(1) GENERAL INFORMATION ON MAPMAKER/QTL VERSION 1.1

MAPMAKER/QTL is a companion program to MAPMAKER/EXP which allows one to map genes controlling poygenic quantitative traits in F2 intercrosses and BC1 backcrosses relative to a genetic linkage map.

A list of references can be found in the 'Release Notes' section of the MAPMAKER/QTL manual and in the READ.ME file included with MAPMAKER.

ABOUT MAPMAKER/QTL Command

Summary: License and Contact Information for MAPMAKER/QTL

License to use this program for non-commercial purposes is provided free of charge. This software and documentation may be freely redistributed, although only under specific terms. This program is provided without any warranty of any kind. See the License Agreement included in the Release Notes and in the READ.ME file for details.

MAPMAKER/QTL 1.1 was written by Stephen E. Lincoln, Mark J. Daly, and Eric S. Lander of the Whitehead Institute for Biomedical Research and the M.I.T Center for Genome Research, Massachusetts Institute of Technology, Department of Biology. Further information is available from:

MAPMAKER
c/o Eric Lander
Whitehead Institute
9 Cambridge Center
Cambridge, Massachusetts 02142 USA

FAX: 617-258-6505
BITNET: mapm@mitwibr
INTERNET: mapmaker@genome.wi.mit.edu

Source code, updates, and other information are available on the Internet by anonymous FTP from genome.wi.mit.edu.

RELEASE NOTES Information

Summary: Information on the Release Notes

Included with MAPMAKER Version 3.0 and MAPMAKER/QTL Version 1.1, you will find a Release Notes document describing:

- What MAPMAKER is
- How to get MAPMAKER
- How to Run MAPMAKER on a PC Compatible Running DOS
- How to Run MAPMAKER on a Sun SPARCStation Running SunOS
- How to Run MAPMAKER on an Apple Macintosh Running A/UX
- License Agreement
- GNU Readline and GhostScript License Information
- Where to Go from Here
- References

and other issues. The Release Notes are included in printed form in the manual, and as a READ.ME file with the program itself. Other important documentation includes:

File	Manual Section
INSTALL.ME	Installation Guide
FEED.ME	Data preparation guide
CHANGE.ME	List of changes since last release, and To-Do list

STARTING MAPMAKER/QTL Information

Summary: How to Start MAPMAKER/QTL and Available Options

Exactly how you start MAPMAKER/QTL depends on how it was installed in your system -- see the Installation Guide included with MAPMAKER/QTL for details. Usually, you will want to first go into the directory which contains your data files (using the "cd" command). Then, you can start MAPMAKER/QTL by simply typing at a DOS or Unix prompt:

```
qtl
```

On UNIX machines, you will be able get such a prompt by opening a terminal window (Sun's Command-Tool or A/UX's CommandShell). Alternatively, to run MAPMAKER/QTL on a Unix system under X-Windows (or OpenWindows), you may wish to type:

```
xqtl &
```

As a third alternative, depending on your system, you may be able to start MAPMAKER/QTL instead by simply clicking on a particular icon or file. Again, see the Installation guide for details.

You may supply additional instructions to MAPMAKER/QTL on the command line. On a Unix system (including Suns and A/UX), these include:

-simple
Do not use any fancy terminal capabilities (e.g., highlighting, screen clearing, command editing, etc.). On DOS, this means that ANSI.SYS is not required. On Unix, this helps with unusual or buggy terminals and terminal-emulators.

-nomore
MAPMAKER/QTL produces output one screenful at a time (with "Hit Return for More" breaks in between) from certain commands, including 'help', 'translate', and all 'list' and 'show' commands. This is useful on DOS systems or on Unix terminals which do not remember text that has scrolled off the screen. The '-nomore' option turns this off: "more" breaks never interrupt output. Regardless of whether you use this option, "more" breaks are never used by time-consuming analytical commands, no matter how much stuff they try to print.

-clear
Clearing the screen can make some output faster and prettier, although it wreaks havoc with some terminals or programs that try to remember text that has scrolled off the screen. By default, we allow the screen to be cleared on DOS systems, and not on Unix systems. The '-clear' option precisely reverses these assumptions.

-photo xxx
MAPMAKER output is copied to the named file (xxx). If the file exists, MAPMAKER output is appended onto the end of it.

-out xxx
MAPMAKER output is copied to the named file (xxx). If the file exists, it is first erased. Only one of "-photo" and "-out" can be used.

-run xxx
Commands are run from the named text file (xxx) before commands are accepted from the keyboard. If the file ends with a 'quit' command, MAPMAKER quits without prompting for additional commands.

The options for the DOS version are identical, except that instead of dash characters (-) you should use forward slashes (/). For example, you might type the DOS command:

```
qtl -nomore -run /scripts/dothis.in      (UNIX)
```

```
qtl /nomore /run \scripts\dothis.in     (DOS)
```

The Unix and DOS '<', '>', and '|' characters indicating for input and output redirection, as well as the Unix '&' argument for running a job in the background, all work with MAPMAKER/QTL as well.

A small number of DOS or Unix environment variables are also used. On a Unix system, you set these with a 'setenv' command in your '.cshrc' file (assuming you use 'csh' as your shell) or in a shell script (e.g. 'xqtl'). Under DOS, you set these with a SET command in your AUTOEXEC.BAT file or in some other batch file (e.g. QTL.BAT). The variables used include:

MAPM_LIB
The directory in which MAPMAKER/QTL will find it's help files and other items it needs. This should usually be set to something like "/home/joe/mapmaker" or "/usr/local/mapmaker" on Unix, or "C:\MAPMAKER" on DOS. This variable must be set to use the online help feature: see the Installation Guide for details.

LINES
The number of lines the screen/terminal/window can display at once. MAPMAKER/QTL tries to figure this out on its own, although if the LINES variable is set, it's value takes precedence. Set LINES if you use the DOS "mode con:lines=xx" command to change the number of lines of text displayed on EGA and VGA monitors or under Windows 3.1.

TERM
The type of terminal in use. MAPMAKER/QTL recognizes only a small number of terminal type names, even though many terminals are ANSI (and thus MAPMAKER/QTL) compatible. Try setting "TERM" to "ansi" or "vt100" if MAPMAKER/QTL is confused (for example, if it does not display "Hit return for more" in reverse-video when you type "help"). Usually set automatically on Unix systems, and usually not needed on DOS systems.

SHELL (Unix) or COMSPEC (DOS)
The name of the program to be started by MAPMAKER/QTL's "system" command. Usually set automatically by the system.

TUTORIAL Information

Summary: Information on the MAPMAKER/QTL Tutorial

MAPMAKER/QTL is provided with a printed tutorial, which illustrates many of the basic MAPMAKER/QTL commands which you will likely use often. The data set used in the tutorial, sample.raw, is provided with the MAPMAKER/QTL distribution so that you can work along with the tutorial. The tutorial itself is not available on-line, however. See the Release Notes (or the READ.ME file) for information.

ENTERING COMMANDS Information

Summary: How to Type Commands into MAPMAKER/QTL

When started, MAPMAKER/QTL responds with its start-up banner and a prompt for the first command:

```
1>
```

Whenever a prompt like this is displayed, any valid MAPMAKER/QTL command may be entered. All commands consist of one or more words separated by spaces, and may be followed by one or more "arguments" to the command. Arguments might include locus numbers, computer file names, or numeric values. Input is always terminated by hitting the RETURN key. In most cases, MAPMAKER/QTL ignores upper-case/lower-case distinctions in its input (the rare exceptions include arguments which need to be case-sensitive, such as file names under Unix). For example, the following are valid MAPMAKER/QTL command entries:

```
1> load data plants.data
1> show linkage maps
1> scan
```

To make typing commands easier, all commands may be abbreviated by only a few characters: see 'abbreviating commands' below for details. In addition, a keyboard-based command-line feature may be enabled on your version of MAPMAKER/QTL: see 'keyboard editing' below for more details. Also, MAPMAKER/QTL remembers commands you have already typed, allowing you to quickly execute the same commands again: see the discussion of 'previous commands' below.

