficients be β_{BB}, β_{BS}, and β_{SS}. Each is tested by a t-test (one-sided), analogous to the autosomal case. If all three are > 0, we do a 3-d.f. F-test. If only two are > 0, we do a 2-d.f. F-test.

For the 3-d.f. F-test, we do two regressions, one with the 3 $\hat{\pi}$s in the model and one without. Call the corresponding residual sums of squares SSE_C (complete model) and SSE_R (reduced model), respectively. Then the F statistic will be

$$\frac{SSE_R - SSE_C}{3MSE_C} \text{ with 3 and } N\text{-}p \text{ d.f.}$$

where

- N is the effective number of independent sib pairs (= total number of sibs - number of sibships) and
- p is the total number of parameters for full model (= number of columns in the design matrix that includes the 3 $\hat{\pi}$s).

For the 2-d.f. F-test, we perform an analogous test for the two coefficients that are positive, but treating the parameter that has a negative coefficient as a nuisance parameter, i.e. included in both models to obtain SSE_C and SSE_R.

Permutations to get empirical p-values are done by permuting within each of the 3 kinds of sib pairs, as well as within and across sibships as done for the autosomal cases, and calculating the above statistics for each permutation replicate.

18.3 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.

18.3.1 Running `sibpal`

A typical run of the SIBPAL program may use flags to identify the file types like the following:

```
>sibpal -p data.par -d data.ped -i ch1.ibd
```

or, rely on a set file order like the following:

```
>sibpal data.par data.ped ch1.ibd
```

where `data.par` is the name of the parameter file, `data.ped` is the name of the pedigree data file, and `ch1.ibd` is the name of the IBD sharing file.

18.3.2 The `sibpal` Block

A `sibpal` block in the parameter file sets the options on how to perform an analysis using SIBPAL. The following table shows the syntax for a `sibpal` parameter which starts the `sibpal` block.

parameter [, attribute]	Explanation
sibpal	Starts a SIBPAL 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: traits Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `sibpal` block.

parameter [, attribute]	Explanation
`mean_test`	Starts a sub-block for specifying tests of mean IBD sharing. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`trait_regression`	Starts a sub-block for specifying a Haseman-Elston type regression of traits on one or more markers, covariates, and interactions. Value Range: N/A Default Value: N/A Required: No 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: N/A Required: No Applicable Notes: 3
`, single` `, single_marker`	Selects single regression on one marker at a time. Value Range: N/A Default Value: N/A Required: No Applicable Notes: None
`, multiple` `, multiple_marker`	Selects multiple regression on all chosen markers at once. Value Range: N/A Default Value: N/A 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: No Applicable Notes: 1

Notes

1. If a `trait_regression` statement does not have either the `single` or `multiple` attribute, 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. Single regression is performed by default.

3. This performs the regression analysis without any marker in the regression model.

18.3.2.1 The `mean_test` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 or created by means of a function block.
	Default Value: None
	Required: No
	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
w1	Specifies a value for w_1 in the equation $\hat{\pi}_j = \hat{f}_{2j} + w_1 \hat{f}_{1j}$ Value Range: [0, 0.5] Default Value: 0.5 Required: No Applicable Notes: 4

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 a binary 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.

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.

4. The value of `w1` cannot be specified (i.e, it keeps its default value 0.5) if half-sibs are being analyzed. Specifying this value to be 0 when the sample analyzed consists of only full sibs leads to the "proportion test" when the mean test is performed.

