Whitehead Institute

Constructing Genetic Linkage Maps with MAPMAKER/EXP Version 3.0: A Tutorial and Reference Manual

Stephen E. Lincoln, Mark J. Daly, and Eric S. Lander

A Whitehead Institute for Biomedical
Research Technical Report
Third Edition, January, 1993
(Beta Distribution 3B)

Please address all correspondence to:

MAPMAKER	Internet:	mapmaker@genome.wi.mit.edu
c/o Dr. Eric Lander	Bitnet:	mapm@mitwibr
Whitehead Institute	FAX:	617-258-6505
9 Cambridge Center		
Cambridge, MA 02142 USA		

Constructing Genetic Linkage Maps with MAPMAKER: A Tutorial and Reference Manual

Table of Contents:

Page 2: T.O.C. 1

Page 3: T.O.C. 2

Page 4: QuickRef 1

Page 5: QuickRef 2

Starting MAPMAKER/EXP

In this tutorial we will illustrate the use of many of MAPMAKER's features while analyzing a real data set. We have provided these data sets with the MAPMAKER software, and we strongly encourage you to follow along with this tutorial on your computer.

When MAPMAKER starts running, you will first see its start-up banner and a prompt "`1>`" for the first command. Precisely how you should start MAPMAKER depends on your computer. These issues are described in detail in the "Installation Guide" included with this manual.

Note that throughout this tutorial, we indicate command you type into MAPMAKER in ***bold italics***, while MAPMAKER output is presented in `regular type`. Throughout this manual, we use the names MAPMAKER and MAPMAKER/QTL interchangeably.

The first step in almost every MAPMAKER session is to load a data file for analysis. If you are starting out an analysis on a new data set, or if you have modified the raw data in an existing data set, you will do this using MAPMAKER's "prepare data" command, as described in the "Data Preparation Guide" , included with MAPMAKER. If instead you are resuming an analysis of a particular (unmodified) data set, you may use the "load data" command, which preserves many of the results from your previous session. Because we are just starting out, we use MAPMAKER's "prepare data" command to load our sample file "sample.raw".

From this file MAPMAKER extracts:

The type of cross, number of markers, and number of scored progeny
The genotype for each marker in each individual (if available)

Other information may be present in the data files, such as quantitative trait data and pre-computed linkage results. These issues will be addressed later.

Before performing any analyses of our data set, we now instruct MAPMAKER to save a transcript of this session in a text file for later reference. Using the "photo" command, we start a transcript named "tutorial.out". Note that if the file already exists, MAPMAKER appends new output to this file.

```
**********************************************************************
* Output from:                                                       *
*                                                                    *
*                            MAPMAKER/EXP                            *
*                          (version 3.0b)                            *
*                                                                    *
**********************************************************************

Type 'help' for help. Type 'about' for general information.
```

1> ***prepare data sample.raw***

```
preparing data from 'sample.raw'...
  F2 intercross data  (333 individuals, 12 loci)... ok
saving genotype data in file 'sample.data'... ok
```

2> ***photo tutorial.out***

```
'photo' is on: file is 'tutorial.out'
```

Finding Linkage Groups by Two-Point Linkage

We begin this session by performing a classical "two-point" , or pairwise, linkage analysis of our data set. While we generally do *not* use two-point analysis for *ordering* markers, we usually do find two-point analysis helpful for *identifying* linkage groups of markers in preliminary analyses. To then order markers within a group, we use other more powerful techniques.

First, we need to tell MAPMAKER which loci we wish to consider in our two-point analysis. We do this using MAPMAKER's "sequence" command. When you type something like:

    sequence locus1 locus2 locus3 ...

MAPMAKER is told which loci (and, in some cases, which *orders* of those loci) any following analysis commands should consider (for ex: locus1, locus2, locus3, etc.). Since almost all of MAPMAKER's analysis functions use the "current sequence" to indicate which loci they should consider, you will find that the "sequence" command must be entered before performing almost any analysis function. The sequence of loci in use remains unchanged until you again type the "sequence" command to change it.

In this two point analysis we want to examine all 12 of the loci in our sample data set. Thus, we now type into MAPMAKER:

    sequence 1 2 3 4 5 6 7 8 9 10 11 12

Note that for two-point analysis, the order in which the loci are listed is unimportant.

We then type MAPMAKER's "group" command, instructing the program to divide the markers in the sequence into linkage groups. To determine whether any two markers are linked, MAPMAKER calculates the maximum-likelihood distance and corresponding LOD score between the two markers: If the LOD score is greater than some threshold, and if the distance is less than some other threshold, then the markers will be considered *linked*. By default, the LOD threshold is 3.0, and the distance threshold is 80 Haldane cM.

For the purpose of finding linkage groups, MAPMAKER considers linkage *transitive*. That is, if marker A is linked to marker B, and if B is linked to C, then A, B, and C will be included in the same linkage group.

As you see, MAPMAKER has divided our data set into two linkage groups, which it names "group1" and "group2". Moreover, there are no unlinked markers in this data set.

Note to users of previous versions of MAPMAKER: It is no longer necessary to use the "two point" command first: MAPMAKER computes two-point data automatically as needed. See the reference manual for details

```
3>  sequence  1  2  3  4  5  6  7  8  9  10  11  12
sequence #1= 1 2 3 4 5 6 7 8 9 10 11 12
```

```
4>  group
Linkage Groups at min LOD 3.00, max Distance 50.0

group1= 1 2 3 5 7
-------
group2= 4 6 8 9 10 11 12
```

Exploring Map Orders by Hand

To determine the most likely order of markers within a linkage group, we could imagine using the following simple procedure: For each possible order of that group, we calculate the maximum-likelihood map (e.g. the distances between all markers given the data), and the corresponding map's likelihood. We then compare these likelihoods and choose the most likely order as the answer. This type of exhaustive analysis may be performed using MAPMAKER's "compare" command.

In practice however, this sort of "exhaustive" analysis is not practical for even medium sized groups: a group of N markers has N!/2 possible orders, a number which becomes unwieldy (for most computers) when N gets to be between 6 and 10. (In practice, one needs to order subsets of the linkage group and then overlap those subsets, mapping any remaining markers relative to those already mapped, a process we will illustrate later).

Luckily, group1, consisting of markers 1, 2, 3, 5, and 7 is small enough that we can use a fully exhaustive analysis.

To do this, we first change MAPMAKER's sequence to "{1 2 3 5 7}". Here, the set-braces indicate that the order of the markers contained within them is unknown, and thus that all possible orders need to be considered.

We then type the "compare" command, instructing MAPMAKER to compute the maximum likelihood map for each specified order of markers, and to report the orders sorted by the likelihoods of their maps. Note that while MAPMAKER examines all *** orders, only the 20 most likely ones are reported (by default).

