J. Ott 20 May 1994
Columbia University, Unit 58 Tel. (212) 960-2507
722 West 168 Street Fax (212) 568-2750 e-mail:
New York, NY 10032 ott@nyspi.Bitnet or jurg.ott@columbia.edu

LINKAGE programs for Borland Pascal 7.0

Version 5.1 for general pedigrees, 5.2 for 3-generation pedigrees

A. INSTALLATION

To work with the LINKAGE programs, it is best to reserve a specific directory on your hard disk, for example, C:\LINKAGE. You may want to put all program files into this directory and any data to be analyzed in another directory, for example, C:\LINKAGE\DATA. Transfer all files received to the LINKAGE directory.

The programs are distributed in compressed form and must be decompressed by a suitable program. On the PC, they may be decompressed with the PKUNZIP program, which is supplied with the LINKAGE programs. Simply type PKUNZIP FINAME, where FINAME is the name of the file to be decompressed, or type PKUNZIP *.ZIP to decompress all zipped files. You may then want to copy the sample files (*.DAT) to your working directory.

Before you can use the programs, note that the CONFIG.SYS file (in the root directory of the boot drive) must contain the following two lines:

```
FILES = 20
DEVICE = ANSI.SYS
```

The number on the first line above is a minimum; in many cases you may want to specify 30 or 40 files. The second line above assumes that the ANSI.SYS file is in the root directory. Alternatively, it may reside in the DOS directory in which case that line should read, for example, DEVICE = C:\DOS\ANSI.SYS. The ANSI.SYS driver is necessary only when you want to use the LCP and LRP programs. Be sure to reboot the computer after modifying the CONFIG.SYS file, or else the changes will not take effect. Users of OS/2 should add the following line to their CONFIG.SYS file (assuming that the system is installed in the C: drive):

```
device=c:\os2\mdos\ansi.sys
```

You may want to run the programs the way they come. To use them with different program constants (eg. different maximum number of pedigrees), make the necessary changes (see section C.2). Also, if your computer has a numeric coprocessor, it is recommended to replace the E+ switch by E- in the SWGEN (SWTHG) file. Then, recompile the programs, for example, by typing

```
COLIG MLINK  or  COLIT CMAP
```

For details on compiling, see section C.3.

Make sure that the LINKAGE directory is accessed by DOS (unless you are working in the LINKAGE directory), for example, by inserting the following line in your AUTOEXEC.BAT file in the root directory:

PATH c:\dos;c:\linkage

If, in the use of the LINKAGE programs, you find what looks like a program error, please let me know. That way other users of the programs can be alerted and the bug can be corrected.

B. BRIEF OVERVIEW OF PROGRAM USAGE

Currently the LINKAGE programs for Borland Pascal are furnished for general pedigrees (version 5.1) and 3-generation (CEPH) pedigrees (version 5.2). A third category, programs for experimental crosses in the mouse, is currently not supported by me. This documentation is generally oriented towards the general pedigree version; differences between the two versions are pointed out where necessary.

B.1 PEDIGREE FILE and MAKEPED PROGRAM

The LINKAGE programs require two input files, a "pedfile" holding the pedigree data, and a "datafile" holding the descriptions of the loci, locus order, etc. (pedfile and datafile are the names of the corresponding files in the program code). Preferably, the first step in the linkage analysis is to create the pedigree file. This must be done using a text editor (word processor) capable of producing ASCII files, for example, MS-DOS EDIT. Word processors usually write documents in their own special manner but most of them can also write files in ASCII format (WordPerfect calls these "DOS text files"; a WordPerfect document is saved as an ASCII file by pressing Ctrl F5).

Write one line of input for each individual, where the following items must be given for each individual (more detailed information is found in the program manual):

- Pedigree name (or number)
- ID name (or number) of given individual
- ID name (or number) of that individual's father
  (0 if father is not in pedigree)
- ID name (or number) of that individual's mother
  (0 if mother is not in pedigree; either both or no parents must be given)
- Sex of individual: 1=male, 2=female
- Phenotype at locus 1
- Phenotype at locus 2, etc.

Each item must be separated from the others by at least one space.