18.3.2.2 The `trait_regression` Sub-Block

The following table lists the parameters and attributes that may occur in a `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 or created by means of a function block.
	Default Value: None
	Required: No
	Applicable Notes: 1
`, mean`	Specifies option to fix the trait mean to a value other than the sample mean.
	Value Range: {blup, sibship, sample, $\mu \in (-\infty, +\infty)$}
	Default Value: blup
	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: N/A
	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 or created by means of a function block.
	Default Value	None
	Required	No
	Applicable Notes	4
`, prod`	Include as covariate the mean-corrected product.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None
`, sum`	Include as covariate the mean-corrected sum.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None
`, diff`	Include as covariate the mean-corrected absolutedifference.	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None
`, all`	Include all covariate terms (sum, difference and product).	
	Value Range	N/A
	Default Value	N/A
	Required	No
	Applicable Notes	None
`, power`	Raise the covariate terms to specified power.	
	Value Range	(- ∞, ∞)
	Default Value	1.0
	Required	No
	Applicable Notes	4
`interaction`	Starts a `interaction` 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	N/A
	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
`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 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
`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. Value Range: {true, false} Default Value: false 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 the width interval 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 first n rows of the design matrix A.	
	Value Range	Character string representing the name of a marker or location listed in the IBD sharing file.
	Default Value	N/A
	Required	N/a
	Applicable Notes	9
`, rows`	Specifies the number of rows to print.	
	Value Range	{0, 1, 2, 3, ..., n }, where n is the sample size.
	Default Value	10
	Required	No
	Applicable Notes	9
`print_correlation_matrix`	Specifies option to print the sibship size-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
`print_QLS`	Specifies option to print the linkage score in Wang and Elston (2006).	
	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	11
`use_pairs`	Specifies option to analyze full-sibs only, half-sibs only, or both.	
	Value Range	{full, half, both}
	Default Value	full
	Required	No
	Applicable Notes	None

w1	Specifies a value for w_1 in the equation $\hat{\pi}_j = \hat{f}_{2j} + w_1 \hat{f}_{1j}$	
	Value Range	[0, 0.5]
	Default Value	0.5
	Required	No
	Applicable Notes	12

Notes

1. The value of a `trait` parameter should be set to the name of a trait or covariate field read from the data file or created by means of a function block. 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. Other options, sample mean of individuals used in the regression, using sibship-specific mean or using the best linear unbiased predictor (BLUP) mean, can be specified by setting `mean` = **sample**, `mean` = **sibship** or `mean` = **blup**. If not specified, the best linear unbiased predictor (BLUP) mean is used by default.

2. The value of a `marker` parameter should be set to the name of a marker for which IBD sharing information was generated by GENIBD and stored in the IBD sharing file. If no valid marker parameters are listed then all markers in the IBD sharing file 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 or covariate` field read from the data file or created by means of a function block. If no valid `covariate` parameters are listed, then by default no covariates are included. The covariate, with its attribute, will be raised to the specified power *before* mean correction is applied.

5. The `interaction` parameter should contain a sub-block of `marker` and `covariate` parameters that specify a multiplicative interaction term in the regression model. Note that interaction terms are allowed only when both corresponding main effects are included in the model, and marker by marker interaction is only allowed in multiple_marker trait regression. 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	-½ squared trait difference ($-\frac{1}{2}$ ×traditional Haseman-Elston).
sum	½ 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)

A general recommendation might be to try using the the last regression method in this list (**W4**) first, and then work upwards until no signs of numerical instability are seen. The default is set at **prod** in case the user does not know how to recognize such instability, which is very unlikely to occur for prod. Unlike all the other options, the results from the simplest option, **diff**, are not affected by the sample mean.

7. 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.

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 `trait_regression` 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. 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 `trait_regression` 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 which location to print.

11. This option is not relevant if the `zero_marker` attribute is specified for the `trait_regression` parameter. If the `multiple_marker` attribute is specified, 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 which location to print. The QLS output lists, for each sib-pair:

- the pedigree ID
- a number indicating the sibship within the pedigree
- the IDs of the two sibs
- a number indicating the sibship within the pedigree
- (type of affected status of sib-pair, if the trait is binary)
- the dependent trait in the HE regression, standardized
- the quamtitative linkage score (QLS) for each sib-pair, averaged over the sib-pairs in the sibships.