For each of these 20 orders, MAPMAKER displays the *log-likelihood* of that order relative to the best likelihood found. Thus the best order:

1 3 2 5 7

is indicated as having a *relative log-likelihood* of 0.0. The second best order:

3 1 2 5 7

is significantly less likely than the best, having a relative log-likelihood of -6.0. Said a different way, the best order of this group is supported by an odds ratio of roughly 1,000,000:1 (10 to the 6th power to one), over any other order. We consider this good evidence that we have found the right order.

```
5> sequence {1 2 3 5 7}
sequence #2= {1 2 3 5 7}
```

```
6> compare

Best 20 orders:
1:    1 3 2 5 7   Like:  0.00
2:    3 1 2 5 7   Like: -6.00
3:    5 7 2 3 1   Like: -20.20
4:    5 7 2 1 3   Like: -26.26
5:    2 5 7 3 1   Like: -27.25
6:    2 5 7 1 3   Like: -28.39
7:    2 3 1 5 7   Like: -28.85
8:    5 2 3 1 7   Like: -32.33
9:    2 1 3 5 7   Like: -34.12
10:   5 7 1 3 2   Like: -35.55
11:   5 2 1 3 7   Like: -37.61
12:   1 3 5 2 7   Like: -37.76
13:   3 1 5 2 7   Like: -39.09
14:   5 7 3 1 2   Like: -40.38
15:   1 3 5 7 2   Like: -40.87
16:   3 1 5 7 2   Like: -41.55
17:   5 2 7 3 1   Like: -43.67
18:   5 2 7 1 3   Like: -44.78
19:   5 1 3 2 7   Like: -47.63
20:   2 5 3 1 7   Like: -52.28
order1 is set
```

Displaying a Genetic Map

When we used the "compare" command previously, MAPMAKER calculated the map distances and log-likelihood for each of the 60 orders we were considering. The "compare" command however only reports the relative log-likelihoods, and afterwards forgets the map distances. To actually display the genetic distances we must instead use the "map" command.

Like "compare", the "map" command instructs MAPMAKER to calculate the maximum likelihood map of each order specified by the current sequence. If the current sequence specifies more than one order (for example, the sequence "{1 2 3 5 7}" specifies 60 orders) then the maps for all specified orders will be calculated and displayed.

Because we found one order of this group to be much more likely than any other, we probably only care to see the map distances for this single order. First, we set MAPMAKER's sequence, putting the markers in their best order and doing away with the set brackets.

Next, we simply type "map" to display this order's maximum likelihood map.

As you can see, the distances between neighboring markers are displayed. Note however, that these distances may be considerably different than the "two-point" distances between those markers: This is because MAPMAKER's so-called *multipoint analysis* facility can take into account much more information, such as flanking marker genotypes and some amount of missing data. This is precisely the reason that we use multipoint analysis rather than two point analysis to order markers: Because more data is taken into account, you have a smaller chance of making a mistake.

```
7> sequence 1 3 2 5 7
sequence #3= 1 3 2 5 7
```

```
8> map
======================================================================
Map:
  Markers          Distance
    1  T175          4.2 cM
    3  C35          15.0 cM
    2  T93          11.9 cM
    5  C66          12.2 cM
    7  T50B      ----------
                    43.2 cM   5 markers   log-likelihood= -424.94
=====================================================================
```

Mapping a Slightly Larger Group

As we mentioned earlier, exhaustive analyses of large linkage groups are not practical. Instead, to find a map order of a larger group, we need to find a subset of markers on which we can perform an exhaustive "compare" analysis. Thus, to map group2 we could pick a subset of its 8 markers at random, although we might do better if we pick markers which are likely to be ordered with high likelihood. Generally, this is true for sets of markers which have (i) as little missing data as possible, and (ii) do not have many closely spaced markers.

To quickly see how much data is available for the markers in this group, we set MAPMAKER's "sequence" appropriately and use MAPMAKER's "list loci" command.

MAPMAKER prints a list of loci, showing each marker by both its MAPMAKER-assigned number as well as it's name in the data file. For each marker, MAPMAKER prints the number of informative progeny (out of the 333 in the data set), and the type of scoring. In this case all loci have been scored using "co-dominant" markers (e.g. RFLP-like genotypes in an F2 intercross), although clearly markers 4 and 6 are the least informative.

To also look for markers which may be too close, we use MAPMAKER's "lod table" command.

MAPMAKER prints both the distance and LOD score between all pairs of markers in the current sequence. Unfortunately, the closest pair is separated by over 6.0 cM, a distance which should almost always be resolvable in a data set with so many informative meioses.

Given the results of these two analyses, a good subset to try might be:

8 9 10 11 12

Note that the above two tests could have been automatically performed using MAPMAKER's "suggest subset" command, documented in the reference section.

```
9> sequence 4 6 8 9 10 11 12
sequence #4= 4 6 8 9 10 11 12


10> list loci

                                Linkage
 Num  Name       Genotypes     Group
   4  T24        273 codom     group2
   6  T209       275 codom     group2
   8  T125       306 codom     group2
   9  T83        327 codom     group2
  10  T17        297 codom     group2
  11  C15        324 codom     group2
  12  T71        319 codom     group2


11> lod table
Bottom number is LOD score, top number is centimorgan distance:

      4     6     8     9     10    11

6     63.1
      3.33

8     16.8  56.0
     39.06  4.33

9     56.3  17.8  54.8
      6.77 36.70  7.68

10   106.3  27.7   -    43.3
      0.89 22.51       15.08

11    14.9  74.0   6.3  65.4   -
     43.78  2.20 80.87  5.76

12    28.2  43.1  18.4  24.1  89.1  30.1
     22.24  9.13 39.84 32.39  2.22 23.90
```

As before, we now change MAPMAKER's sequence to specify the subset we wish to test, and then type the "compare" command. This time, the results are even more conclusive, with one order at least 10 to the 14.6*th* more likely than any other.

We also again select the best order as the current sequence using MAPMAKER's "sequence" command. This time however, we so this using a special shortcut, "order1", which is a name MAPMAKER often sets to indicate the results of its analyses.

To determine the map position of the remaining two markers in group2, we will use the following procedure: Starting with the known order of 5 markers, we will place the other two (one at a time) into every interval in this order and then recalculate the maximum-likelihood map of each resulting 6 marker order. In this analysis, MAPMAKER recalculates *all* recombination fractions for *all* intervals in each map (not just the ones involving the newly placed markers).

This function is performed by MAPMAKER's "try" command. In its output, MAPMAKER again displays relative log-likelihood of each position for the inserted markers. The relative log-likelihood of 0 indicates the best position, while the negative log-likelihoods indicate the odd against placement in each other interval.

In this case, we see that marker 6 strongly prefers to be in-between markers 9 and 10. Even the next most likely position for marker 6 is more than 10 to the 21.09*th* power times less likely.