Phenotype symbols depend on the locus type used. Each locus must be coded in one of four possible locus type formats (only Allele Numbers and Binary Factors locus types may be used in the programs for 3-generation pedigrees, and they must specify codominant inheritance). The locus types and corresponding phenotypes are as follows:

a) Affection status: 2=affected, 1=unaffected, 0=unknown. If more than one liability class is used, a second number must be added designating the liability class. Usually used for coding disease loci.

b) Allele numbers: two numbers, corresponding to the two alleles present, eg, 2 5 (alleles 2 and 5 present), or 1 2. 0 0 denotes unknown. Homozygotes and hemizygotes (males in

X-linked case) must be given two identical numbers. Usually used for RFLPs (co-dominant).

c) Binary factors: a sequence of 0's and 1's indicating absence or presence of the i-th factor. Used for dominant marker loci, eg, ABO locus.

d) Quantitative traits: quantitative measurement, eg, CPK level.

For more details on phenotypes, please consult the User's Guide. In the pedigree file, list the phenotypes of all loci known for the individuals. You will later determine with which of the loci you want to do calculations.

One pedigree may be entered after another, each pedigree with its own pedigree id. After the last line is entered, make sure that there are no trailing blank (empty) lines after you exit from the editor. The DOS and other editors append an empty line when you press the <Enter> key at the end of the last input line. So, either you do not press <Enter> at the end of the last line (the cursor then stays at the far right ON the last line), or you insert an end-of-file [EOF] mark in column 1 after the last input line. To enter [EOF], press Alt-2-6 (press 2 and then 6 on the NUMERIC KEYPAD while holding down the Alt key); you should then see a small right arrow on the screen.

Save the file under a name with the extension PRE, eg, as SAMPLE.PRE. It is convenient to use the same file name for the input files of a given problem but distinguish datafile and pedfile by using different extensions.

The sample pedigree file so created, SAMPLE.PRE, must now be processed by the MAKEPED program to make it suitable for input to the analysis programs. Invoke the MAKEPED program (actually, the MAKEPED.BAT file) with the input and output file names on the command line, for example, enter

```
MAKEPED  SAMPLE.PRE  SAMPLE.PED  N
```

where the last N tells the program that no loops are present and that probands should be selected automatically. If N is omitted, follow directions issued by program. Recommended further responses:

- Loops present? -> n (unless your pedigree contains loops)
- Should probands be selected automatically? -> y.

If a pedigree contains a marriage or consanguinity loop, answer Y to the corresponding question from the MAKEPED program and indicate one individual per pedigree at which the loop should be broken. If more than one loop is present in any one pedigree (the maximum number of loops is specified by the constant MAXLOOP), proceed as above and identify as many individuals in each pedigree as necessary at which loops should be broken. For example, if in pedigree 1, loops should be broken at individuals 5 and 9, your interaction with the MAKEPED program would look as follows:

```
Pedigree  --> 1
Person    --> 5

Pedigree  --> 1
Person    --> 9
```

Pedigree --> 0

MAKEPED will then duplicate each of these individuals and will assign the same positive number (different for each pair) in the proband field (column) to the resulting two duplicated individuals. After exiting from MAKEPED, read the pedigree file into your text editor and verify that MAKEPED has made the appropriate duplications and entries in the proband field. If a duplicate individual is to be the proband, this individual must correspond to the first loop to be broken, and the proband field for the two duplicates has to contain a 1 and a 2 (this rule also applies to a single loop only).

Note that for a pedigree file to be suitable for use by the analysis programs, each individual within a pedigree must be numbered sequentially from 1 through n, except for duplicate individuals (loops broken) who can be out of order, where n is the total number of individuals (including duplicated individuals) in that pedigree. Pedigree id's, too, must be numbers, but they need not be sequential and can be in any order. It is the MAKEPED program's job to bring pedigrees into this form required by the LINKAGE programs.

Two example input files (already processed by the MAKEPED program) are provided. PEDIN.DAT contains three-generation pedigrees and one non-CEPH pedigree; PEDIN3.DAT contains only two-generation and three-generation pedigrees and is suitable for testing out the 3-generation programs.

As pointed out above, it is recommended to use the same file names for the same problem but distinguish the associated datafile and pedigree files with the extensions DAT, PRE, and PED, respectively, where PRE refers to the preliminary pedigree file and PED to the one processed by the MAKEPED program. For example, in a study of CF families, the three files would be named CF.DAT, CF.PRE, and CF.PED. For families without loops and automatic proband designation, a third parameter, n, may be given on the command line which tells MAKEPED that no loops are present and that all probands should be chosen automatically. Thus, you might enter MAKEPED SAMPLE.DAT SAMPLE.PED N.