ABBREVIATING COMMANDS Information

Summary: How to Abbreviate Commands You Type

MAPMAKER/QTL allows you to abbreviate commands in a number of ways. In general, only enough of a command name needs to be entered in order to distinguish it from other MAPMAKER/QTL commands. Not all of the words in a multi-word command (such as "forget all scans") are needed, nor are all letters of any particular word, as long as the entry is not ambiguous. When more than one word of a two or three word command is used, the words must be separated by spaces. Any arguments to a command must also be separated by spaces.

For example, to turn MAPMAKER's 'auto save data' option on, you may type:

```
1> auto on
```

Or to enter a sequence of intervals, type (for example):

```
1> seq [chrom1]
```

Some frequently used commands also have one or two letter abbreviations which may be used to invoke the command very rapidly. For example, you can enter a sequence by typing:

```
1> s [1 2 3 4]
```

or load a data file by typing:

```
1> ld myfile.data
```

On some systems, an elaborate keyboard-based editing may be enabled, making it even easier to type commands. See 'keyboard editing' below for details.

KEYBOARD EDITING Information

Summary: Editing Commands and Sequences Using the Keyboard

On Unix systems (including Suns and A/UX), an interactive command editing facility may be optionally installed into MAPMAKER/QTL. If this feature is installed, you can recall previously typed commands using the up-arrow and down-arrow keys, edit them (e.g., to fix a mistake, add a locus to a sequence, etc.), and then hit RETURN to execute the command. The set of keys which work includes the standard ones (arrows, DELETE, BACKSPACE, etc.). Various control-key combinations also do useful thigs, although you will only find these familiar if you have used some version of the Emacs text editor. Some control keys include:

```
control-a        move to beginning of line
control-e        move to end of line
control-f        move forward one character
control-b        move backward one character
control-d        delete character
control-k        delete the rest of the line
```

The ’edit’ command can similarly be used to edit sequences: see the ’edit’ command below for information.

The keyboard editing functions are implemented using the GNU Readline package, published by the Free Software Foundation. See the Release Notes for details and licensing of this package.

The keyboard editing package (including the ’edit’ command) does not at present work on DOS systems. However, the simple default DOS 5.0 input editing features (e.g. F3 key, etc) do work, allowing you to at least edit and re-enter the last line you typed. DOSKEY has no effect on MAPMAKER/QTL’s keyboard editing (or lack thereof).

POSTSCRIPT OUTPUT Information

Summary: What to Do with PostScript Graphic Output

MAPMAKER/QTL’s "draw scan" command produces graphic output as standard level 1 PostScipt files (text files with the extension ".PS") which can be displayed and printed in a number of ways. This issue is discussed in detail in the installation instructions in the manual (or the INSTALL.ME) file. Here we provide a brief summary:

These files print "as-is" simply by sending them to a PostScript printer (such as an Apple LaserWriter) in the usual way: On Unix (including Suns and A/UX) you can usually print these files using the ’lpr’ command (although you should not use lpr’s ’-p’ option with PostScript files). For example:

```
lpr myscan.ps
```

On a DOS system, the "PRINT" command will usually work:

```
print myscan.ps
```

In either case, you can get a Unix or DOS prompt from which to issue this command using MAPMAKER/QTL’s ’system’ comand, described below. Both examples above assume that your software is configured correctly and in the default way. Ask your computer support people for details.

The "GhostScript" package included with MAPMAKER/QTL on some systems may be helpful for displaying PostScript files in a window under either X-Windows (including Sun’s OpenWindows), or under Microsoft Windows. Again, see the Installation Instructions for details.

(2) BASIC MAPMAKER/QTL COMMANDS

There are several MAPMAKER/QTL commands you will find yourself using frequently. These include:

```
'help' command, to read on-line help information
'photo' command, to save MAPMAKER output to a text file
'load data' command, to load a data set for analysis
'quit' command, to save your data set and exit from MAPMAKER/QTL
```

Type 'help' followed by the command name above for more information on that command.

HELP Command (abbreviation '?')

```
Summary:     Read On-Line Help Information
Argument:    <command name or topic number>
Default:     with no arguments, display a list of all commands and topics
```

The reference section of the manual is available through MAPMAKER/QTL while the program is running. For a list of all MAPMAKER/QTL commands and other topics on which help is available, simply type "help". For help on a particular command, type "help" followed by the command name (or, at least enough of it so as not to be ambiguous). For example, you may type:

```
1> help prep
```

for help on the "prepare data" command. For help on one of the other listed topics, type "help" followed by the topic number.

PHOTO Command

```
Summary:     Begin Saving MAPMAKER/QTL Output to a Text File
Argument:    <file-name>
Default:     with no arguments, display the current 'photo' status
```

The "photo" command is used to keep a transcript of the MAPMAKER/QTL session on the computer. If one types "photo <file name>", for example,

```
1> photo new7q.out
```

all MAPMAKER/QTL input and output from that point on will be copied into the file specified (here, the file named "new7q.out"). Typing "photo off" or quitting MAPMAKER/QTL terminates this process and closes the photo file. Typing "photo on" resumes copying to the previously specified photo file, or, if no photo file name has been specified previously, photo-ing is started to the file "session.out". Typing "photo" alone displays the current state of the photo option. The default extension for a transcript file is ".out".

For example,

```
1> photo june11.out
photo is on.  file is 'june11.out'.

2> seq [11 12 13 14]      <-    From this point on, all MAPMAKER input
        .                       and output is copied to the file
        .                       'june11.out'.
        .

5> photo
photo is on.  file is 'june11.out'.
```

```
    6> photo off
    photo is off.

    7>                      <-         Input and output will no longer
                                       be copied.
```

Note that when MAPMAKER/QTL is photo-ing to a particular file, the transcript is APPENDED to the end of that file: none of the file's previous contents are lost. Successive MAPMAKER/QTL transcripts may be collected into single files in this way.

LOAD DATA Command (abbreviation 'ld')

Summary: Load Data Files into MAPMAKER/QTL
Argument: <file-name>

The "load data" command loads prepared data into MAPMAKER/QTL for analysis. Only one data set may be loaded at any time: any data previously loaded will be forgotten.

The name of the data file should be given as an argument. A directory may optionally be specified in the manner supported by your system. The data file(s) must use the default extensions (e.g. ".data" for the main genetic data file), which you do not need to specify, although you may. For example,

```
    1> load data example
    1> load data /users/joe/chromosome7              (UNIX)
or  1> load data C:\joe\expt1\maize                  (DOS)
```

When a data set is loaded, MAPMAKER/QTL will also load in the corresponding ".traits" file, which was also created using the "prepare data" command. MAPMAKER/QTL may also load in a number of option settings as well as some saved scan results from the associated ".qtls" file. This feature lets you quit and restart MAPMAKER/QTL, resuming your analysis right where you left off. Because of this however, some of MAPMAKER/QTL's option settings may change from their previous values after loading data: If you want to set any of MAPMAKER/QTL's options before running any analyses, we recommend that you do so only after loading your data set. A list of the items which are saved in this manner is provided below, under the description of the "auto save data" option.

SAVE DATA Command

Summary: Save MAPMAKER/QTL Data and Status Info to Disk
No Arguments

This simple command instructs MAPMAKER/QTL to immediately update all of your data files with its current option settings, any new "scan" and compare results, etc. This behavior is precisely the same as that used when quitting MAPMAKER/QTL with the "auto save data" option on.

QUIT Command (abbreviation 'q')

Summary: Quit from a MAPMAKER/QTL Session
No Arguments

The "quit" command ends a MAPMAKER/QTL session. If the "auto save data" option is "on", the data files will be updated to include current option and parameter settings as well as any new "scan" and "compare" data computed.