The "try" command not only tries to place markers in each interval in the framework, but also tries to place each marker infinitely far away (that is, forced 50% recombination between it and the framework). The relative log-likelihoods for this position are indicated following the "INF" entry in the MAPMAKER output. In the same way that a two-point LOD score indicates the odds of linkage between two loci when they are separated by their maximum likelihood distance, these relative log-likelihoods indicate the odds supporting linkage between one locus and a framework of loci when the locus is placed in its most likely position. In the above test, we see that a log-likelihood of 44.66 supports linkage between 4 and the rest of the group.

```
12> sequence {8 9 10 11 12}
sequence #5= {8 9 10 11 12}

13> compare

Best 20 orders:
1:    11 8 12 9 10   Like:  0.00
2:    10 11 8 12 9   Like: -14.57
3:    8 11 12 9 10   Like: -15.23
4:    10 9 11 8 12   Like: -27.20
5:    11 8 12 10 9   Like: -29.97
6:    10 8 11 12 9   Like: -30.14
7:    9 10 11 8 12   Like: -32.23
8:    8 11 10 9 12   Like: -39.80
9:    10 9 8 11 12   Like: -39.91
10:   9 11 8 12 10   Like: -40.05
11:   11 8 10 9 12   Like: -40.25
12:   11 8 9 12 10   Like: -44.73
13:   8 11 12 10 9   Like: -45.21
14:   10 11 8 9 12   Like: -46.57
15:   8 11 9 12 10   Like: -47.46
16:   9 10 8 11 12   Like: -47.94
17:   10 8 11 9 12   Like: -49.61
18:   8 11 10 12 9   Like: -52.71
19:   9 8 11 12 10   Like: -52.74
20:   11 8 10 12 9   Like: -53.07
order1 is set
```

```
14> sequence order1
sequence #6= order1

15> try 4 6

         4       6
       ---------------
      |  0.00  -42.68 |
11    |               |
      |-35.57  -118.6 |
8     |               |
      |-19.65  -70.19 |
12    |               |
      |-46.80  -28.09 |
9     |               |
      |-51.35    0.00 |
10    |               |
      |-43.40  -21.09 |
      |---------------|
INF   |-44.66  -45.03 |
       ---------------
BEST  -619.33 -612.03
```

As a last step, we now type the complete sequence for this group, adding markers 4 and 6 into their most likely positions. Then we type "map" to see the complete map of all markers in this group.

```
16> sequence 4 11 8 12 9 6 10
sequence #7= 4 11 8 12 9 6 10

17> map
========================================================================
Map:
  Markers          Distance
    4  T24          14.8 cM
   11  C15           6.4 cM
    8  T125         18.9 cM
   12  T71          24.0 cM
    9  T83          18.1 cM
    6  T209         28.6 cM
   10  T17        ----------
                   110.8 cM   7 markers   log-likelihood= -688.99
========================================================================
```

Dealing with Larger Data Sets

At this point in our tutorial, we now discuss the issues presented when analyzing larger data sets. As our first step, we load a new, and much larger data set: "mouse.raw". This file describes 308 markers scored on 46 progeny of an F2 intercross between the Mouse strains *Mus castaneous* with *Mus musculus* (C57/BL-6). These markers span the 19 mouse autosomes at an average density of about one every 5 cM. Note that our previous data set is saved when we load a new one.

Upon loading the data, you see that MAPMAKER begins executing a long stream of commands. This is not entirely automatic behavior, in the sense that we explicitly told MAPMAKER to do this by supplying an "initialization file", named (in this case) "mouse.prep". (If MAPMAKER does *not* now start doing this, you must be missing this file. Unfortunately, you need it to proceed with this tutorial). In this initialization file we execute many commands to help configure MAPMAKER to this particular data set, including:

- We define the number of mouse chromosomes and tell MAPMAKER their names.
- We "anchor" a few physically localized markers to these chromosomes, so that MAPMAKER can assign linkage groups to chromosomes.
- We tell MAPMAKER to pre-compute all two-point data, results we will make use of in further analyses.
- We define what criteria define a "highly-informative" marker.
- We set various other user-selectable preferences.

These steps can take anywhere from 5-30 minutes, depending on your computer. To learn how the commands in our initialization file work, you can look the commands up in the reference section following this tutorial. In addition, these issues are discussed in detail in the section on "Preparing data", included with this manual.

One remark we will make here is that, for the rest of this tutorial, we will be working with marker names rather than MAPMAKER-assigned numbers (that is, our initialization file contained the command "print names on"). Our naming convention for the mouse data set is that all markers are given 4 character names, starting with a letter then three digits. This format is *not* a requirement of MAPMAKER however.

```
18> prepare data mouse.raw
save loaded data first?  yes

saving genotype data in file 'sample.data'... ok
saving map data in file 'sample.maps'... ok
saving two-point data in file 'sample.2pt'... ok
saving traits data in file 'sample.traits'... ok

preparing data from file 'mouse.raw'... ok
  F2 intercross data  (46 individuals, 308 loci)... ok
running initialization commands from file 'mouse.prep'...


19> units cm
the 'units' are now set to (Haldane) centimorgans.


20> centimorgan func haldane
centimorgan function: Haldane


21> print names on
'print names' is on.


22> triple error detection on
'triple error detection' is on.


23> informativeness criteria 4.0 44 codominant
Informativeness Criteria: min Distance 4.0, min #Individuals 44
(codominant markers only)


24> make chromosome chrom1 chrom2 chrom3 chrom4 chrom5 chrom6
chromosomes defined: chrom1 chrom2 chrom3 chrom4 chrom5 chrom6


25> make chromosome chrom7 chrom8 chrom9 chrom10 chrom11
chromosomes defined: chrom1 chrom2 chrom3 chrom4 chrom5 chrom6 chrom7
chrom8 chrom9 chrom10 chrom11


26> make chromosome chrom12 chrom13 chrom14 chrom15 chrom16
chromosomes defined: chrom1 chrom2 chrom3 chrom4 chrom5 chrom6 chrom7
chrom8 chrom9 chrom10 chrom11 chrom12 chrom13 chrom14 chrom15 chrom16


27> make chromosome chrom17 chrom18 chrom19
chromosomes defined: chrom1 chrom2 chrom3 chrom4 chrom5 chrom6 chrom7
chrom8 chrom9 chrom10 chrom11 chrom12 chrom13 chrom14 chrom15 chrom16
chrom17 chrom18 chrom19

       .
       .                                        « More Output Follows
       .
```

Finding Chromosome Assignments

As before, we begin this session by performing a two-point linkage analysis of our entire data set in order to find groups. However, for large data sets spanning a genome, we prefer not to work with anonymous *linkage groups*, but instead we will work with entire mouse *chromosomes*.

To group large numbers of markers spanning a genome, we need to develop a slightly different notion of linkage. For the mouse genome, a LOD score of 3.0 represents about a 1 in 20 significance level, meaning that in the roughly 50,000 pairwise comparisons ($308^2/2$), we will falsely declare over 2,000 linkages! Because of transitivity, as we discussed earlier, many chromosomes may become linked into any one linkage group.

The solution is not necessarily to blindly raise the LOD threshold: to raise it high enough will exclude a large number of true linkages. For this reason we will not use MAPMAKER's "group" command, but instead we use the "assign" command. Here's how it works:

In the initialization file, certain markers have been pre-assigned to chromosomes: this will determine which linkage groups belong on which chromosomes. (These are the so called "anchor" loci). To assign the rest of the markers to chromosomes we:

(1) Select a strict LOD threshold.
(2) Look for linkage between any unassigned marker and an assigned marker. If the strength of evidence for assignment is good, assign the markers in question to their respective chromosomes.
(3) Slightly relax the LOD threshold and repeat.

The algorithm is in fact slightly more complex than this, in order to deal with conflicting data well (cases where a marker shows linkage to more than one chromosome). See the description in the reference section for details.