When loops exist in a pedigree and are not declared in MAKEPED, this error may or may not provoke error messages by the analysis programs. Thus, an undetected loop may lead to an apparently normal termination of the programs yet the resulting likelihoods can be completely wrong. To avoid such problems, a program called LOOP was developed by Xiaoli Xie. It detects marriage and consanguinity loops and is automatically invoked after each run of the MAKEPED program.

B.2 DATAFILE and PREPLINK PROGRAM

The datafile should reflect the loci given for each individual, where the loci are ordered corresponding to the order of the phenotypes in the pedigree file. The datafile is best created using the PREPLINK program. After PREPLINK is invoked, it will present various menus with default assumptions on number of loci, locus types, etc. Proceed in the following manner:

(1) Choose the number of loci as present in your pedigree file. When prompted to furnish information on new loci beyond locus 2, simply accept default information, ie, exit and go to next higher locus. When asked for the locus order, simply

enter 0 (for 1 2 3 etc.), since the particular chromosomal order will be given in the analysis program (LCP) anyway.

(2) Select locus types. It is important to do this first, before any more specific locus descriptions are given. Changing a locus type will set most other locus parameters back to their default value.

(3) For each locus, look at its parameters ("see or modify a locus") and adjust where needed. For example, for a disease locus, you may want to adjust gene frequencies to 0.99 and 0.01 so that the disease allele is allele number 2. Generally, choose allele 2 as the disease allele.

(4) If everything is correct, go to the main menu and save the file ("write datafile"), preferably with the extension DAT, for example, under the name of SAMPLE.DAT, corresponding to SAMPLE.PED. Exit from PREPLINK. Should you need to modify a previously created datafile, simply invoke PREPLINK and read in that datafile.

Note that various parameters need not be set in the PREPLINK program as they must be given in the LCP program anyway. These are, for example, locus order, program used, and recombination fractions.

To modify an existing datafile, invoke PREPLINK and use the "m" option to read in that file. If parameters other than recombination fractions are to be estimated in the ILINK program, you will need to modify the datafile in your text editor after leaving the PREPLINK program. The last line of the datafile contains a series of 1's and 0's indicating whether or not a particular parameter should be estimated, that parameter being defined by the order of appearance of the number 1 or 0 (see manual for full details). For example, with 2 loci, if male recombination and female-to-male map distance are to be estimated, there should be two 1's on the last line of the datafile.

On the second but last line, the number given identifies the locus which may have iterated parameters such as gene frequencies. In this case (only recombination fractions estimated), the value of that number is irrelevant as no locus-specific parameters are estimated. Hint: If no locus specific parameters are to be estimated, choose a "locus with iterated parameters" with only a small number or no penetrance classes since these may then potentially be estimated which calls for a large value of the constant MAXN.

Two sample datafiles are provided: DATAIN.DAT may be used in connection with PEDIN.DAT, and DATAIN3.DAT corresponds to PEDIN3.DAT (3-generation families).

B.3 THE LCP SHELL

The LCP program prepares the data for a series of production runs. You will be able to make various choices, eg, loci to be used, and to set parameter values such as recombination fractions. All these choices will be saved in a batch file (command file) that you can run by typing its name after exiting from the LCP program. The default name of that command file is PEDIN.BAT.

After you invoked LCP, change the file names presented on the first screen as needed. Usually, you will only have to adjust the names of your pedigree file and datafile (parameter

file). When you have chosen these file names, move back and forth among the screens with the PgDn and PgUp keys. However, watch for the screen identified by the title, COMMAND SCREEN, shown in reverse video. Pressing PgDn on such a command screen will save in the batch file the choices you just made, and failure to press PgDn on a command screen will not save these choices. Leave the LCP program by pressing Ctrl-Z.

To execute the runs you selected in LCP, enter the name of the batch file (PEDIN by default). If nothing happens, you failed to press the PgDn key on the Command screen in which case you have to invoke LCP again and repeat the selections desired.