The linkage score in Wang and Elston (2006) is then obtained by adding together all the scores in a pedigree. However, this will be proportional to the score as defined by Wang and Elston (2006), rather than the score they defined. Thus it can be used for the same purpose, but the actual value will not be the same.

12. The value of `w1` cannot be specified (i.e, it keeps its default value 0.5) if half-sibs are being analyzed, i.e., if the value of `use_pairs` parameter is set to either **half** or **both**.

18.4 Program Output

SIBPAL produces several output files that contain results and diagnostic information:

File Name	File Type	Description
sibpal.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
means.out	Mean analysis output file	Contains the results of each test of mean allele sharing IBD.
traits.out	Trait Regression analysis summary output file	Contains the summary results of each linkage test.
traits.det	Trait Regression analysis detailed output file	Contains the detailed results of each linkage test.

18.4.1 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.

Example:

```
=========================================================================
  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.


  w1: 0.50

=========================================================================
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
-------------------------------------------------------------------------
.
.
.
```

18.4.2 Trait Regression Analysis Summary 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.

Example:

```
================================================================================
  Haseman-Elston Regression Analysis of Full Sibs
  - single_marker regression
================================================================================

    Binary trait : affection, affected = 'A' , unaffected = 'U'

    Dependent variate : Mean-corrected squared trait sum

    Other options used :
      Identity weights = no
      Robust variance  = no
      Use sibship mean = no
      Use BLUP mean    = no

    Legend :
      Note - kurtosis = coefficient of kurtosis - 3
      *   - significance  .05 level;
      **  - significance  .01 level;
      *** - significance .001 level;
      #   - Negative intercept set to 0;

================================================================================
Test  Independent                                                    Nominal
No.   variable          Pairs   Parameter    Estimate Std Error   P-value
----- ----------------- ----- ----------- ----------- --------- ---------
1     D5G1                 28 (A+D)GenVar     0.0108     0.0619 0.4311918
2     D5G2                 28 (A+D)GenVar     0.0109     0.0523 0.4186353
3     D5G3                 28 (A+D)GenVar    -0.0091     0.0561 0.5638613
4     D5G4                 28 (A+D)GenVar    -0.0139     0.0638 0.5852159
5     D5G5                 28 (A+D)GenVar    -0.0397     0.0474 0.7952521
6     D5G6                 28 (A+D)GenVar    -0.0064     0.0536 0.5474505
7     D5G7                 28 (A+D)GenVar    -0.0568     0.0589 0.8284099
8     D5G8                 28 (A+D)GenVar     0.0000     0.0643 0.5000000
9     D5G9                 28 (A+D)GenVar     0.0436 #   0.0491 0.1909162
10    D5G10                28 (A+D)GenVar    -0.0000     0.0580 0.5000000
11    D5G11                28 (A+D)GenVar    -0.0036     0.0564 0.5251796
12    D5G12                28 (A+D)GenVar    -0.0074     0.0556 0.5521124
.
.
.
24    D5G24                28 (A+D)GenVar    -0.0120     0.0522 0.5899965
25    D5G25                28 (A+D)GenVar    -0.0803     0.0472 0.9497509
================================================================================
```

18.4.3 Trait Regression Analysis Detailed Output File

One Trait Regression analysis output file, named "`traits.det`", is generated per run of SIBPAL. It contains the detailed results of all Trait Regression linkage tests.

Example:

```
================================================================================
Test 1
================================================================================
-----
Model
-----

affection ~ Intercept + D5G1(A+D) + e

------
Sample
------

Number of all sibs            =   27
Number of full sib pairs      =   28

-----
Trait
-----

Sample mean                   =    0.0370
Sample variance               =    0.0370
Sample skewness               =    4.9029
Sample kurtosis               =   22.0385
Pairwise full sib correlation =   -0.0182
Intra sibship correlation     =    0.0610

-----------------
Dependent variate
-----------------

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

----------
Regression
----------
                   Estimate Std Error   P-value
----------------- --------- --------- ---------
Intercept            0.0130
D5G1                 0.0108    0.0619 0.4311918
Total variance       0.0062
Residual variance    0.0067
Residual skewness    4.9942
Residual kurtosis   22.9761
================================================================================
Test 2
================================================================================
.
.
.
```

Chapter 19

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 affecting a binary trait. It is a valid test for association in the presence of linkage only if there is only one offspring in each family.