To assign all markers in the data set to chromosomes, we first set MAPMAKER's sequence. As we do not wish to type in the number or name of each locus, we will use the special abbreviation "all". Next we simply type "assign".

As you see, MAPMAKER's "assign" command produces quite a bit of output, indicating the strength of assignment for each of the markers in the data set. Thankfully, MAPMAKER provides efficient ways of viewing these results.

```
55> sequence  all
sequence #21= all
```

```
56> assign
L016     - anchor locus on chrom12...cannot re-assign
L018     - anchor locus on chrom2...cannot re-assign
L019     - anchor locus on chrom9...cannot re-assign
L020     - anchor locus on chrom1...cannot re-assign
  .
  .                                          « More  Output  Follows
  .
L036     - assigned to chrom17 at LOD 10.4
L046     - assigned to chrom1 at LOD  3.6
L010     - assigned to chrom15 at LOD 18.1
D004     - assigned to chrom9 at LOD 13.1
L007     - assigned to chrom16 at LOD 21.1
D016     - assigned to chrom17 at LOD  9.9
D021     - assigned to chrom17 at LOD 21.1
D017     - assigned to chrom15 at LOD  3.1
  .
  .                                          « More  Output  Follows
  .
M251     - unassigned
A066     - unassigned
```

One way to summarize the chromosome assignments is to use MAPMAKER's "list chromosomes" command. This simply prints a summary of the total number of markers assigned to each chromosome (in the first column) as well as other information for analyses we have not yet performed.

As you can see in the "Total" line, not all markers were assigned (that is, only 305 of the 308 were assigned). In fact, if you review the "assign" command's output, you will see that markers M228, M251 and A066 remain unassigned. Your computer may or may not provide a convenient way to scroll backwards through the output (PC's in particular lack this ability). If this is the case, you may try using the "review output" command, described in the reference section.

As a different way to review a summary of these results, we may use the "list status" command. To see the information for only the three unassigned markers, we first change the sequence, using another convenient "sequence" abbreviation: "unassigned".

As you can see, *conflicting data* exist for locus M228 -- it shows evidence of linkage to two different chromosomes. This may indicate a possible problem in the raw data, or it may be the result of statistical fluctuation: After performing almost 50,000 pairwise tests, it is not unlikely to see a spurious linkage at LOD 3.8. The other two markers, M251 and A066, showed no conclusive evidence of linkage.

To provide more information, we can also easily generate a list of all loci linked to the three unassigned markers. For this, we use MAPMAKER's "links" command. At a LOD threshold of 3.0 (the default) we see the basis to the conflicting linkage data for M228 -- It shows evidence of linkage to markers on both chromosomes 3 and 14 (although certainly the evidence for linkage to chromosome 14 is much stronger than the evidence of linkage to chromosome 3 (highest LOD of 11.75 vs. 3.58). This suggests that the chromosome 3 linkage may in fact be spurious -- if we check the data and come to believe the chromosome 14 assignment, we could then simply assign M228 to chromosome 14 using MAPMAKER's "attach" command.

By contrast, marker A066 appears truly unlinked, although this could in principle be due to true recombinational distance or massive data errors. Marker M251 shows weak evidence of linkage to chromosome 13 -- it may in fact be quite distant, although this too would be a good marker to double check in the lab.

57> ***list chromosomes***

```
 Chromosome:  #Anchors  #Total
   chrom1         2       18
   chrom2         2       30
   chrom3         2       21
   chrom4         2       19
   chrom5         2       12
   chrom6         2       19
   chrom7         2       20
   chrom8         2       14
   chrom9         2       22
   chrom10        2       15
   chrom11        2       15
   chrom12        2       12
   chrom13        2       11
   chrom14        2        9
   chrom15        2       18
   chrom16        2        7
   chrom17        2       22
   chrom18        2       16
   chrom19        2        5
   Total:        38      305
```

58> ***sequence unassigned***

sequence #22= unassigned

59> ***list status***

```
 Num  Name    Assignment   Chrom   LOD
 181  M228    conflict!      -      -
 196  M251    unlinked       -      -
 274  A066    unlinked       -      -
```

60> ***links***

Linked Markers in Data Set at min LOD 3.00, max Distance 50.0

```
 Marker-1   Marker-2   Theta    LOD     cM
 M228       L038        0.18    3.58   23.00   (chrom3)
 M228       T010        0.16    5.55   19.13   (chrom14)
 M228       M032        0.05   11.75    5.28   (chrom14)
 M228       M214        0.08    9.70    8.28   (chrom14)
 M228       A024        0.13    6.51   15.22   (chrom14)
 M228       A034        0.20    3.37   24.73   (chrom3)
 M228       A060        0.17    3.81   21.27   (chrom3)
 M228       A103        0.17    5.13   21.20   (chrom14)
 M228       A119        0.14    6.43   16.01   (chrom14)
 M251       no linked markers
 A066       no linked markers
```

Automatically Finding a Map Orders

Having divided our data set into chromosomes, we can now determine the likely map for each of those chromosomes. However, because of the quantity of data to manage, we will not try to do this using the manual mapping commands presented earlier (such as "try" and "compare"). As you will see, these functions may be performed in a more automated fashion using simple MAPMAKER commands.

First, we use the "sequence" command to select a particular chromosome for analysis. While again we could type the names of all the loci on that chromosome, MAPMAKER allows us to type the chromosome name alone as a convenient shortcut. For illustrative purposes, we select the set of markers assigned to chromosome 10.

Next, we use the "three point" command to pre-compute the likelihoods of all three-point crosses for this chromosome We do *not* have to do this step to proceed, although three-point analysis provides a powerful way to speed up the steps we perform below. Here's how it works:

We would like our mapping analyses to consider as many orders of any group as possible. However, most of these orders are very unlikely, given the observed data, and are simply not worth wasting computer time on. In MAPMAKER/EXP 3.0, three-point analysis simply excludes the majority of these "ridiculous" orders from consideration, allowing MAPMAKER to spend time carefully examining only those orders reasonably consistent with the observed data.

When you type the "three-point" command, MAPMAKER first finds every linked triple of markers in the current sequence. For each triple, MAPMAKER computes the most likely map distances and likelihoods for all 3 possible orders. For each order, MAPMAKER displays the 'relative log-likelihood' of that order as compared to the most likely (or best) order of the triple. As before, the most likely order of the three is has a relative log-likelihood of 0.0, while the others have negative relative log-likelihoods.

MAPMAKER will make use of these data as follows: any three-point order will be considered *excluded* if its relative log-likelihood is worse than the best by some threshold (by default, the threshold is 4.0). Any multiple locus order which contains one or more excluded three-point sub-orders will itself be considered excluded, and only non-excluded multipoint orders will be evaluated by full multipoint analysis.