Note the following feature of LCP: When choosing ILINK as the analysis program, generally all recombination fractions between loci will be estimated. If you want to keep some of them fixed at their initial value, enter the recombination fraction with an equal sign in front of it.

You may inspect the PEDIN.BAT file with your text editor. It consists of a sequence of commands (DOS commands and calls to programs). Essentially, it extracts loci information from your input files and prepares new input files (called datafile.dat and pedfile.dat) for the Unknown program and then invokes the analysis program. After the runs are completed, all intermediate files are deleted. If you do not want intermediate files deleted, you have to invoke the command file with the command line parameter NODELETE, eg, by entering PEDIN NODELETE. One reason for doing that would be, for example, to retain the files (DATAFILE.DAT, PEDFILE.DAT, IPEDFILE.DAT, SPEEDFIL.DAT) containing the loci extracted from the original files and to modify DATAFILE.DAT so that parameters other than recombination fractions can be estimated by ILINK; currently, this cannot be done through LCP.

LCP cannot yet exploit all the features of the analysis programs (MLINK, LINKMAP, ILINK). For example, a female/male distance ratio different from 1 is not allowed for MLINK although the MLINK program when used directly will accept any such ratio, and haplotype frequencies cannot presently be specified through LCP.

C. TECHNICAL NOTES

C.1 LIMITATIONS

The programs may not carry out calculations when only a single locus is used. For such cases, expand the data by adding a dummy marker locus at which everybody is homozygous. Also, if a single individual should be part of your pedigree data, add two parents with unknown phenotypes and have these three individuals form one pedigree.

C.2 PROGRAM CONSTANTS

A number of constants may be set by the user prior to recompiling the programs. These constants define upper limits for number of loci, number of alleles per locus, etc. Since no array or other variable can exceed 64K bytes in size, the constants cannot be set freely. In practice, the major restriction is that MAXHAP must not be larger than 126. Also, the product, MAXFEM x MAXPED, must not be larger than 65,536, where MAXPED = max. no. of pedigrees, MAXHAP = max. no. of haplotypes, and MAXFEM = MAXHAP x (MAXHAP+1)/2 = max. no. of genotypes (the

latter is automatically computed from MAXHAP). For example, with MAXHAP = 64 (MAXFEM = 2080), no more than MAXPED = 31 pedigrees can be analyzed in a single run. Also, in the ILINK program, even a relatively small maximum number of liability classes, eg, MAXLIAB = 8, calls for a high maximum value of parameters that can be estimated in ILINK, eg, MAXN=60. Generally, at most 5 marker loci can be analyzed jointly in addition to a disease locus. Of course, most of these restrictions are imposed by Borland Pascal and do not exist in the OS/2 version.

In the 3-generation pedigree programs, the maximum number of loci and alleles is much higher than in the general pedigree programs. This increase works only for codominant loci in CEPH type families and is achieved by breaking the original families into smaller independent units with fewer loci each. It is not possible to predict the maximum number of loci after transformation. The CFACTOR program writes into a file (CFACTOR.OUT) the number of new pedigrees and individuals produced by the transformation.

These constants reside in files specific to each program (version 5.1: in the file GENC.PAS, which is accessed by all programs). These files are named ???C.PAS, for example, MLKC.PAS for the MLINK program, and LDSC.PAS for the LODSCORE program (version 5.2 only). Near the top of these files, a "TYPE real=" statement determines the number of bytes used for real variables. The following choices are possible:

TYPE real=	Variable length (bytes)	Mantissa length (bits)
single	4	23
(real)	6	39
double	8	52
extended	10	63

Whenever possible one should work with double precision variables (see section C.6, Underflows). To work with Borland Pascal real variables, indicated by "(real)" in the above table, disable the "TYPE real=" statement by setting it within curly brackets {}.

C.3 COMPILING

To take advantage of Borland Pascal's enhanced capabilities vis-a-vis Turbo Pascal, I converted most of our programs from Turbo to Borland Pascal. The following features are characteristic for the new program versions:

- Programs may be compiled to run under DOS protected mode. That way they can address up to 16MB of memory. Thus, a heap overflow error is no longer likely. On the other hand, the data segment (memory for variables and arrays) is still limited to 64KB.

- Programs may be compiled to run under Windows. Then, they also run in protected mode.