(3) SEQUENCE COMMAND AND RELATED FEATURES

Before using most of MAPMAKER/QTL's analysis functions, you need to specify first the possible QTL positions which should be considered in the analysis. MAPMAKER/QTL's sequence facility is quite powerful and, by allowing you to specify many possible QTL locations (or combinations of locations for multiple QTL models) as well as other characteristics of the model in question, can enable you to perform complex analyses very simply. Commands related to the use of sequences include:

'Sequence' Command
'History' Command (Sequence History Feature)
'List Loci' Command
'Edit' Command

MAPMAKER/QTL also allows you to assign names to particular sequences, and to then simply refer to these sequences by name. Commands which help you do this include:

'Let' Command
'Names' Command
'Forget Named Sequence' Command

SEQUENCE Command (abbreviation 's')

Summary: Set the Sequence of Intervals to Look for QTLs in
Argument: <sequence>
Default: with no arguments, displays the current sequence

In effect, setting MAPMAKER/QTL's "sequence" perscribes a particular 'model', involving both genetic (QTL) and non-genetic terms, which the program will use in subsequent analyses to attempt to explain the selected trait data. MAPMAKER/QTL's "map" and "scan" commands examine this model in light of the observed genetic and phenotypic data , and (1) first fill in the unset parameters of the model with their maximally likely values, and then (2) determine the likelihood that the (filled in) model explains the observed data. While making it easy to perform the most common and relatively simple QTL mapping analyses, more sophisticated uses of the "sequence" command can specify models allowing very complex types of analyses to also.

With regard to the genetic component of the model (e.g. the putative QTLs), the "sequence" also specifies which interval(s), or combinations of intervals, are to be searched for QTLs. Other constraints on the QTLs, including the exact location of the putative QTL(s) and the mode of inheretence they exhibit may also be specified. Many such examples of uses of the "sequence" command exist in the MAPMAKER/QTL tutorial.

To set the MAPMAKER/QTL sequence, simply type "sequence xxxx", where xxxx is a valid MAPMAKER/QTL sequence as described below. Typing the "sequence" command alone causes the current setting of the sequence to be displayed. Here, we provide a brief summary of the syntax used by MAPMAKER/QTL's "sequence" command:

* Possible QTL locations may be specified in a number of ways: (1) You may simply give the left locus name or number, indicating the interval between that locus and the next consecutive locus in the genetic map. In this case, as no precise location is given, the interval will be searched to find the most likely QTL position. (2) You may give a range of intervals (for example, "1-4") specifying all intervals in the map falling between the two loci. For example, the sequence "1-4" specifies the three intervals numbered "1", "2", and "3", assuming that the genetic order of the markers is "1 2 3 4". Note that by 'between' we are refering to order of the markers in the linkage map, not necessarily their numeric order. Intervals between chromosomes are not searched. (3) You may indicate an interval as two loci separated by a vertical bar, for example "1|4". Unlike the previous case, this specifies the

single interval with marker 1 on the left and marker 4 on the right. Thus, with the genetic order "1 2 3 4", data for markers 2 and 3 will be completely ignored.

* You may precisely fix the position of a QTL within an interval by adding the notation "+ distance",where distance is the map distance between the left marker of the interval and the QTL (thus, distance must be less than the total length of the interval). If distance is less than 0.50, it is assumed to be a recombination fraction, while if it is greater than or equal to 0.50, then it is assumed to be in centimorgans. The fixed-position notation may not be used with locus ranges, listed under (2) above. Note also that this notation may not be used in the interval to be seached with the "scan" command (always the rightmost interval).

* For each QTL to be allowed, the possible locations must be enclosed within square brackets, e.g. "[...]". Any number of possibile locations for each QTL may be given, separated by spaces, and each using the syntax described above. In general, each such location will be tried one at a time. Thus, the sequence:

```
[1-4 9]
```

is functionally equivalent to the sequence:

```
[1 2 3 9]
```

where both specify that four intervals are to be tried, each separately.

Multiple QTL models may be specified by simply listing the possible locations for each QTL in separate pairs of brackets (e.g. '[1][9]'). If multiple possible locations are provided for 2 or more separate QTL(s), then all possible combinations will be tried. For example, the sequence:

```
[1 2 3][12+10 25|27+15]
```

specifies the six combinations:

```
[1][12+10]    [1][25|27+15]
[2][12+10]    [2][25|27+15]
[3][12+10]    [3][25|27+15]
```

* You may try n-tuples of QTL possibilities by preceeding the brackets listing the possible locations with "number of ", where number may range from 2 to 7. Of course, number must be less than the number of possible QTL locations. For example, the sequence:

```
2 of [12+10 25|27+15 41+0]
```

specifies the three pairwise combinations:

```
[12+10][25|27+15]
[12+10][41+0]
[25|27+15][41+0]
```

* There are other even more sophisticated models which we can specify with the "sequence" command, as well as other related features. These include:
    - Models which use one trait to predict another, with or without additional QTLs.
    - Models of F2 data which constrain the modes of inheritence of the QTLs.
    - The ability to store and name sequences for convenience.

These topics are discussed in the following sections. See also the section on the "Names Facility".

Explaining Traits with Other Traits

In addition to adding QTLs to the sequence using square brackets, you may specify models which use a trait, possibly in addition to QTLs, to explain the data for another trait. As an example, if we had a hypothetical data set where:

```
    Map and Genotype Data: From F2 Progeny
             f2trait data: Measured from F2 Progeny
             f3trait data: Mean trait measured from F3
poulation from each F2 individual
```

Then we might wish to perform an analysis with the sequence:

```
    {f2trait}[12+5]
```

indicating that we want to see how well the F2 trait values plus a QTL located 5 cM distal of marker 12 predict the F3 trait data. In this case, our model for the mean phenotype of each F3 population derived from F2 individual number i would be:

```
    F2Traiti = Mean + Noise + (Weight 4 Numi) + (Dominance 4 Heti) +
                (Const 4 F3Traiti)
```

where:

```
    Mean = the mean value of the component of the trait not controlled
           by these two QTLs (in effect, the average trait value for
           individuals A/A at both QTLs).
    Noise = variation in the trait not controlled by either QTL
            (a normal random variable)
    Weight = the additive component of the QTL's B allele effect
    Numi = the number of B alleles caried by individual number i, at the QTL
    Dominance = the dominance component of the QTL's B allele effect
    Heti = 1 if individual number i is an A/B heterozygote at the QTL,
           0 otherwise
    Const = The effect of the contribution of the F2 trait on the F3 trait
```

You should compare this expression with that given on page *** of the tutorial, showing the model for a single F2 QTL alone. If we set this trait and sequence, and thenissued the "map" command, MAPMAKER/QTL would procede to determine the maximally likely values of the Weight, Dominance, and Const terms. To determine if these effects are significant, we might compare the likelihood of this map with the maps for the two sequences:

```
    {f2trait}
    and [12+5]
```

Specifying Modes of QTL Inheretence for F2 Data

With F2 data, you may specify an assumed mode of inheretence for the QTL's alleles within the brackets by preceeding the mode with a colon. Allowed modes of inheretence are "free", "additive", "dominant", and "recessive", all defined with respect to the B parent's allele. For example, you may type the two-QTL sequence:

```
    [12+10:additive][25|27+14:dominant]
```

This would correspond to a model where the phenotype of each individual i is given by:

```
    Traiti = Mean + Noise + (Weight1 4 Num1i) + (Dominance2 4 Het2i) +
             (Weight2 4 Num2i) + (Dominance2 4 Het2i)
```

where:

```
    Mean = the mean value of the component of the trait not controlled
```

by these two QTLs (in effect, the average trait value for
individuals A/A at both QTLs).