19.1 Limitations

The TDTEX program makes the following assumptions:

1. Each marker transmits alleles in a Mendialian fashion.
2. Only autosomal loci are considered.
3. Only binary traits 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 more than around 300 observations, or with more than about 8 alleles. In such cases, the asymptotic or Monte Carlo test statistics are recommended instead.

19.2 Theory

TDTEX consists of four main components:

1. A scoring algorithm to identify which alleles or genotypes are transmitted to affected offspring.

2. Production of 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.

19.2.1 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 approach 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 Figure19.1).

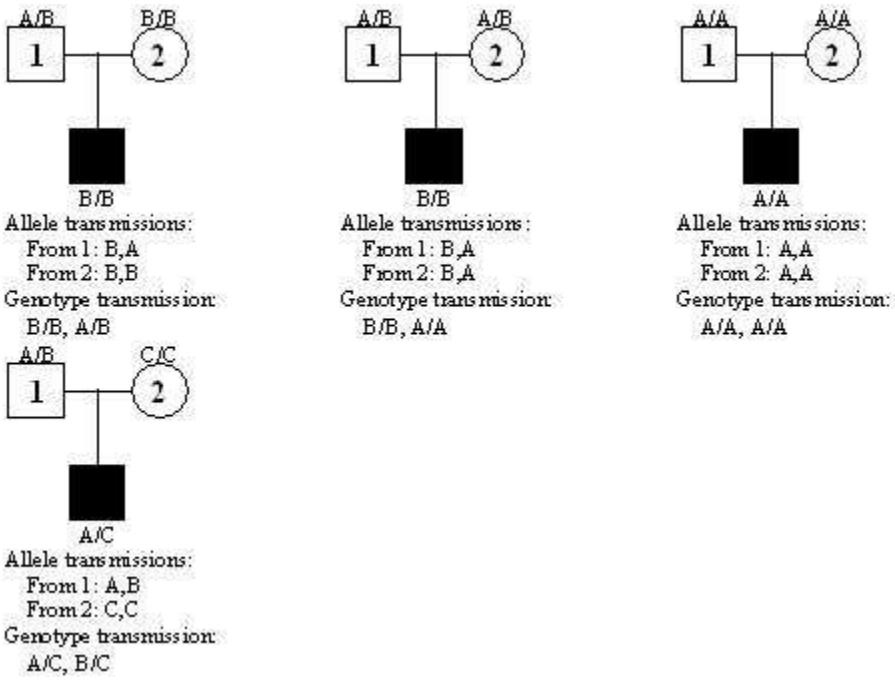


Figure 19.1: Allele and Genotype Transmission Examples

19.2.2 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 19.1 presents the transmissions that are scored and 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.

In this situation, it is known that either an A allele is transmitted and a B allele is not transmitted, or vice versa (see Sham et. al. 2000). In the case of sex-specific counting, however, this type of transmission is not considered informative (since it is not known from which parent an individual received a particular allele).

19.2.3 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; Cleves et al., 1997). This

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	*A,B	*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 19.1: Transmissions scored for all possible distinct configurations of parents and offspring.