- Batch files are available for easy compilation of the programs for any of the target systems (DOS real mode, DOS protected mode, Windows). For DOS real mode, the batch files are compatible with Turbo Pascal 5.0 or higher.

- The executable programs shipped on floppies or available on the ftp site (york.cpmc.columbia.edu) have been compiled to run under DOS protected mode. That way, they are somewhat (perhaps 10%) slower than when compiled for DOS real mode but, as mentioned above, can use much more memory. Users may recompile the programs with Borland or Turbo Pascal.

  To compile one of the programs, simply type, for example,

  COLIG T MLINK <Enter> (general pedigrees) or
  COLIT T CILINK <Enter> (3-generation pedigrees) or
  COMPILE T SLINK <Enter>

where COLIG stands for COmpile and LInk General pedigree programs, and T must be replaced by an appropriate code. That code identifies the target operating environment of the compiled program. Make sure that the compiler (TPC.EXE or BPC.EXE) will be found by DOS (eg. is in the path).

T =	Compiler	Target system
nothing	TPC	DOS real mode
D	BPC	DOS real mode
P	BPC	DOS protected mode
W	BPC	Windows

For example, to compile SLINK such that it will run under DOS protected mode, you would enter

  COMPILE P SLINK.

Borland Pascal can produce programs running under DOS real mode (the "usual" DOS) in which case memory is limited to 640KB or less. Under DOS protected mode (letter P after COLIG) and Windows (letter W after COLIG) programs may dynamically address up to 16MB of memory. Notice that this is DPMI (DOS protected mode interface) memory. The default amount of DPMI memory available is one-half the installed memory (in a DOS window of OS/2, the DPMI memory may be set by the user; default: 4MB). This default may be changed with the RTM environment variable (see Borland Language Guide page 213).

To run protected mode programs, the two Borland files RTM.EXE and DPMI16BI.OVL (supplied) must be either in the current directory or in the path.

Various compiler switches allow producing executable programs suitable for your particular needs. They generally reside in an include file, SWGEN.PAS (SWTHG.PAS), that is furnished with the source code and is read by each of the LINKAGE programs and units. Part of it looks as follows:

```
{$O-}               {Overlays; + or -}
{$N+}               {Use numeric coprocessor; + or -}
{$E+}               {Emulate coprocessor}
```

In combination with the precision of real variables (see above), the recommended settings are as follows:

```
System with coprocessor:          real=double    N+  E-
System without coprocessor:       real=double    N+  E+
Small data set, no coprocessor:   real=turbo     N-  E-
```

Any precision other than turbo requires N+. If no coproces-

sor is installed and precision is other than turbo, the E+ switch is mandatory. If there is no danger of an underflow, on machines without a coprocessor, DEFINE turbo will produce faster running code than DEFINE double.

C.4 OVERLAYS

Using overlays (with the O+ compiler switch) can reduce the load size of the program by 30K bytes or more. However, due to the disk accesses required, the program will run considerably slower if the overlay file resides on disk as compared to a non-overlaid program. The best solution is to transfer the overlay file (eg, MLINK.OVR) to a RAM disk and set the DOS path such that the RAM disk is accessed; then, only a small penalty must be paid for using overlays. But make sure that the overlay file is deleted in the current directory or in the directory where the executable program resides. Experimenting with the possibility of loading the overlay file into EMS memory have shown that this results in rather long execution times.

Fastest program execution is achieved when the program resides all in memory, ie, when it is not overlaid. However, it then occupies more memory than when it is overlaid so that one runs out of heap space sooner (see error messages, below).

C.5 NUMERIC COPROCESSOR

It is highly recommended to use the LINKAGE programs on a machine with a coprocessor (Intel 8087, 80287, 80387, or an 80486 processor) (see below for further explanations). If a program is compiled with the N+ and E+ compiler switches, it will sense the presence of a coprocessor and will use it when it is present. On some machines, the signals received by the program may be wrong so that, eg, a coprocessor is assumed present while in reality it is absent. Such a situation will crash the program and probably freeze the machine. To prevent this from happening (necessary only on some non-IBM machines), type the following DOS command once before using the programs:

```
SET 87=n
```

when no coprocessor is present, and

```
SET 87=y
```

when a coprocessor is installed. This command is profitably made part of the AUTOEXEC.BAT file which is executed automatically at start-up.