Had we not performed this step, then MAPMAKER would use full-multipoint analysis in the steps below to evaluate all possible orders. This definitely would be slower, but presumably would produce identical answers.

```
61> sequence chrom10
sequence #23= chrom10: assigned

62> three point
Linkage Groups at min LOD 3.00, max Distance 50.0
Triplet criteria: LOD 3.00, Max-Dist 50.0, #Linkages 2
'triple error detection' is on.
counting...143 linked triplets in 1 linkage group

                                     log-likelihood differences
 count  markers                        a-b-c  b-a-c  a-c-b
    1:  L062 D030 M067                  0.00  -8.93  -0.58
    2:  L062 D030 M003                 -8.63   0.00  -0.74
    3:  L062 D030 M007                 -5.00   0.00  -4.11
    4:  L062 D030 M172                 -4.19   0.00  -4.75
    5:  L062 D030 M139                  0.00  -4.32  -1.36
    6:  L062 D030 M153                  0.00  -1.24  -3.25
    7:  L062 D030 A037                 -5.91   0.00  -3.33
    8:  L062 D030 A114                  0.00  -4.62  -0.55
    9:  L062 D030 T032                 -3.00   0.00  -5.78
   10:  L062 T031 M139                 -1.89  -4.33   0.00
   11:  L062 T031 A114                 -2.59  -3.75   0.00
   12:  L062 M067 M003                 -9.36   0.00  -0.70
   13:  L062 M067 M007                 -5.45   0.00  -3.77
   14:  L062 M067 M172                 -4.56   0.00  -4.33
   15:  L062 M067 M139                  0.00  -7.01  -0.37
   16:  L062 M067 M153                  0.00  -1.59  -2.43
   17:  L062 M067 A037                 -6.44   0.00  -3.08
   18:  L062 M067 A114                 -0.30  -7.97   0.00
   19:  L062 M067 T032                 -3.18   0.00  -5.21
   20:  L062 M003 M007                  0.00  -0.80  -6.08
   21:  L062 M003 M172                  0.00  -0.77  -7.77
   22:  L062 M003 M139                 -0.62   0.00 -10.45
   23:  L062 M003 A037                  0.00  -0.83  -3.53
   24:  L062 M003 A114                 -0.66   0.00  -9.68
   25:  L062 M003 A063                  0.00  -0.56 -11.32
   26:  L062 M003 T032                  0.00  -0.69 -10.00
   27:  L062 M007 M175                  0.00  -2.63  -5.85
   28:  L062 M007 M172                  0.00  -8.74  -0.75
   29:  L062 M007 M139                 -2.49   0.00  -5.77
   30:  L062 M007 A037                 -0.78  -6.98   0.00
   31:  L062 M007 A114                 -2.77   0.00  -5.34
   32:  L062 M007 A063                  0.00  -5.33  -4.36
   33:  L062 M007 T032                  0.00  -6.86  -2.92
   34:  L062 M175 M172                 -4.35  -3.65   0.00
   35:  L062 M175 T032                 -2.53  -5.42   0.00
   36:  L062 M172 M139                 -3.00   0.00  -4.83
   37:  L062 M172 A037                 -2.42  -6.05   0.00
   38:  L062 M172 A114                 -3.36   0.00  -4.48
   39:  L062 M172 A063                  0.00  -7.05  -2.93
   40:  L062 M172 T032                  0.00  -9.23  -1.46
             .
             .                                 « More  Output  Follows
             .
```

We now use MAPMAKER's "order" command to find a linear order of the markers on chromosome 10. Briefly, this command performs the following analyses: (1) it tries to find a small subset of loci (by default, 5 loci), for which a single order is found to be much more likely than any other using a "compare" style analysis; (2) remaining markers which can be mapped to a unique position are added to this order one at a time; (3) at the end, any markers which cannot be mapped to a *unique* position in the order are mapped into multiple intervals. A more detailed explanation is provided in the reference section.

We have a few hints on the use of the "orders" command, and three-point analysis more generally. First, three-point analysis is not nearly as complete as multipoint analysis, and it is a mistake to rely on it for all map ordering. We suggest that you use three-point analysis only with very conservative thresholds, as described in the reference manual. This way, MAPMAKER will default to using its more powerful multipoint analysis mechanism to resolve any questionable cases.

Secondly, three-point analysis is somewhat more sensitive to genotyping errors than full multipoint analysis (because the bulk of the raw data is not available to indicate the correct crossover positions). Not only does this mean that conservative thresholds are in order, but also that using three-point with "error detection" is wise in dense data sets which may have some small number of genotyping errors present (again see the reference manual for details). We selected this option in our initialization file using the "triple error detection" command.

Lastly, the ability of the "orders" command to find a starting order depends heavily on its ability to find a good highly informative subset. Thus, it is imperative that you use the "informativeness criteria" command to set good criteria for your data set (because the criteria depend on the nature of your data set, the default criteria are extremely lax.) We also did this in our initialization file -- please see the reference manual for details.

Because the "order" command uses some randomized sorting, your results may disagree slightly with those shown here (in fact, if you run the command twice, you may see slightly different output). Of course, you should not see dramatic *qualitative* differences in the MAPMAKER analyses (or let's say that if you do, this likely indicates contradictory data in the raw data set, perhaps the result of genotyping errors).

The output from the "order" command can be quite lengthy, essentially because of the multiple automatic steps executed by this command. We explain this output on the next page.

63> ***order***

```
Linkage Groups at min LOD 3.00, max Distance 50.0
Starting Orders: Size 5, Log-Likelihood 3.00, Searching up to 50 subsets
Informativeness: min #Individuals 44 (codominant only), min Distance 4.0
Placement Threshold-1 3.00, Threshold-2 2.00, Npt-Window 7

============================================================================
Linkage group 1, 15 Markers:
  56 L062        63 D030         81 T031        109 M067        122 M003
 123 M007       159 M175        164 M172        197 M024        212 M139
 222 M153       263 A037        277 A114        278 A063        301 T032

Most informative subset: 278 222 109 122 212 159 164
Searching for a unique starting order containing 5 of 7 informative loci...
Got one at log-likelihood 13.19

Placing at log-likelihood threshold 3.00...
Start:   109 122 164 278 159
Npt:     109 122 (263) 164 278 159
Npt:     109 (56) 122 263 164 278 159
Npt:     109 56 122 263 (123) 164 278 159
Npt:     109 56 122 263 123 164 (301) 278 159
No unique placements for 6 remaining markers

Map:
  Markers          Distance
  109  M067         18.5 cM
   56  L062          2.4 cM
  122  M003          6.0 cM
  263  A037          2.3 cM
  123  M007          2.3 cM
  164  M172          3.5 cM
  301  T032          4.7 cM
  278  A063          8.9 cM
  159  M175       ----------
                    48.4 cM   9 markers   log-likelihood= -75.40

Markers placed relative to above map:
         109  56   122  263  123  164  301  278  159
          :-19-:--2-:--6-:--2-:--2-:--3-:--5-:--9-:
 277 2 **.:..*.:....:....:....:....:....:....:...
 212 2 **.:..*.:....:....:....:....:....:....:...
  63 2 .*.:.**.:....:....:....:....:....:....:...
 222 2 **.:..*.:....:....:....:....:....:....:...
  81 2 **.:....:....:....:....:....:....:....:.*.
 197 3 **.:..*.:....:....:....:....:....:....:.*.
----------------------------------------------------------------------------
Placing at log-likelihood threshold 2.00...
Start:   109 56 122 263 123 164 301 278 159
Npt-End: (222) 109 56 122 263 123 164 301 278 159
Npt:     222 (212) 109 56 122 263 123 164 301 278 159
Npt:     222 212 109 (63) 56 122 263 123 164 301 278 159
Npt:     222 (197) 212 109 63 56 122 263 123 164 301 278 159
Npt:     222 197 212 (277) 109 63 56 122 263 123 164 301 278 159
No unique placements for 1 remaining marker
```

As we walk through the "order" output, refer to the output section numbers indicated on the right of the page.