Weightn = the additive component of the nth QTL's B allele effect
Numni = the number of B alleles caried by individual number i, at the nth QTL
Dominancen = the dominance component of the nth QTL's B allele effect
Hetni = 1 if individual number i is an A/B heterozygote at the nth QTL, 0 otherwise
Noise = variation in the trait not controlled by either QTL (a normal random variable)

and where the constraints on the modes of inheretence imply that:

```
Dominance1 = 0.0
Dominance2 = -Weight2
```

In addition, you may specify the mode of inheretence as "try", indicating that each of the four modes (free, dominant, additive, and recessive) should be tried separately. Only one mode of inheretence may be specified for each QTL in the sequence, and the mode will apply to all possible QTL locations to be tried. In other words, the following is an illegal sequence:

```
[1+12:additive 35+20:dominant] ,   Wrong!
```

HISTORY Command (abbreviation 'h')

Summary: List Previous Sequences
Argument: <number of previous sequences to display>
Default: 20

MAPMAKER/QTL allows you to easily recall sequences previously entered. These sequences are remembered and are collectively referred to as the 'sequence history'. In MAPMAKER/QTL's "sequence" command, the syntax "#n", where n is a number, instructs MAPMAKER/QTL to recall and insert the n-th sequence used. To list the sequence history, simply type "history".

Sequences from the MAPMAKER/QTL sequence history may be used as pieces of new commands by including the history reference(s) with other sequence elements.

LIST LOCI Command

Summary: Display Trait and Locus Names and Numbers
No Arguments

The "list loci" command displays a table of all locus numbers and their corresponding names, followed by a similar chart of all trait numbers and their respective names. These names and numbers are interchangable in nearly all MAPMAKER/QTL commands.

EDIT SEQUENCE Command

Summary: Interactive Keyboard Driven Sequence Editor
No Arguments

The "edit" command allows the user to employ standard editing functions to change the current sequence. It is only available on those systems which support the Curses screen control package. If this function is available, typing "edit" will display the current sequence and let you change it by using arrow keys, control keys, the backspace/delete key, and by typing new text into it. When the sequence is to your liking, hit RETURN and it will become the current sequence.

LET Command

Summary: Name a Sequence or Portion of a Sequence
Arguments: <name> = <sequence>

The "let" command assigns a name to any valid MAPMAKER/QTL sequence. The sequence may any allowed types of brackets, distances, locus names or numbers, as well as other names set using the "let" command. To refer to a previously named sequence, you simply use the assigned sequence name. Names must start with an alphabetic character, are limited to 10 characters, and may not conflict with any locus names listed in the data set. For example:

```
3> let new7q = 14 12 6

4> sequence  [new7q]
The current sequence is now '[14 12 6]'.
```

Note that the matching of sequence names is not sensitive to alphabetic case (e.g., "sevenq" will match "SevenQ"). Moreover, only enough characters of a name need be given in order to specify that name unambiguously. (For example, the construct "pdg" will match the sequence named "PDGregion" so long as no other locus or sequence name begins with the letters "pdg").

MAPMAKER automatically assigns the name "all" upon loading data to refer to a sequence which lists all of the loci in the loaded data set in the order in which they appear in the linkage groups in the data file. In addition, names of any saved chromosomes from MAPMAKER Version II or from the raw data file are also loaded into the names table.

NAMES Command

Summary: Display Values of All Names Set Using 'Let'
No Arguments

The "names" command displays a list of all of the sequence names which have been set. For example:

```
1> let new7q = 14 12 6

2> names
chrom1=     2 5 9 3 4 7 8 1 11
chrom2=     10 14 12 6 13 15
all=        chrom1 chrom2
new7q=      14 12 6
```

FORGET NAMED SEQUENCE Command

Summary: Forget the Assignment of a Name (Undo 'Let')
Argument: <name>

This command instructs MAPMAKER/QTL to remove a name from the current list of names remembered. For example,

```
1> let psar1= 1 2 3
psar1 = 1 2 3

2> let psar2= 4 5 6
```

```
psar2 = 4 5 6

3> forget psar1

4> names
psar2 = 4 5 6
```

(4) TRAIT COMMAND AND RELATED FEATURES

Before using most of MAPMAKER/QTL's analysis functions, you must also specify the particular trait data to be used in the analysis. MAPMAKER/QTL's trait facility enables you to easily select and display mathematical characteristics of your trait data, as well as create and store new traits as functions of one or more already existing traits. Commands related to the use of the trait facility include:

'Trait' Command
'Make Trait' Command
'Show Trait' Command
'List Traits' Command
'Forget Trait' Command

TRAIT Command (abbreviation 't')

Summary: Set the Trait for QTL Mapping
Argument: <trait name or number>
Default: with no arguments, displays the current trait

The "trait" command is used to select which trait data in the data set are to be used in subsequent analyses. Traits may be specified by name or by number, and may include either trait data included in the original raw data file, or transformed traits which have been generated with the "make trait" command. Only one trait may be selected at a time. For example:

```
1> trait weight
The trait is now: 1 (weight)
```

Typing the "trait" command with no arguments displays the trait currently selected.

MAKE TRAIT Command

Summary: Add a New Trait (as a Function of Existing Traits)
Arguments: <name> = <equation>

The "make trait" command is used to creat a new trait as a function of one or more traits already existing in the data set. The trait will then be added to the data set, so that it can be used for later analysis. For example:

```
1> make trait logwt = log(weight)
```

would create a trait named logwt, which would be the base 10 log of the trait weight. The command would also print out important statistical values and a histogram for the new trait. The make trait supports several mathematical functions including addition, subtraction, multiplication, division, exponent notation, log, ln, sine, arcsine, cosine, arccosine, tangent and arctangent.

SHOW TRAIT Command

Summary: Display Statistics and Histogram for a Given Trait
Argument: <trait>
Default: the current trait

The "show trait" command prints several pieces of statistical data and a histogram for a given trait. Values displayed include the mean, standard deviation, kurtosis, skewness, quartile ratio, and a percentage of values with one-quarter, one-half, one, two, and three standard deviations of the mean. Typing the "show trait" command with no arguments displays the values for the trait currently selected.

LIST TRAITS Command

Summary: Display a List of Traits Currently in Data Set
No Arguments

The "list traits" function presents a table, similar to that produced by the "list loci" command, listing all traits in the data set by number and name, including those traits created using the "make traits" command.

FORGET TRAIT Command

Summary: Remove a Trait from the Data Set
Argument: <trait>

As its name indicates, the "forget trait" command removes a trait from the data set. When given a trait as an argument, "forget trait" searches for a trait of that name and number and, upon finding it, removes it from the data set. This removal will be made permanent if the data set is rewritten using the "save status" command, or when exiting with the "auto save data" option "on".

(5) MAPMAKER/QTL MAPPING COMMANDS

MAPMAKER/QTL provides two primary commands for calculating QTL maps based on the sequence (model) and trait currently selected by the user: the "map" and "scan" commands. A variety of related features are also provided to make the processing and visualizing of these results easier. Commands related to QTL mapping include:

- ’Map’ Command
- ’Scan’ Command and the Related Commands:
    - ’List Scans’, ’Show Scan’, ’Draw Scan’, ’Forget Scan’, ’Forget All Scans’, ’Show Peaks’, and ’Show Trys’
- ’Compare’ Command and the Related Commands:
    - ’List Compares’, ’Show Compare’, ’Forget Compare’, ’Forget All Compares’, and ’Show Best Maps’
- ’Show Linkage Maps’ Command

MAP Command (abbreviation ’m’)

Summary: Display QTL-Maps for Specified Intervals
No Arguments

The "map" command is used to calculate a QTL map or maps, describing the ability of the model specified by the current sequence to explain the selected trait data. The model may may contain both genetic (including one or many QTLs) and/or non-genetic components (e.g. the value of one trait used to explain another), with the remaining variance (that not explained by the genetic and non-genetic portions of the model) attributed to normal random noise. If the current sequence lists multiple possibile locations for any QTL(s), then multiple QTL maps are calculated, one for each of the possible locations or combination of locations. Each QTL map may involve some limited searching for the most likely QTL position(s) between flanking markers, although more elaborate genome searches generally require the "scan" command instead. Many examples of the usage of the "map" command are provided in the MAPMAKER/QTL tutorial. Below, we provide a more technical discussion of the "map" command’s use.