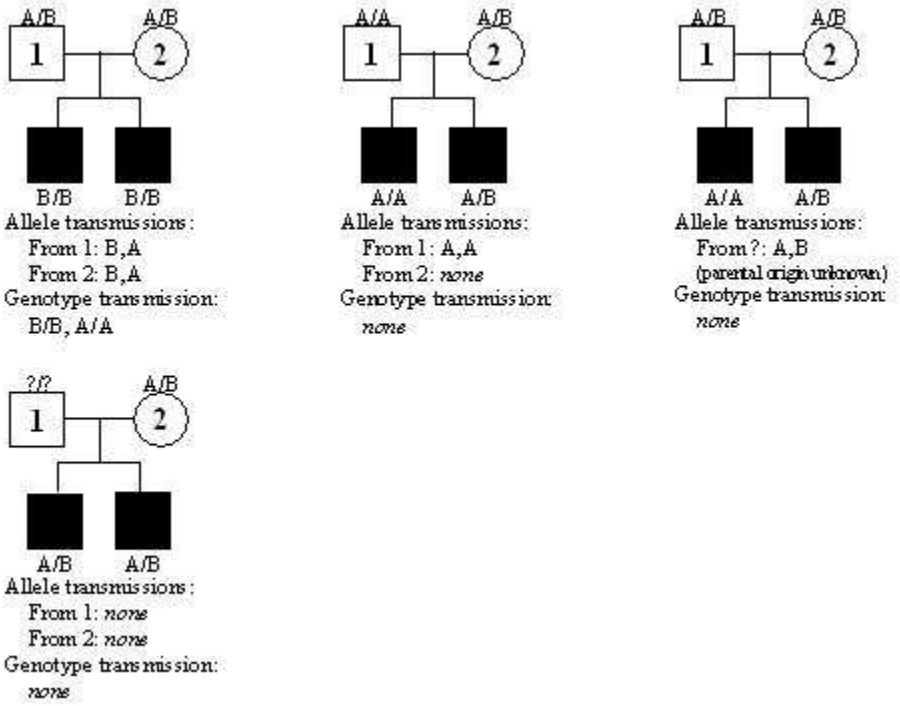


Figure 19.2: Allele and Genotype Transmission for Sibling Pairs

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 Figure19.2).

Table 19.2 presents the transmissions that are scored and 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 19.2: Transmissions scored for all possible distinct configurations of parents and two offspring receiving the same transmission

* - parental origin is unknown, we cannot know whether the same allele is transmitted from a given parent to both children.

19.2.4 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 (Table19.3). 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 19.3: The structure of a transmission table

19.2.5 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, 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 datasets 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.

19.2.6 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 transmissions 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.

19.2.6.1 Asymptotic Tests

Under the hypothesis of complete symmetry, the McNemar 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 McNemar 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 often, provided the table margins are independent of each other (Spielman and Ewens, 1996).

19.2.6.2 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, that 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 $n_{22} = 4$.

19.2.6.3 Approximation by Permutation Sampling

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 sampling 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.

19.3 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.

19.3.1 Running `tdtex`

A typical run of the TDTEX program may use flags to identify the file types like the following:

```
>tdtex -p data.par -d data.ped
```

or, rely on a set file order like the following:

```
>tdtex data.par data.ped
```

where `data.par` is the name of the parameter file and `data.ped` is the name of the pedigree data file.

19.3.2 The `tdtex` Block

A `tdtex` block in the parameter file sets the options on how to perform an analysis using TDTEX.

The following table shows the syntax for a `tdtex` parameter which starts the `tdtex` block.

parameter [, attribute]	Explanation
`tdtex`	Starts a TDTEX 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: tdtex Required: No Applicable Notes: None

The following table lists the parameters and attributes that may occur in a `tdtex` block.

parameter [, attribute]	Explanation
`trait`	Specifies a trait denoting affection status for all offspring and sibling pairs. Value Range: Character string representing the name of a valid trait or covariate listed in the data file. Default Value: None Required: Yes Applicable Notes: None
`marker`	Specifies a marker for which transmissions are to be scored. Value Range: Character string representing the name of a valid marker listed in the pedigree data file. Default Value: None Required: No Applicable Notes: 1
`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 binary trait or covariate listed in the 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: 2, 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: 3, 4
`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: 5
`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: 6
`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: 6
`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: 6

1. 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.