Whereas a program compiled with N+ and E+ can run on machines with or without coprocessor, the emulation code produced by E+ will increase program size by about 10K bytes. Furthermore, software emulation greatly increases computation time if no coprocessor is installed.

C.6 UNDERFLOWS

Underflows occur when a real number becomes smaller than a critical limit, eg, 10^(-38) for reals of the 'turbo' or 'single' type where ^ stands for exponentiation. In Borland Pascal, when an underflow has occurred in a variable, its value will simply be set equal to zero and computation continues. This may lead to apparent errors and inconsistencies that are difficult to pinpoint. Underflows may largely be avoided by choosing a type of

real variable with a low critical bound, eg, the 'double' type.

As outlined above, on machines without a numeric coprocessor, programs run fastest when the 'turbo' type of reals is defined. However, this is the type most prone to underflows. Using TYPE real=double and the compiler switches N+ and E+ will largely prevent underflows but will lead to an increase in running time since the action of the coprocessor is emulated in software. Users without a coprocessor in their machines will, therefore, have to carefully weigh the different compiler settings for compilation.

C.7 PROGRAM UNITS

Under Borland Pascal 5, the Linkage programs are too large to be compiled as a single routine. Therefore, they are broken into a number of units, each of which contains a set of procedures or functions.

C.8 ERROR MESSAGES

The procedure ERRTRAP reports errors in plain English rather than providing error numbers only (exception: stack overflow; see below). Some of the less than obvious error messages are explained below.

=Range check error= One of the constants is too small for the problem to be analyzed. Check each of these constants. For example, the number of haplotypes, h, may have to be as large as the product of the number of alleles for all loci. This error message may occasionally be quite cryptic and it may be difficult to determine which of the constants must be increased. For example, in ILINK, having a large number of penetrance classes requires a high value of MAXN, the max. number of parameters that can be estimated in ILINK, since penetrances may potentially be estimated in ILINK (if at the end of the datafile, the locus with iterated parameters is the one for which penetrance classes are defined).

=Stack overflow (error number 202)= The program ran out of stack space. This may occur when the stack segment is too small to hold all local variables in which case one must increase the stack size in the M compiler switch (the first of the three numbers in curly brackets) at the beginning of the main program. However, the stack segment is usually large enough and the most common reason for the occurrence of this error is the presence of an undeclared loop in a pedigree.

=Heap overflow= There is not enough free (dynamically allocated) memory to hold all the data. This error should only occur when you compile for DOS real mode. A program running in DOS protected mode or under Windows can address up to 16MB of memory. To reduce memory requirements the following actions may be taken:

1) Reduce program constants to their smallest possible values.

2) Analyze only one pedigree at a time and set the max. number of pedigrees to 1.

3) Reduce the compiler switch 'DEFINE double' to 'DEFINE single'.

4) If you have a coprocessor installed, be sure to use the

compiler switch E-. This will reduce program size so that more memory is available for data.

5) Compile the program with overlays using the O+ compiler switch.

6) Set the compiler switch R-. Note that this may freeze the computer when an array bound is exceeded.

=Data segment too large= The variables and arrays occupy too much memory. Reduce some of the program constants to make array sizes smaller, or go from double to single precision. It may happen that for the same programs this error occurs when compiling for Windows but not for DOS.

C.9 'RUN' BATCH FILE

This batch file allows running any one of the Linkage programs without going through the LCP shell provided that all the loci in the data file are to be analyzed (no possibility of extracting loci). To initiate this batch file, execute the command

RUN DATNAME PEDNAME PROGNAME

where DATNAME is the name of the file holding the locus descriptions (the datafile, as processed by the PREPLINK program), PEDNAME refers to the file holding the pedigree data (as processed by the MAKEPED program), and PROGNAME is the name of the program to be used.

The major reason for using the RUN batch file is to be able to make use of some features not yet implemented in LCP (see end of section on LCP, above), in particular, haplotype frequencies which may be important in risk calculation.