The models used by MAPMAKER/QTL must be additive, and are of the form:

```
    Phenoi = Mean + Effect1i + ... + Effectni + Noise
```

where:

```
    Phenoi = The measured trait value for individual number i.
    Mean = The mean value of the component of the trait not controlled
           by either the genetic or non-genetic components of the model.
           Effectively, this is the "base trait value" for individuals A/A
           at each QTL and with no contribution from other traits.
    Effectji = As described below, the j-th term describing the effect on
               the phenotype of individual number i, determined from individual
               i’s genotypes at QTLs, individual i’s other traits, and constant
               parameters.
    Noise = variation in the trait not controlled by this QTL (a normal
            random variable with mean 0.0 and standard deviation s2).
```

An effect term for a QTL (here, QTL number j) contributing to Phenoi is of the form:

```
    for BC1 data: Effectji = (Weightj 4 Numij)
    for F2 data:  Effectji = (Weightj 4 Numij) + (Dominancej 4 Hetij)
```

where:

```
    Weightj = The additive component of the B allele effect for QTL number j.
    Numij = The number of B alleles caried by individual number i at QTL
```

number j. This is 0 or 1 for BC1 data, and 0, 1, or 2 for F2 data.
Dominancej = The dominance component of the B allele effect for QTL number j.
Hetij = 1 if individual number i is an A/B heterozygote at QTL number j, 0 otherwise.

An effect term for a non-genetic contribution to Phenoi (here, a contibution from another measured trait, number k) is of the form:

    Effectki = (Weightk 4 Traitik)

where:

    Weightk = The amount of the contribution from trait number k on the phenotype
    Traitik = The measured value of trait k for individual i, included in
              the data set.

When calculating a QTL map, MAPMAKER/QTL first fits the model to the observed data, determining the maximally likely values of the parameters Weightj, Dominancej, Weightk, Mean, and sigma. Then, MAPMAKER/QTL determines the overall likelihood that this model, with the parameters filled in, explains the observed phenotypic data. The log10 of this likelihood, along with all of the mentioned parameters are displayed. In addition, the corresponding c2 value and the fraction of the total variance explained by the model (excluding noise) is printed. The user can compare the log-likelihoods of various QTL maps to each other and to the appropriate log-likelihood threshold to declare QTLs. Note that MAPMAKER/QTL displays the QTL position in either centimorgans or as a recombination fraction, depending on the setting of the "units" parameter. Similarly, genetic markers will be displayed with their names or their numbers, depending on the "print names" option.

For example:

```
16> trait 2
The current trait is now: 2 (logwt)

17> sequence [6+6.1 2+11.0]
The interval-list is now ’[6+6.1 2+11.0]’

18> map
==============================================================
QTL map for trait 2 (logwt):

INTERVAL   LENGTH  QTL-POS  WEIGHT  DOMINANCE
6-17       11.9    6.1     -0.08   -0.04

chi%2= 22.893 (2 D.F.)         log-likelihood= 4.97
mean= 0.805   sigma%2= 0.052   variance-explained= 8.4 %
==============================================================
QTL map for trait 2 (logwt):

INTERVAL   LENGTH  QTL-POS  WEIGHT  DOMINANCE
2-7        36.7    11.0    -0.15   -0.00

chi%2= 40.801 (2 D.F.)         log-likelihood= 8.86
mean= 0.873   sigma%2= 0.046   variance-explained= 19.0%
==============================================================
```

Here, we have used the "sequence" command to specify a simple one QTL model for a trait in an F2 data set:

    Phenoi = Mean + (Weight1 4 Numi1) + (Dominance1 4 Heti1) + Noise

In addition, we have supplied two possible (and precise) locations for the QTL. For each of the two locations, MAPMAKER/QTL has calculated the values of the parameters in the model, as well as the log-likelihood of these two QTL maps.

Compare this result to:

```
17> sequence [6+6.1][2+11.0]
The interval-list is now '[6+6.1][2+11.0]'

18> map
=============================================================
QTL map for trait 2 (logwt):

INTERVAL   LENGTH  QTL-POS  WEIGHT  DOMINANCE
6-17       11.9    6.6      -0.07   -0.05
2-7        36.7    10.4     -0.15   -0.00

chi%2= 54.832 (4 D.F.)         log-likelihood= 12.52
mean= 0.855   sigma%2= 0.030   variance-explained= 26.6%
=============================================================
```

where we have applied a model allowing two QTLs to explain the trait data:

```
Phenoi = Mean + (Weight1 4 Numi1) + (Dominance1 4 Heti1) +
         (Weight2 4 Numi2) + (Dominance2 4 Heti2) + Noise
```

Here, MAPMAKER/QTL has again estimated the values of the parameters describing both QTLs, although this time assuming that they are acting together to control the trait, rather than separately.

SCAN Command

Summary: Search for QTLs, Stepping Through the Right Interval
Arguments: <cM-step> <LOD-threshold> <scale>
Defaults: 1.0 2.0 0.25

The "scan" command provides a powerful method for searching a genome (or regions of a genome) for QTLs. The method employed actually calculates many QTL maps, each corresponding to a particular point in the genome. For each such map, MAPMAKER/QTL determines the maximally likely QTL map parameters and likelihood (just as discussed for the "map" command) assuming that a putative QTL is located precisely at that point. MAPMAKER/QTL displays the resulting likelihood surface as a graph of log-likelihood vs. position. This graph provides a simple visual representation of the results, and allows one to easily pick out likelihood peaks corresponding to the likely locations of putative QTLs.

The "scan" command is most often used to simply search all intervals indicated by the current sequence (assuming it allows only one QTL) for unconstrained QTLs controlling the specified trait alone. For example:

```
7> trait 2
The current trait is now: 2 (logwt)

8> sequence [1-9]
The interval-list is '[1-9]'

9> scan

QTL maps for trait 2 (logwt):
Sequence: [1-9]
LOD threshold: 2.00  Scale: 0.25 per '*'

No fixed-QTLs.
Scanned QTL genetics are unconstrained.

POS     WEIGHT  DOM     %VAR  LOG-LIKE |
---------------------------------------| 1-3 4.8 cM
0.0    -0.102  -0.007   9.0%    5.663  | ***************
```

```
    2.0    -0.110  -0.008  10.4%    6.178  | ****************
    4.0    -0.116  -0.008  11.4%    6.605  | ******************
    -------------------------------------| 3-2 6.3 cM
    0.0    -0.109  -0.010   9.9%    6.778  | *******************
    2.0    -0.118  -0.012  11.5%    7.469  | *********************
                     .
                     .
                     .
```

The "scan" command can take up to three arguments:
1. The spacing between data points visited in the genome. If greater than or equal to 0.50, this number is assumed to be a centimorgan map distance, otherwise it is assumed to be a recombination fraction. The default is 2.0 cM.
2. The minimum log-likelihood required before MAPMAKER/QTL displays any asterisk ("*") characters next to the map data. The default is 2.0.
3. The log-likelihood increase required for each additional asterisk. The default is 0.25.