2. 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**.

3. Regardless of the values of `max_children` and `max_sib_pairs`, pedigrees must have at least one typed parent.

4. 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**.

5. If false, sex of parent is ignored.

6. If set to false, the computation time may be excessive. See 19.2.6.3.

19.4 Program Output

TDTEX produces several output files that contain results and diagnostic information:

File Name	File Type	Description
tdtex.inf	Information output file	Contains informational diagnostic messages, warnings and program errors. **EXAMINE THIS FILE FIRST BEFORE OPENING ANY OTHER ANALYSIS OUTPUT FILES.**
tdtex.out	TDTEX analysis output file	Contains the results of each TDT analysis.

19.4.1 TDTEX Analysis Output File

One analysis output file, named "tdtex.out", is generated per run of TDTEX. It contains the results of all tests.

Example:

```
====================
     Analysis #1
====================


  Transmission type:        Allele
  Marker:                   D7S821
  Trait:                    SPCHONLY
  Parental trait:           Not specified
  Parental sex(es) scored:  Paternal/maternal
  Max. children/family:     Unlimited
  Max. sib pairs/family:    Unlimited


-----------------------------------------------------
  T/NT  238  242  246  250  254  258  262  266  270
-----------------------------------------------------
  238     0    0    0    0    0    0    2    2    1
  242     0    0    0    0    0    0    1    0    0
  246     0    0    0    0    1    1    0    0    0
  250     1    0    1    0    1    2    3    1    0
  254     0    0    1    1    2    1    2    2    1
  258     0    1    2    3    5   11    1    1    1
  262     5    0    3    2    1    6    2    0    0
  266     1    1    1    1    1    1    3    0    0
  270     0    0    0    0    0    0    1    1    0


-------------------------------------------------------------------------
  Group               Informative count  Total count  % Informative
-------------------------------------------------------------------------
  Pedigrees                          41           89   0.460674
  Families                           41           91   0.450549
  Affected children                  32           32   1.000000
  Affected sib pairs                 14           15   0.933333
  Sample size                        67           82   0.817073


-------------------------------------------------------------
  Exact test statistic              P-value    Std. err.
-------------------------------------------------------------
  Exact McNemar test                (skipped)
  Monte Carlo McNemar test          (skipped)
  Monte Carlo Marginal Homogeneity  (skipped)


-----------------------------------------------------
  Asymptotic test statistic          P-value
-----------------------------------------------------
  McNemar test                        0.44904203
  Continuity corrected McNemar test   0.99917563
  Marginal homogeneity test           0.72933127
```

Chapter 20

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. A novel use of the program DESPAIR has been presented by Ochs-Balcom et al (2010)

Note: The DESPAIR program can only be run on the S.A.G.E. web site at:

`http://darwin.cwru.edu/despair/`.

20.1 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.

20.2 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 recombination 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:

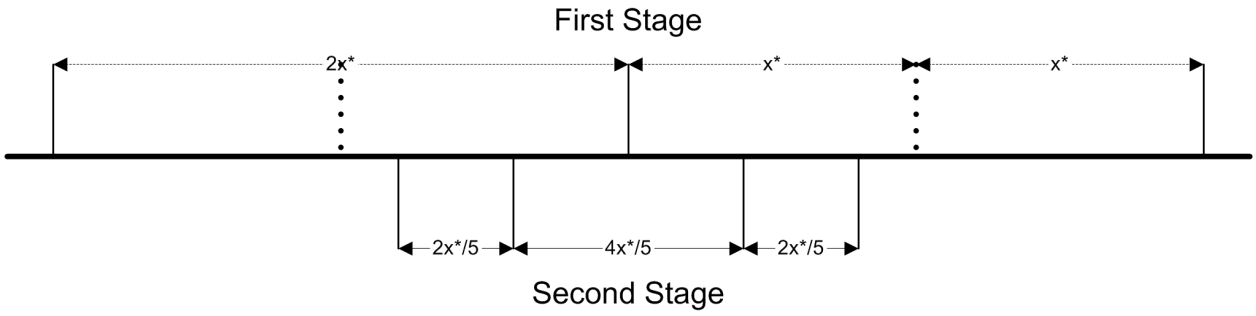
$$\lambda^- = \frac{Pr(\text{first degree relative of affected person is unaffected})}{Pr(\text{random member of population is unaffected})}.$$