D. LITERATURE

Lathrop GM, Lalouel JM, Julier C, Ott J: Strategies for multilocus linkage analysis in humans. Proc Natl Acad Sci USA 81:3443- 3446, 1984

Lathrop GM, Lalouel JM, Julier C, Ott J: Multilocus linkage analysis in humans: detection of linkage and estimation of recombination. Am J Hum Genet 37:482-498, 1985

Ott J: Analysis of Human Genetic Linkage (revised edition). Johns Hopkins University Press, Baltimore, 1991

Terwilliger JD, Ott J: Handbook of Human Linkage Analysis. Johns Hopkins University Press, Baltimore, 1994

E. The LINKLODS program

The LINKLODS program reads output from the LINKMAP or MLINK (LINKAGE) program and for each family converts log likelihoods to lod scores. The LINKAGE output file (input to the LINKLODS program) will usually be FINAL.OUT (produced when PEDIN.BAT command file is run), but other file names may be given at run time in response to a prompt by the program. While the analysis programs do not furnish family specific lod scores, these may be obtained as an option in the LRP program but many users prefer to use LINKLODS.

In the input file, for a collection of families, an initial

set of likelihoods with one of the theta values being equal to 0.5 has to precede those sets of likelihoods, for which lod scores should be calculated. Several such initial 'baseline' sets of likelihoods may occur throughout the input file. A sample input file, FINAL.OUT, is provided.

Resulting lod scores will be written to the file FINAL.LOD, and an existing file by that name will be overwritten.

Files names may also be given on the command line. If only one file name is provided, it will be taken to be that of the input file. A second file name is that of the output file (overrides FINAL.OUT).

Notice that the LINKLODS program makes certain rigid assumptions on the structure of the input file as produced by the LINKAGE programs. For example, the first likelihood must be on the fourth line after the line, which lists the theta values. Therefore, if the input file has been manipulated, the LINKLODS program may no longer be able to process it properly and will issue some error message.

F. ANALYSIS HINTS

F.1 PENETRANCE/LIABILITY CLASSES IN LIPED AND LINKAGE

In the LIPED program, each phenotype is associated with an array of penetrances, that is, the conditional probabilities that the phenotype is observed given a genotype. In the Linkage programs, one may code phenotypes in several ways, depending on the type of locus considered (binary factors, affection status or quantitative phenotypes locus). With a binary factors locus, one may code for codominant or dominant phenotypes but not both types mixed. This sometimes poses a problem, for example, in the following situation. Assume a locus with two alleles, A and B, whose individual presence in a person can usually be detected (codominant situation). Sometimes, however, a test is used that detects A only (dominant situation). Using conditional probabilities (penetrances), this situation is represented in LIPED as follows:

	Phenotypes				
Geno-	Dominant		Codominant		
type	A+	A-	AA	AB	BB
A / A	1	0	1	0	0
A / B	1	0	0	1	0
B / B	0	1	0	0	1

In the Linkage programs, it is not possible to allow for all these phenotypes at a binary factor locus. A simple general solution for using tables such as the one above in the Linkage programs is as follows. Define the locus in question as an affection status locus with as many liability classes as there are columns in the table above. In the pedigree file of the Linkage programs, each phenotype is then represented by two numbers, 2 i, where i is the column number in the table above, that is, each individual is defined to be affected, except that the unknown phenotype is coded as 0 1. Each column in that table represents a liability class whose penetrances (the entries

in the column) must be furnished in the datafile.

This coding scheme may be wasteful in the number of liability classes needed. Depending on the particular situation, one may be able to apply a similar coding scheme requiring a smaller number of penetrance classes. In the given example, above, a possible solution is the following. Define an individual as affected when the A-allele is detected, and distinguish 3 liability classes, depending on whether the A-allele is seen in the dominant or codominant situation. The correspondence between phenotypes in Liped and in Linkage is then as follows:

Liped	A+	A-	AA	AB	BB
Linkage	2 1	1 1	2 2	2 3	1 1

In the datafile, the following liabilities must be given for each genotype and each liability class:

	Liability class		
Genotype	1	2	3
A / A	1	1	0
A / B	1	0	1
B / B	0	0	0

F.2 CODING FOR MUSCULAR DYSTROPHY

Generally, in the LINKAGE programs, one would code phenotypes (CK levels for females, aff. or unaff. for males) as a quantitative trait locus. A special case in which simple coding as an affection status locus is possible is the following.