We can also employ the "scan" command to perform more elaborate searches. For example, assuming that we have already found one very likely QTL, we we can "fix" this QTL in place and then scan the genome, searching for other QTLs, which, together with the first one, provide likely explanations of the data. In this case, the increased sensitivity (from controlling some of the variation in the population with the first QTL) may help you detect new QTL's which would not otherwise have met your required likelihood threshold. Also, this same method may be used to determine whether multiple neighboring likelihood surface peaks indicate one or many putative QTLs. Note that while we fix the position of the first QTL, we re-estimate the effects of it each time we calculate a new QTL map, insuring that the estimated parameteters are in fact the correct maxima, and that, should two QTLs control the same portion of the variation in the trait, then that QTL which provides the most likely explanation of the data will be assigned the bulk of the effect.

To use the "scan" command for such analyses, you must observe a few simple rules:

* In the model specified by the current sequence, the locations to be 'scanned' must be those indicated as possibilities for the rightmost QTL. For example, to fix one dominant QTL 12 cM distal of marker 27, and then scan the entire genome for additional QTLs, we would use the sequence:

    [27+12:dominant][all]

    not the sequence:

    [all][27+12:dominant]

* Because of the nature of the computation performed by the "scan" command on the rightmost QTL, it would not make sense to use a fixed QTL position for any of the possible locations of that QTL. Thus, the sequence should not use the "+" notation in the rightmost QTL (for example: the incorrect sequence above would produce an error because of this). For similar reasons, the rightmost term in the sequence must specify a QTL and not a non-genetic effect (e.g. another trait variable using the "{...}" notation).

* If the leftmost QTLs (and non-genetic components) in the sequence specify more than one possibility (e.g. a QTL with multiple possible locations, or, with F2 data, a QTL using ":try" to test multiple genetic models), then each possible leftmost combination will be used separately, each generating a unique scan of the rightmost QTL. For example, using the sequence:

    [1+10 9+0 24+4.8 50+0][all]

would result in four separate scans of the entire genome ("all" intervals), one each fixing a QTL near markers 1, 9, 24, and 50, respectively.

Compare this to the sequence:

    [1+1][9+0][24+4.8][50+0][all]

which would result in one scan of the genome with four fixed QTLs.

Similarly, with F2 data, if you specify ":try" genetics for the rightmost QTL, then one scan will be produced for each of the four possible genetic models.

* Lastly, it is wise to list the intervals for the rightmost QTL in their genetic order. The "scan" command will examine the intervals in the order specified, and out of order results will be difficult to interpret (although completely correct).

As a fairly complex example, we could search for QTLs on chromosomes one and six, fixing two QTLs one at a time and adding a non-genetic component to explain the data with the sequence:

    {predictor}[12 37][chrom1 chrom6]

Note that neither of the two fixed QTLs in this example have their precise locations specified, and thus MAPMAKER/QTL will place them in their most likely positions within the specified interval each time it calculates a map for a new point in the genome. These positions may not be the same for different data points, but because of the generally weak position resolution in QTL mapping, this allows the model to be adjusted slightly to best explain the data for each point.

LIST SCANS Command

Summary: List Scan Results Which Have Been Saved
Argument: <scan number>

Each time you execute the "scan" command, MAPMAKER/QTL saves the resulting data in the computer’s memory. (In fact, these data may be written back out into your data files when you quit MAPMAKER/QTL, and will be reloaded when the program is restarted, depending on the setting of the "auto save" option). You may list all "scan" results which have been saved in this manner by typing the "list scans" command. For example:

```
10> list scans
Saved scan results:

NUM   TRAIT   SEQUENCE
1.1     2     [all]
2.x     2     [8-9:try]
```

Each scan command is represented by a decimal number of the form n.m. The left portion of this number (n) indicates that data are from the n-th "scan" command executed. If the sequence used with scan number n actually instructed MAPMAKER/QTL to generate multiple scans, then each such scan result is assigned a unique number m. In such cases, MAPMAKER/QTL lists the scan results as n.x, indicating that multiple results have ben saved. Otherwise, n.1 is displayed.

Thus, the sequence:

    [1 25][all]

if used in the third scan command, would generate two complete genome scans, numbered 3.1 (fixing a QTL in interval 1) and 3.2 (fixing a QTL in interval 25). Similarly, the sequence:

    [chrom7:try]

used with the next "scan" command would generate four scans of chromosome 7, numbered 4.1 through 4.4. Scan 4.1 corresponds to an unconstrained (ordinary) search, scan 4.2 is a search with the QTL forced to exhibit dominant genetics, scan 4.3 forces the QTL to exhibit recessive genetics, and scan 4.4 forces the QTL to exhibit additive genetics. In these cases, you can list the individual scans calculated by typing "list scan number", where number is the appropriate value of n, as described above. For example:

```
10> list scan 2

Scan results numbers 2.1-2.4 for trait 2 (logwt).
Sequence: [8-9:try]

NUM   GENETICS  FIXED-QTLS
2.1   free      none
2.2   dominant  none
2.3   recessive none
2.4   additive  none
```

SHOW SCAN Command

Summary: Display Saved Scan Data for a Particular Scan
Arguments: <scan-number> <LOD-threshold> <scale>
Defaults: <last-scan> 2.0 0.25

The "show scan" re-displays the scan data for one scan result saved in MAPMAKER/QTLs memory (as described for the "list scans" command). For example, to display the mth scan result generated by the nth use of the "scan" command, simply type "show scan n.m", with n and m filed in appropriately. If the nth use of the scan command generated only one result, you may instead type "show scan n", where MAPMAKER/QTL assumes that you mean scan n.1.

The "show scan" command may take two additional arguments, specifying (1) The minimum log-likelihood required before MAPMAKER/QTL displays any asterisk ("*") characters next to the map data; and (2) The log-likelihood increase required for each additional asterisk. The defaults are 2.0 and 0.25, respectively.

DRAW SCAN Command

Summary: Draws Saved Scan Data in PostScript
Argument: <scan-number> <LOD-threshold>
Default: <last-scan> 2.0

The "draw scan" command draws the scan data for one scan result saved in MAPMAKER/QTLs memory (as described for the "list scans" command) as a PostScript graphic file. For example, to draw the mth scan result generated by the nth use of the "scan" command, simply type "draw scan n.m", with n and m filed in appropriately. If the nth use of the scan command generated only one result, you may instead type "draw scan n", where MAPMAKER/QTL assumes that you mean scan n.1.

The scan will be drawn one chromosome per page as a likelihood graph with increasing log-likelihood running up the page and the markers of the chromosome itself listed along the x-axis from left to right. A LOD threshold may be specified on the command line which will draw a dotted line at the given threshold to make identification of regions with greater than a certain log-likelihood much easier.

FORGET SCAN Command

Summary: Delete Particular Scan Results from Memory
Argument: <scan-number>

The numerous "scan" results saved by MAPMAKER/QTL can be quite large and may consume vast amounts of memory (and data file space) for longer analyses. You can delete particular "scan" results and reclaim this memory by typing "forget scan n", where n indicates the results of the n-th scan command executed (as described for the "list scans" command). By default, the last scan is deleted.

Similarly, the "forget all scans" command deletes all scan results which MAPMAKER/QTL has saved.

FORGET ALL SCANS Command

Summary: Delete All Scan Results from Memory
No Arguments

The numerous "scan" results saved by MAPMAKER/QTL can be quite large and may consume vast amounts of memory (and data file space) for longer analyses. You can delete all saved "scan" results and reclaim this memory by typing "forget all scans". Similarly, the "forget scan" command deletes a particular scan result which MAPMAKER/QTL has saved.

SHOW PEAKS Command

Summary: Display LOD Peaks and QTL Maps for a Particular Scan
Arguments: <scan-number> <LOD-threshold> <falloff>
Defaults: <last-scan> 2.0 -1.0

The "show peaks" command can be used to sutomatically find the likelihood surface peaks of a saved scan result (the process of saving scan results is described under the "list scans" command). As arguments, you need to specify:

1. The scan results number, in the same manner as for the "show scan" command. With no arguments, the "show peaks" command examines the last "scan" data generated.
2. The minimum log-likelihood threshold required to declare a peak.
3. The log-likelihood decrease from the maximum needed to declare the confidence interval around the peak.
4. The log-likelihood decrease from the maximum needed to declare separate peaks, as opposed to local maxima of the same peak.