In section 1, MAPMAKER displays the parameters it will use in its analysis. These are described in detail in the reference section.

In section 2, MAPMAKER prints the markers in the group (by name and by number), as well the most informative subset from which it will try to find a starting order. To save space, most of the orders output uses numbers rather than names. In this case, MAPMAKER found a 5 marker subset for which one order was preferred over all others by a relative likelihood of at least 13.19.

In section 3, MAPMAKER has begun iteratively adding markers to the starting order. In this first pass, MAPMAKER is using a strict criteria log-likelihood criteria of 3.0 -- only makers which map to a single interval at this log-likelihood are placed in the order. Nine of the group's 15 markers were added to the order this way.

In section 4, MAPMAKER prints the map of the nine marker order it found. A cartoon is drawn showing the placements of the remaining six markers relative to these nine. Here a star indicates a possible map position, while two stars indicate the best map position.

In section 5, MAPMAKER has begun adding markers to the order again, this time at a less-strict log-likelihood threshold of 2.0. This time, all but one of the markers were mapped.

In section 6, MAPMAKER again displays the map of the extended order and the placement cartoon for the remaining unplaced marker.

In section 7, gives the accepted order of loci a name (here "order1") so that we do not need to type the whole list of loci if we wish to perform further analyses. However, these names are a bit fleeting -- other MAPMAKER commands will also set the value of "order1" (etc.) to report their results.

In section 8, MAPMAKER displays the map(s) for any remaining unplaced markers. That is, each unplaced marker is placed in its most likely position, and a new map is calculated.

To conveniently name this order for future reference, we now give it a new name, "best", using MAPMAKER's "let" command. The name "best" will not be changed by MAPMAKER unless we tell it to do so.

```
Map:                                    « Order Output Continues
  Markers          Distance
  222  M153          7.6 cM
  197  M024          3.7 cM
  212  M139          1.1 cM
  277  A114          6.0 cM
  109  M067          3.4 cM
   63  D030         16.6 cM
   56  L062          2.4 cM
  122  M003          6.0 cM
  263  A037          2.3 cM
  123  M007          2.3 cM
  164  M172          3.5 cM
  301  T032          4.7 cM
  278  A063          8.9 cM
  159  M175       ----------
                    68.2 cM   14 markers   log-likelihood= -102.55

Markers placed relative to above map:
         222  197  212  277  109  63   56   122  263  123  164  301  278  159
          :--8-:--4-:--1-:--6-:--3-:-17-:--2-:--6-:--2-:--2-:--3-:--5-:--9-:
  81 2 **.:..*.:....:....:....:....:....:....:....:....:....:....:....:....:..


order1= M153 M024 M139 A114 M067 D030 L062 M003 A037 M007 M172 T032 A063 M175
other1= T031
---------------------------------------------------------------------------
Best placement of T031:
  Markers          Distance
  (81) T031          1.2 cM
  222  M153          7.6 cM
  197  M024          3.7 cM
  212  M139          1.1 cM
  277  A114          6.0 cM
  109  M067          3.4 cM
   63  D030         16.6 cM
   56  L062          2.4 cM
  122  M003          6.0 cM
  263  A037          2.3 cM
  123  M007          2.3 cM
  164  M172          3.5 cM
  301  T032          4.7 cM
  278  A063          8.9 cM
  159  M175       ----------
                    69.4 cM   15 markers   log-likelihood= -104.62
===========================================================================
```

64> ***let best=order1***

```
best= M153 M024 M139 A114 M067 D030 L062 M003 A037 M007 M172 T032 A063 M175
```

Verifying a Map Order

There are some obvious caveats with the "order" analysis which are faced by any such "greedy" algorithm (that is, one which adds markers sequentially, building off the previous result with each iteration). If MAPMAKER makes a mistake early on, then all following results may be incorrect. In practice, this is an addressable issue: (1) One reason that MAPMAKER uses a semi-random starting point and addition order is that you should be able to run the "orders" command repeatedly to verify that you get consistent results; (2) MAPMAKER's error detection algorithm, described below, can be used to limit the possible ill-effects of small data errors; and (3) MAPMAKER provides a variety of simple ways of testing the results found by the "order" command.

One powerful command for accomplishing this test is called "ripple". Essentially, given a known (or assumed) map order, "ripple" instructs MAPMAKER to permute the order of neighboring markers, and to compare the likelihoods of the resulting maps. Any order which has a log-likelihood within some threshold amount of the assumed order's likelihood will be displayed as a viable alternative. Like the "orders" command, "ripple" knows how to use three-point analysis to speed its search, although in the end it uses the power of multipoint analysis with *all* flanking markers to finally compare likelihoods of the consistent orders.

To do this, we first use MAPMAKER's "sequence" command to select the final order we discovered on the previous page, using the name we just defined. Next, we simply type the "ripple" command. By default, this command will permute 5 neighboring loci at a time and flag all alternative orders within a log-likelihood of 2.0 (that is, within 100:1 or better odds) of that of our known order.

As you can see, MAPMAKER lists the sets of neighboring loci it is permuting as it works. Here, MAPMAKER has reported no alternative orders worth considering. Thus, we have little reason to suspect MAPMAKER's "order" results.

65> ***sequence best***

```
sequence #24= chrom10: M153 M024 M139 A114 M067 D030 L062 M003 A037 M007
M172 T032 A063 M175
```

66> ***ripple***

```
================================================================================
Map To Test:
  Markers          Distance
  222  M153          7.6 cM
  197  M024          3.7 cM
  212  M139          1.1 cM
  277  A114          6.0 cM
  109  M067          3.4 cM
   63  D030         16.6 cM
   56  L062          2.4 cM
  122  M003          6.0 cM
  263  A037          2.3 cM
  123  M007          2.3 cM
  164  M172          3.5 cM
  301  T032          4.7 cM
  278  A063          8.9 cM
  159  M175       ----------
                    68.2 cM   14 markers   log-likelihood= -102.55
================================================================================
Window-size: 5  Log-likelihood Threshold: 2.00
Comparing maps with ALL flanking markers...

compare    {M153 M024 M139 A114 M067}... ok
compare ...{M024 M139 A114 M067 D030}... ok
compare ...{M139 A114 M067 D030 L062}... ok
compare ...{A114 M067 D030 L062 M003}... ok
compare ...{M067 D030 L062 M003 A037}... ok
compare ...{D030 L062 M003 A037 M007}... ok
compare ...{L062 M003 A037 M007 M172}... ok
compare ...{M003 A037 M007 M172 T032}... ok
compare ...{A037 M007 M172 T032 A063}... ok
compare ...{M007 M172 T032 A063 M175}    ok
================================================================================
```

Analyzing Another Chromosome

You may have noted that MAPMAKER's "sequence" command continues to display the name of the *chromosome* we had been working on, now with the new sequence of loci. In fact, once we select a chromosome for analysis (as we did using the sequence command on page 27), MAPMAKER will restrict *all* following analyses to that chromosome until explicitly told otherwise. These uses of the "sequence" command are discussed in the "Genome Analysis" section in the Reference Manual.