Males: affected or not affected
Females (not affected): CK+ or CK-
(CK = creatine kinase level, high or low)
Alleles: D = disease allele
d = normal allele

Possible coding scheme for LIPED:

	Phenotypes				
	females			males	
Genotype	AF	CK+	CK-	AFF	NA <- phenotype codes
D/D or D/y	1	0	0	1	0
D/d	0	.66	.34	*	*
d/d or d/y	0	.05	.95	0	1

Unknown: special phenotype, eg, blank.
AF = affected female
* = value irrelevant (X-linked case)

In LIPED, this coding scheme has the effect that no unaffected female, whether or not she has an elevated CK level, is ever assumed to be homozygous for the disease allele.

In LINKAGE, this case may be treated by the general method outlined in section 1, above, leading to 3 penetrance classes. To code such a situation with a single liability class, one may adopt the following coding scheme in the Linkage programs:

D = allele 1, d = allele 2, risk allele = #1.

Define disease status as having an elevated CK value. This works fine when only unaffected females are observed (usual situation).

Phenotypes in LIPED	MLINK (in pedfile)	
CK+	2	\ unaffected
CK-	1	/ females
AFF	2	affected male
NA	1	unaffected male

Unknown phenotype: 0

In the datafile, the penetrances (= probabilities of being affected) are given as follows:

Females		Males	
Genotype	Penetrance	Allele	Penetrance
1 / 1	1	1	1
1 / 2	.66	2	0
2 / 2	.05		

Note that "affected" (CK+) females now potentially are homozygous for the disease allele (CK- females still cannot be homozygous for the disease allele). If this is undesired, or if truly affected females are present, one better uses the scheme with 3 penetrance classes corresponding to the LIPED notation.

F.3 IDENTIFYING OBLIGATE CARRIERS

To identify an obligate heterozygote in LIPED, one might label such an individual with the phenotype NA2 and define the following penetrances:

	Phenotypes			
Genotype	AFF	NA1	NA2	<- phenotype codes
D / D	1	0	0	
D / d	0	1	1	
d / d	0	1	0	

Again, this case may be treated as outlined in section 1, above. Using only 2 rather than 3 liability classes, one may define these in the datafile as follows:

	Penetrance class	
Genotype	1	2
D / D	1	1
D / d	0	0
d / d	0	1

In the pedfile, the following phenotype codes are used:

	Phenotypes in LIPED	MLINK
AFF	2	1
NA1	1	1
NA2	1	2

F.4 EVERYBODY UNKNOWN AT ONE LOCUS

In multipoint linkage analysis, for a given family pedigree, it sometimes happens that all individuals have not been tested at one of the loci and thus have phenotype 'unknown' at that locus. In the present implementation of the Linkage programs, the presence of many unknowns slows down execution speed. There is, however, a simple remedy. If everybody in that pedigree is given the same homozygous phenotype (uniquely identifying the homozygous genotype), this will not change the lod score but will considerably increase computing speed. This feature has now been implemented in the UNKNOWN program except when allele frequencies are to be estimated.

F.5 RUNNING ILINK ON NEW DATA

With new data and several marker loci, it is often useful to first find or confirm estimates of interlocus distances, that is, to run the ILINK program for the marker loci only. However, before doing that, it is a good idea to do one run with the MLINK program to verify that the likelihood is nonzero in all pedigrees. If the likelihood is zero in one or more pedigrees, for example, due to genotype inconsistencies, then the ILINK program will still try to maximize the likelihood and will, of course, fail but only after running for a possibly very long time.

F.6 RISKS FOR X-LINKED RECESSIVE DELETERIOUS TRAITS

With X-linked recessive deleterious traits, for a female founder individual (no parents in pedigree), the prior probability, q, of being a carrier of the disease gene is a multiple of the mutation rate, □ For example, in Duchenne muscular dystrophy (DMD), q=4□(Murphy and Chase, "Principles of Genetic Counseling"). In the likelihood calculation of pedigree data, on the other hand, the prior probability of a founder's genotype is always determined by the gene frequency, p. The prior probability that a founder woman is heterozygous is given by 2p(1-p). To implement the prior probability, q, that she is heterozygous for an X-linked recessive deleterious gene, in the likelihood calculation, one has to choose the gene frequency of the deleterious gene, p, such that q=2p(1-p) or, approximately, p=q/2. For example, in DMD, when the mutation rate is assumed to be equal to □ the gene frequency of the disease allele must be taken to be equal to p=2□