The default values for the last three arguments are 2.0, 1.0, and 2.0, respectively. For example:

```
10> show peaks

LOD score peaks for scan 1.1 of trait 2 (logwt).
Sequence: [chrom1]
No fixed-QTLs.
Scanned QTL genetics are unconstrained.
Peak Threshold: 2.00  C.I. Falloff: 1.00  Peak Falloff: 2.00

==============================================================
QTL-Map for peak 1:
Confidence Interval:   Left Boundary= 2|3 + 4.0
                      Right Boundary= 3|4 + 20.0

INTERVAL   LENGTH  QTL-POS  GENETICS    WEIGHT  DOMINANCE
2|3        36.7    10.0     free       -0.15   -0.00
```

```
chi%2= 40.783 (2 D.F.)         log-likelihood= 8.86
mu= 0.872    sigma%2= 0.046    variance-explained= 18.6%
==============================================================
```

SHOW TRYS Command

Summary: Display LOD Scores for a Scan Using ':try' Genetics
Arguments: <scan-number> <LOD-threshold>
Defaults: <last-scan> 2.0

The "show trys" command is used in the same manner as the "show scans" and "show peaks" commands to display the results of previously executed "scan" commands. The "show trys" command however only applies to scans where ":try" was specified for the genetics of the scanned (rightmost) QTL. If this is the case, the "show trys" command displays the log-likelihood of the free genetics map, as well as the log-likelihood decrease from this of the dominant, recessive, and additive maps.

As arguments, you may specify:

1. The scan results number, in the same manner as for the "show scan" command. With no arguments, the "show trys" command examines the last "scan" data generated.
2. The minimum log-likelihood threshold required to declare a significant result, which will be flagged with an asterisk ("*") character. The default is 2.0.

For example:

```
13> show trys 2.1 8.0

Test genetics results for trait 2 (logwt).
Sequence: [3-7:try]
No fixed-QTLs.
Scan numbers: 2.1-2.4  Threshold: 8.00
```

GENETICS:	FREE				DOMINANT		RECESSIVE		ADDITIVE		
POS	WEIGHT	DOM	%VAR	LOG LIKE	%VAR	LIKE DIFF	%VAR	LIKE DIFF	%VAR	LIKE DIFF	
interval= 2|3 length= 6.3 cM											
0.0	-0.109	-0.010	9.9	6.78	7.4	-1.76	5.5	-3.15	9.9	-0.03	
2.0	-0.118	-0.012	11.5	7.47	8.6	-1.96	6.6	-3.38	11.5	-0.04	
4.0	-0.122	-0.015	12.1	7.87	9.0	-2.11	6.8	-3.60	12.0	-0.06	
6.0	-0.122	-0.017	11.9	8.01	8.8	-2.22	6.5	-3.74	11.7	-0.09	*
interval= 3|4 length= 20.7 cM											
0.0	-0.122	-0.017	11.8	8.01	8.7	-2.23	6.5	-3.75	11.7	-0.09	*
2.0	-0.129	-0.015	13.3	8.26	10.1	-2.25	8.1	-3.61	13.3	-0.06	*
4.0	-0.135	-0.013	14.9	8.48	11.4	-2.28	9.9	-3.44	14.9	-0.04	*
6.0	-0.141	-0.009	16.3	8.66	12.5	-2.32	11.7	-3.27	16.3	-0.02	*
8.0	-0.146	-0.006	17.6	8.79	13.4	-2.37	13.3	-3.09	17.6	0.00	*
10.0	-0.150	-0.002	18.6	8.86	14.1	-2.41	14.6	-2.92	18.6	0.00	*
12.0	-0.152	0.000	19.3	8.84	14.5	-2.44	15.6	-2.76	19.3	0.00	*
14.0	-0.153	0.002	19.7	8.75	14.7	-2.44	16.1	-2.61	19.7	0.00	*
16.0	-0.153	0.004	19.7	8.57	14.7	-2.41	16.3	-2.47	19.6	0.00	*
18.0	-0.151	0.005	19.2	8.31	14.5	-2.35	16.0	-2.33	19.2	-0.01	*
20.0	-0.148	0.005	18.5	7.97	14.0	-2.26	15.4	-2.20	18.4	-0.01	

COMPARE Command

Summary: Compare QTL-Maps for Specified Interval Combinations
No Arguments

The "compare" command is used to calculate QTL maps for all locations specified by the current sequence and compare their likelihoods. As with the "map" command, some searching is done for the maximally likely QTL position within a given interval. Instead of printing out full maps for each location, the compare command merely displays a short summary of the map results for each map. For example:

```
2> seq [1 3 2 7]

3> compare

     INTERVAL : %VAR  LOG-LIKE
#1   1-3        10.7  6.23
#2   3-2        11.4  7.73
#3   2-7        20.3  8.97
#4   7-8        4.6   3.00
```

indicates that a QTL is more likely to be found in interval 3 than in any of the other intervals. All compare results are saved in the ".qtls" file and can be viewed using "show compare", "list compares", and "show best maps".

LIST COMPARES Command

Summary: List Compare Results Which Have Been Saved
No Arguments

MAPMAKER/QTL saves the results of each "compare" command in the computer's memory. (In fact, these data may be written back out into your data files when you quit MAPMAKER/QTL, and will be reloaded when the program is restarted, depending on the setting of the "auto save" option). Typing "list compares" lists all "compare" results which have been saved in this manner.

SHOW COMPARE Command

Summary: Display Saved Data from a Particular 'Compare'
Arguments: <compare-number> <LOD-threshold> <falloff>
Defaults: <last-compare> 2.0 -1.0

Like the "show scan" command, displays the result of a previously executed and stored "compare" command.

FORGET COMPARE Command

Summary: Delete Particular Compare Results from Memory
Argument: <compare-number>

Deletes the "compare" result indicated by the numeric argument and reclaims the memory it was using. Use "list compares" to view the numbered list of stored "compare" results.

FORGET ALL COMPARES Command

Summary: Delete All Compare Results from Memory
No Arguments

Deletes and reclaims the memory used by all previously stored results of the "compare" command.

SHOW BEST MAPS Command

Summary: Display Best Results from a Particular 'Compare'
Arguments: <compare-number> <LOD-threshold> <falloff>
Defaults: <last-compare> 2.0 -1.0

The "show best maps" command displays the best QTL maps calculated in the indicated "compare". The maps displayed will be the map with the highest log-likelihood (if it is greater than the given LOD-threshold) and all those with log-likelihoods within the indicated falloff value.

SHOW LINKAGE MAP Command

Summary: Display Linkage Map Locus Order and Distances
Argument: <locus name or number, or 'let' name>
Default: Displays linkage map for all chromosomes

The "show linkage map" command displays the linkage map order and distances for all genetic markers listed in the current sequence. This information is obtained from saved linkage maps which were loaded into MAPMAKER/QTL during the "load data" command.

(6) MAPMAKER/QTL PARAMETERS AND OPTIONS

Here we desrcibe other MAPMAKER/QTL options and parameters not described elsewhere.

PRINT NAMES Option

Summary: Set Displaying of Locus Names or Numbers
Argument: <'on' or 'off'>
Default: with no arguments, displays the current value

The "print names" option, when "on", instructs MAPMAKER/QTL to display locus and trait names in its output instead of numbers.