We will continue mapping this data set by performing an "order" analysis on chromosome 12.

```
67> sequence  chrom12
sequence #25= chrom12: assigned

68> three  point
Linkage Groups at min LOD 3.00, max Distance 50.0
Triplet criteria: LOD 3.00, Max-Dist 50.0, #Linkages 2
'triple error detection' is on.
counting...91 linked triplets in 1 linkage group

                                     log-likelihood differences
 count  markers                         a-b-c  b-a-c  a-c-b
    1:  L016 L041 L058                  -1.03  -6.65   0.00
    2:  L016 L041 T002                  -3.20   0.00  -6.32
    3:  L016 L041 T051                   0.00  -6.73  -2.86
    4:  L016 L041 M027                   0.00  -5.34  -3.78
               .
               .                                   « More  Output  Follows
               .

69> order
Linkage Groups at min LOD 3.00, max Distance 50.0
Starting Orders: Size 5, Log-Likelihood 3.00, Searching up to 50 subsets
Informativeness: min #Individuals 44 (codominant only), min Distance 4.0
Placement Threshold-1 3.00, Threshold-2 2.00, Npt-Window 7
===============================================================================
Linkage group 1, 12 Markers:
   3 L016         21 L041         53 L058         65 T002         85 T051
 127 M027        132 M050        140 M062        188 T064        237 D007
 243 T001        283 A064

Most informative subset: 3 188 53 65 85 140
Searching for a unique starting order containing 5 of 6 informative loci...
Got one using three-point analysis alone.

Placing at log-likelihood threshold 3.00...
Start:   188 85 53 140 65
Npt:     188 85 (21) 53 140 65
Npt:     188 (127) 85 21 53 140 65
Npt:     188 127 85 21 53 (3) 140 65
Npt-End: 188 127 85 21 53 3 140 65 (243)
No unique placements for 3 remaining markers

Map:
  Markers          Distance
  188  T064         14.3 cM
  127  M027          3.7 cM
   85  T051          7.3 cM
   21  L041          4.7 cM
   53  L058         10.0 cM
    3  L016          5.2 cM
  140  M062         17.5 cM
   65  T002          0.0 cM
  243  T001       ----------
                    62.8 cM   9 markers   log-likelihood= -85.08

Markers placed relative to above map:
         188  127  85   21   53    3   140  65   243
          :-14-:--4-:--7-:--5-:-10-:--5-:-17-:--0-:
 132 2 **.:..*.:....:....:....:....:....:....:....:...
 283 2 ...:....:....:..*.:.**.:....:....:....:....:...
 237 2 ...:....:....:....:....:....:....:.**.:....:.*.
-------------------------------------------------------------------------------
```

In this case, MAPMAKER has uniquely ordered 10 of the 12 markers assigned to chromosome 12. The other two markers (D007 and M050) place near the two ends of the 10 marker order.

```
Placing at log-likelihood threshold 2.00...
Start:   188 127 85 21 53 3 140 65 243
Npt:     188 127 85 21 (283) 53 3 140 65 243
No unique placements for 2 remaining markers

Map:
  Markers          Distance
  188  T064         14.3 cM
  127  M027          3.7 cM
   85  T051          7.3 cM
   21  L041          1.1 cM
  283  A064          3.4 cM
   53  L058         10.0 cM
    3  L016          5.2 cM
  140  M062         17.5 cM
   65  T002          0.0 cM
  243  T001       ----------
                    62.6 cM   10 markers   log-likelihood= -86.08

Markers placed relative to above map:
         188  127  85   21   283  53    3   140  65   243
          :-14-:--4-:--7-:--1-:--3-:-10-:--5-:-17-:--0-:
 132 2 **.:..*.:....:....:....:....:....:....:....:....:...
 237 2 ...:....:....:....:....:....:....:.**.:....:.*.

order1= T064 M027 T051 L041 A064 L058 L016 M062 T002 T001
other1= M050 D007
--------------------------------------------------------------------------------
Best placement of D007:
  Markers          Distance
  188  T064         14.3 cM
  127  M027          3.7 cM
   85  T051          7.3 cM
   21  L041          1.1 cM
  283  A064          3.4 cM
   53  L058         10.0 cM
    3  L016          5.4 cM
  140  M062         13.9 cM
 (237) D007          2.3 cM
   65  T002          0.0 cM
  243  T001       ----------
                    61.4 cM   11 markers   log-likelihood= -88.63
--------------------------------------------------------------------------------
Best placement of M050:
  Markers          Distance
 (132) M050          3.5 cM
  188  T064         14.3 cM
  127  M027          3.7 cM
   85  T051          7.3 cM
   21  L041          1.1 cM
  283  A064          3.4 cM
   53  L058         10.0 cM
    3  L016          5.2 cM
  140  M062         17.5 cM
   65  T002          0.0 cM
  243  T001       ----------
                    66.1 cM   11 markers   log-likelihood= -91.48
================================================================================
```

Automatic Error Detection

Our sample data set contains over 14,000 genotypes, and it is almost certain that, in spite of our best efforts in the lab, a few errors may be present. In fact, if even 99% of the genotypes are correct, we would be well ahead of the average often reported in the literature.

In a recent Genomics article (Genomics 14: 604-610), we proposed a new method for dealing with the possibility of genotyping error in these sorts of data sets, a method incorporated into this version of MAPMAKER. Because the Genomics article provides a comprehensive discussion, we only summarize it briefly here.

Essentially, the probability of error is now included in MAPMAKER's fundamental underlying model of genetics -- mistypings are considered rare events, as are crossovers in small regions (the *a priori* probability of error is set using the "error probability" command, discussed in the reference section -- the default is a 1% chance per individual genotype). MAPMAKER thus can weigh the evidence in the entire data set and generate maximum-likelihood maps which are reasonably correct, even when the data have some small number if isolated errors. Moreover, when the "error detection" algorithm is turned on, *all* of MAPMAKER's three-point and multipoint data analysis commands can take it into account when searching to find map orders and distances. However, not all commands display these data (although error detection does affect their results).

As a helpful side-effect, given a multipoint map order and distances, we are able to calculate *a postiori* (e.g. in light of all available raw data) the probability that each individual genotype is right or wrong. These numbers are presented as a "LOD of error", and represent on a log-scale the strength of the evidence that a marker is mistyped. For typical data sets, double-checking all genotypes with a LOD-error of about 1.0 or greater (usually a small fraction of the data set) will correct the vast majority of the errors. Note that MAPMAKER does not calculate LOD-error values for markers at the end of an order (simply because, without flanking markers, there is minimal power to tell recombination from mistyping).