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 types of 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 $2k$ 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)



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 20.1: Stage-1 and Stage-2 Marker Placement

to span in an optimal manner the interval of interest suggested by the significant first-stage marker (see Figure20.1).

Assume that we want to design a study 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

$$2n\{R+m+2k[\alpha^* m+(1-\beta)d]\}. \tag{20.1}$$

.

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

$$C = 2n\{R+m+2k[\alpha^*(m-\sum_{i=1}^{d} l_i)+\sum_{i=1}^{d}\sum_{j=1}^{l_i}(1-\beta_{ij})]\}, \tag{20.2}$$

where l_i is the number of first stage markers linked to disease locus i, and $1-\beta_{ij}$ is the probability that $2k$ 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 (20.2), 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 (20.2) approaches cost function (20.1), which was the function used in the original version of DESPAIR.

Suppose the following are specified: $\alpha, \beta, \lambda, R, d, g, M$, 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). Given all this, 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 $M < m < 1000$M (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 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.

20.3 Running the Program

DESPAIR can be run by clicking on the DESPAIR GUI link on the S.A.G.E. website (found in the drop-down menu under Programs & Links)

http://darwin.cwru.edu/despair/

and inputting one or more sets of parameters for which the sample size (numbers of relative 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 (a d b in the output) 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 (m, p in the output).	
	Value Range	**M** (mean statistic) **P** (proportion statistic)
	Default Value	M
	Required	Yes
	Applicable Notes	1
`offspr_recurrence_risk`	Specifies λ_o, the locus-specific relative recurrence risk ratio of disease for an offspring of an affected person.	
	Value Range	$[1, +\infty)$ Constraint: $1 \leq \lambda_o \leq \lambda_s$
	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 an affected person.	
	Value Range	$(0, 1]$ Constraint: $0 < \lambda_s^- \leq \lambda_o^- \leq 1$, and $\frac{1}{3} < \lambda_o^-$
	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 an affected person.	
	Value Range	$[\lambda_o, +\infty)$ Constraint: $\lambda_o \leq \lambda_s$
	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 an affected person.	
	Value Range	$(0, \lambda_o^-)$ Constraint: $\lambda_s^- \leq \lambda_o^- \leq 1$, and $(1+\lambda_o^-)/4 \leq \lambda_s^-$
	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	(0.1, $+\infty$)
	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) of the markers (assumed the same for all markers).	
	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 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: $\{ M+1, M+2 ..., 1000\text{M}\}$ 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 parameter 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.

20.4 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

20.4.1 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 21

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

Blackwelder WC, Elston RC. (1985) *A comparison of sib-pair linkage tests for disease susceptibility loci*. Genetic Epidemiology; 2:85-97

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*. American Journal of Medical Genetics; 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*. Journal of the Royal Statistical Society [B]; 26:211-252

Cannings C, Thompson EA, Skolnick MH. (1978) *Probability functions on complex pedigrees*. Advanced Applied Probability; 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.* Annals of Human Genetics; 59:323-336

Demenais FM, Murigande C, Bonney GE. (1990) *Search for faster methods of fitting the regressive models to quantitative traits*. Genetic Epidemiology; 7:319-334

Demenais FM, Elston RC. (1981) *A General Transmission Probability Model for Pedigree Data.* American Journal of Human Genetics; 33:300-306

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.* Genetic Epidemiology; 13:535-558.