UNITS Parameter

Summary: Display Recombination Fractions or cM Distances
Argument: <'cm' or 'rf'>
Default: with no arguments, displays the current value

The "units" parameter is used to instruct MAPMAKER/QTL as to what units should be used when displaying recombination distances. Two choices are available:

```
    Recombination Fractions (0.00...0.50)
and Centimorgan Distances (0.50...999.0)
```

For example, to use centimorgan distances, you would type

```
1> units centimorgans
The 'units' are now set to haldane centimorgans.
```

Most of MAPMAKER/QTL's commands are able to alter their output based on the setting of this parameter. The function for converting recombination fractions to centimorgan distances is selected by "centimorgan function" parameter. When MAPMAKER/QTL begins, map distances are displayed as recombination fractions. Note that when asking for a map distance as a command argument or as input to a prompt, MAPMAKER/QTL usually assumes that values greater than or equal to 0.50 are centimorgan distances, while values less than 0.50 are recombination fractions.

AUTO SAVE DATA Option

Summary: Force MAPMAKER/QTL to Save Data When You Quit
Argument: <'on' or 'off'>
Default: with no arguments, displays the current value

MAPMAKER/QTL saves the status of many of its current settings, as well as any computed "scan" and "compare" results, into your data files when you quit the program. This behavior is controlled by the "auto save data" option, and is intended to easily let you quit and restart MAPMAKER/QTL, resuming your analysis right where you left off. By default, this option is set to "on", although you may turn it off by typing:

```
1> auto save data off
```

Items which are saved include:

* most option and parameter values
* the command and sequence histories
* the list of saved sequence names (created by "let" and other commands)

* all "scan" and "compare" calculations
* the saved chromosome maps

Because of the quantity of data written out, the saving process can take quite some time on slower computers, and thus you may wish to disable it after loading.

MORE MODE Option

Summary: Enable Pauses After Each Screen of Output
Argument: <'on' or 'off'>
Default: with no arguments, displays the current value

The "more mode" option, when "on", causes MAPMAKER/QTL to pause in between screenfuls of text, prompting you to type a key before it continues displaying output. Even when set to "on", this option applies only to certain commands which perform no computationally difficult analyses but often generate large amounts of output. In addition, "more mode" is enabled only if you are using MAPMAKER/QTL interactively. "More mode" is "off" by default.

(7) MISCELLANEOUS COMMANDS

Here we describe other MAPMAKER/QTL commands not described elsewhere.

RUN Command (abbreviation 'r')

Summary: Accept Commands from an Input File
Argument: <file-name>

The "run" command instructs MAPMAKER/QTL to take a series of commands from a file. Such a file may be created using any ordinary text editor, and should contain lines of commands and other input to MAPMAKER/QTL just as they would be typed into MAPMAKER/QTL interactively. Some word-processing programs will work for this purpose, so long as they have a mode for saving "ASCII text only" (otherwise, they insert formatting commands into the file which will confuse MAPMAKER/QTL).

For example, a "run" file to place a locus in a particular known order might read

```
load data cross1.data
photo place3.out
trait 3
sequence [2 5 4 7 9 6]
scan
quit
```

and could be run with the command

```
1> run place3.in
```

Unless the command file contains a "quit" command, MAPMAKER/QTL returns to reading user input after the commands in the file have been executed. (The "quit" command acts in a special way in command files, in that no "yes" response is required afterwards). By default, MAPMAKER/QTL assumes that the file name ends with the ".in" extension, although you may explicitly specify a different one. File names specifying directories (as supported on the system) may be used as follows:

```
    1> run /users/joe/work/place3.in        (on a UNIX system)
or  1> run [joe.work]place3.in              (on a VMS system)
```

CHANGE DIRECTORY Command (abbreviation 'cd')

Summary: Changes Default Directory Used by MAPMAKER/QTL
Argument: <directory-name>
Default: with no arguments, display the current default directory

The 'cd' command works essentially the same way it does under DOS or Unix. By default, all files are read or written from thie directory unless specified otherwise. However, under DOS, it is currently NOT possible to change the current default drive (only the directory). We recommend that you only use files on your primary hard disk (usually C:) and that you are sure this disk is selected (displayed in your prompt) before you run MAPMAKER/QTL.

SYSTEM Command (abbreviation '!')

Summary: Run a system command, or go to a system prompt
Argument: <command>

The "system" command is used to temporarily interrupt MAPMAKER/QTL and start up a new command interpreter from the operating system. Commands which are normally typed to the operating system may then be issued. This now works on most Unix and VAX/VMS systems. You can usually return to MAPMAKER/QTL by typing either control-D or "exit" (Unix), or "logout" (VMS). For example, on a Unix system:

```
3> system

$ ls
mapmaker       oldrflps.data   newrflps.data
june20.out     june21.out      place3.out

$ lp june20.out
file 'june20.out' will be printed on device "lab laser writer"

$ exit

Back in MAPMAKER/QTL

4>
```

If you wish to execute just one operating system command, that command may be given as an argument to "system". For instance, typing "system ls" will, on a UNIX-based system, produce a listing of all file names in your current working directory. This feature, however, does not work on VAX/VMS systems, and only works for simple commands (technically speaking: only commands which work from the Bourne shell without reading your ".profile" configuration file).

Alternatively, MAPMAKER/QTL works in a text based terminal emulator window within most windowing systems, including X-Windows (aka DECWindows) and SunView. Thus, you may alternatively open a new window to run other programs or commands while you use MAPMAKER/QTL. MAPMAKER/QTL does not correctly handle resizing of its window while it is running, however.

As yet another alternative, MAPMAKER/QTL may be stopped and restarted using Unix job control features (e.g., from the C-shell, type control-Z).

PREVIOUS COMMANDS Command

Summary: Display previous MAPMAKER/QTL commands
Argument: <number of commands to display>
Default: <all previous commands>

Typing "previous commands" instructs MAPMAKER/QTL to display a list of previous commands entered in the current session of MAPMAKER/QTL. To reenter a command without explicitly retyping it, simply type the number of the command you wish to enter. For example:

```
4> previous commands
Previous commands:
1  load data cross1
2  seq [1 2 3]
3  seq [4 5 6]

5> 2
=> seq [1 2 3]
The interval list is '[1 2 3]'
```

REVIEW OUTPUT Command

```
Summary:     Display previous MAPMAKER/QTL output
No Arguments
```

This command causes the last 100 lines of MAPMAKER/QTL output to be redisplayed on the screen, with MAPMAKER/QTL pausing in between each screenful. This is useful for reviewing the results of analyses which have since scrolled off the top of the screen (if you are using a terminal or terminal-emulator which does not provide a "scrollback" feature).

TIME Command

```
Summary:     Display current time (useful for timing commands)
No Arguments
```

The "time" command reports the current date and time, which may be useful for timing MAPMAKER functions. For example:

```
    3> time
    The current time is:  Fri Oct 2 18:47:08 1987
```

COMMENT Command

```
Summary:     Enter a comment into the photo file (does nothing)
Argument:    <comment>
Default:     with no args, allows you to enter a long comment
```

The "comment" command allows one to type comments into MAPMAKER/QTL which will be recorded in the output transcript file (if "photo"-ing is "on"). "Comment" allows both single and multiple line comments in the following manner:

For a single line comment, type the comment as an argument:

```
    3> comment  Now we will examine trait 16

    4>
```

For a multiple line comment, give no arguments:

```
    4> comment
    Enter your comment. End it with a period ('.') on a line by itself.

    We will now examine the scan results for trait 5
    to see if we can find a likely position for a qtl.
    .

    5>
```

WIZARD MODE Command

```
Summary:     Danger Will Robinson, Danger!
Argument:    <on or off>
```

The "wizard mode" enables MAPMAKER/QTL features which are only useful for debugging or which have been deliberately disabled. Steer clear.

MAPMAKER/QTL 1.1B MANUAL COMMAND REFERENCE:

MAPMAKER/QTL 1.1B MANUAL QUICK REFERENCE:

comment...................Enter a comment into the photo file (does nothing)
wizard mode...............Danger Will Robinson, Danger!

* = reference information only - not a command