Elston RC, Stewart J. (1971) *A general model for the genetic analysis of pedigree data*. Human Heredity; 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 Heredity; 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*. Theort of Applied Genetics; 88:573-580

George VT, Elston RC. (1987) *Testing the association between polymorphic markers and quantitative traits in pedigrees*. Genetic Epidemiology; 4:193-201

George VT, Elston RC (1988) *Generalized modulus power transformations*. Commun Statististics – Theory Methodology; 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. (2006) *Theoretical Aspects of Pedigree Analysis.* Tel-Aviv, Israel. Ramot Publishing House.

Go RCP, Elston RC, Kaplan EB. (1978) *Efficiency, robustness of pedigree segregation analysis*. American Journal of Human Genetics; 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

Guo X, Elston RC. (1999) *Linkage information content of polymorphic genetic markers.* Human Heredity; 49:112-118

Guo X, Elston RC. (2000) *Two-stage global search designs for linkage analysis II: Including discordant relative pairs in the study.* Genetic Epidemiology; 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.* Human Heredity; 53:45-48.

Hanson R, Knowler W. (1998) *Analytic Strategies to detect linkage to a common disorder with genetically determined age of onset.* Genetic Epidemiology; 15:299-315

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*. American Journal of Medical Genetics; 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

Mathew G, Song Y, Elston RC. (2011) *Interval estimation of familial correlations from pedigrees*. Statistical Applications in Genetics and Molecular Biology; 10(1): Article 11

McCulloch CE, Neuhaus, JM. (2011) Misspecifying the Shape of a Random Effects Distribution: Why Getting It Wrong May Not Matter. Statistical Science; 26(3): 388–402

Ochs-Balcom HM, Guo X, Yonebayashi T, Wiesner G, Elston RC. (2010) *Program Update and Novel Use of the DESPAIR Program to Design a Genome-Wide Linkage Study Using Relative Pairs*. Human Heredity; 69(1): 45-51

Olson JM, Wijsman EM. (1993) *Linkage between quantitative trait and marker loci: methods using all relative pairs*. Genetic Epidemiolgy; 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 Genetics; 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. II. The power of affected relative pairs.* American Journal of Human Genetics; 46:229-241.

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

Schnell AH, Sun X, Igo RP, Elston RC. (2012) *Some capabilities for model-based and model-free linkage analysis using the program package S.A.G.E. (Statistical Analysis for Genetic Epidemiology).* Human Hereidty; in press

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*. Journal of the American Statatistica Association; 82:605-610

Sinha M, Song Y, Elston RC, Olson JM, Goddard KAB. (2006) *Prediction of empirical p values from asymptotic p values for conditional logistic affected relative pair linkage analysis*. Human Hereidty; 61(1): 45-54

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 of 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 of Human Genetics; 52:506-516

Wang N, Akey JM, Zhang K, Chakraboryy R, Jin L. (2002) *Distribution of Recombination Crossovers and the Origin of Haplotype Blocks: The Interplay of Population History, Recombination and Mutation*. American Journal of Human Genetics; 71:1227-1234

Wang S, Kidd KK, Zhao H. (2003) *On the Use of DNA Pooling to Estimate Haplotype Frequencies*. Genetic Epidemiology; 24:74-82

Wang T, Elston RC. (2005) *Two-level Haseman-Elston regression for general pedigree data analysis*. Genetic Epidemiology; 29:12-22

Wang T, Elston RC. (2006) *A quantitative linkage score for an association study following a linkage analysis*. BMC Genetics; 7:5

Wang T, Elston RC. (2007) *Regression-based multivariate linkage analysis with an application to blood pressure and body mass index*. Annals of Human Genetics; 71:96-106

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

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.* Genetic Epidemiology; 20:397-400.

Zhu X, Olson JM, Schnell AH, Elston RC. (1997) *Genetic Analysis Workshop 10: Model-free age-of-onset methods applied to the linkage of bipolar disorder.* Genetic Epidemiology; 14:711-716.