As a quick example of the use of this method, we briefly turn the "error detection" option "on", and then re-display the map shown on the previous pages. Here, you can see that candidate mistypings are printed next to the locus in question, with an indication of the individual number, the flanking marker genotypes, and the LOD of error (only errors with LODs greater than a certain minimum, selected with the "error thresholds command", are displayed). Note that the map distances are also somewhat shrunken, particularly near the locus which had apparent mistypings.

Of course, in actual use we would leave the error detection feature on at all times while constructing maps (also, we would enable error detection for three-point, as well as multipoint analyses, using the "triplet errors" option). This has the unfortunate effect of slowing down most analyses (by roughly a factor of three with F2 intercross data, for example). A more detailed discussion of these features is provided in the Reference manual.

```
81> sequence order1
sequence #29= chrom12: order1

82> map
================================================================================
Map:
  Markers           Distance
   65  T002          17.5 cM
  140  M062           5.2 cM
    3  L016          10.0 cM
   53  L058           3.4 cM
  283  A064           1.1 cM
   21  L041           7.3 cM
   85  T051           3.7 cM
  127  M027          14.3 cM
  188  T064        ----------
                     62.6 cM   9 markers   log-likelihood= -86.06
================================================================================

83> error detection on
'error detection' is on.

84> map
================================================================================
Map:                         Apriori
  Markers           Distance   Prob  Candidate Errors
   65  T002          17.6 cM
  140  M062           5.2 cM   1.0%  -
    3  L016          10.2 cM   1.0%  -
   53  L058           3.3 cM   1.0%  -
  283  A064           1.1 cM   1.0%  -
   21  L041           7.3 cM   1.0%  -
   85  T051           3.2 cM   1.0%  -
  127  M027          13.8 cM   1.0%  [#35 H-A-H 1.77]
  188  T064        ----------
                     61.7 cM   9 markers   log-likelihood= -87.09
================================================================================

85> error detection off
'error detection' is off.
```

Saving and Drawing Information About a Mapped Chromosome

For each chromosome, MAPMAKER can remember many facts. These include:

1. The set of markers "assigned" to that chromosome, as illustrated earlier;
2. A "framework order" for each chromosome -- usually a well-trusted order of a subset of markers assigned to that chromosome, with no ambiguities; and
3. The "placements" of the remaining markers relative to the framework.

MAPMAKER saves these results between sessions, so that you rarely have to repeat the calculations (until you change your raw data set, of course). Among other things, this mechanism is used to save map data for MAPMAKER/QTL.

For example, we will now store the framework and placements we found for chromosome 10. First we set the sequence to indicate the framework order. Using the name we defined earlier, recalling this sequence without much typing is easy.

One concern is that we are again switching chromosomes (from chromosome 12 back to chromosome 10). MAPMAKER's interface is designed to work one chromosome at a time, and we need to tell it explicitly that we are switching. This is a simple matter, as we need only to precede the list of loci with the name of the chromosome, separated by a colon.

The "framework" command is then used to declare the likely framework order and map of this linkage group. This command computes and then stores this map in your data set, which will be read into MAPMAKER/QTL upon loading. For more details on what MAPMAKER/QTL might do with this information, consult the MAPMAKER/QTL manual.

```
86> sequence  chrom10:  best
sequence #30= chrom10: best
```

```
87> framework  chrom10
setting framework for chromosome chrom10...
================================================================================
chrom10 framework:

  Markers          Distance
  222  M153          7.6 cM
  197  M024          3.7 cM
  212  M139          1.1 cM
  277  A114          6.0 cM
  109  M067          3.4 cM
   63  D030         16.6 cM
   56  L062          2.4 cM
  122  M003          6.0 cM
  263  A037          2.3 cM
  123  M007          2.3 cM
  164  M172          3.5 cM
  301  T032          4.7 cM
  278  A063          8.9 cM
  159  M175       ----------
                    68.2 cM   14 markers   log-likelihood= -102.55
================================================================================
```

Once a framework is set, we can compute and store the map position (relative to the framework) of any other markers assigned to this chromosome. For this step, we use MAPMAKER's "place" command.

Again, because we don't like to type, we will use a "sequence" shortcut to indicate that we wish to place all assigned markers on this chromosome (MAPMAKER is smart enough not to place markers already in the framework).

The "place" command's output is very similar to that of the order command, described earlier. Briefly, you see:

Section 1: The log-likelihood threshold used by MAPMAKER (2.0 is the default)

Section 2: The map of the framework, not including any of the placed markers.

Section 3: A summary of the placement results and a cartoon of the possible map positions of each marker.

Section 4: The maximum likelihood map(s) resulting when the placed markers are inserted (individually) into their most likely positions. Only the nearby region of the map may be printed, simply to save space.

More detailed information on the "place" command is presented in the Reference section of this manual.

Lastly, we graphically draw these results in a PostScript file using MAPMAKER's "draw chromosome" command. The resulting graphic image is shown on the following page. How you might display or print these graphics is a topic covered in the Installation Guide.

```
88> sequence  assigned
sequence #31= chrom10: assigned

89> place
Placing markers at log-likelihood threshold 2.00...
================================================================================
1 marker to place on chromosome chrom10...

Framework Map:
  Markers          Distance
  222  M153          7.6 cM
  197  M024          3.7 cM
  212  M139          1.1 cM
  277  A114          6.0 cM
  109  M067          3.4 cM
   63  D030         16.6 cM
   56  L062          2.4 cM
  122  M003          6.0 cM
  263  A037          2.3 cM
  123  M007          2.3 cM
  164  M172          3.5 cM
  301  T032          4.7 cM
  278  A063          8.9 cM
  159  M175       ----------
                    68.2 cM   14 markers   log-likelihood= -102.55

Placing Markers:
   81  T031      - placed off end of framework, likelihood>5.00

Placements:
         222  197  212  277  109  63   56   122  263  123  164  301  278  159
          :--8-:--4-:--1-:--6-:--3-:-17-:--2-:--6-:--2-:--2-:--3-:--5-:--9-:
  81 2 **.:..*.:....:....:....:....:....:....:....:....:....:....:....:....:...
================================================================================
Best placement of T031:
  Markers          Distance
  (81) T031          1.2 cM
  222  M153          7.6 cM
  197  M024          3.7 cM
  212  M139       ----------
                    12.4 cM   4 markers   log-likelihood= -34.53
================================================================================



90> draw  chromosome
Drawing chromosome chrom10 in PostScript file 'chrom10.ps'... ok
```

Here we see the graphic image produced by MAPMAKER. The framework map is drawn vertically, to scale, with the framework markers in bold type. In regular type we would see markers placed relative to the framework (those italicized cannot be placed in a single unique interval). When showing the distances for placed markers, or small distances (when the number would not fit between the tick marks), MAPMAKER draws the distance next to the markers.

GRAPHICS

While we have tried to cover many of the MAPMAKER features you will need to analyze typical data sets, there are many other features you may find useful. In fact, this tutorial really only serves as an introduction to the many features available inside MAPMAKER. As you become more familiar with MAPMAKER, we strongly urge you to read through the reference section of this manual, or (equivalently) make use of the on-line "help" command.

Finally, we exit MAPMAKER using the "quit" command. We can now print out our transcript file, "sample.out" to review the results of our work. Because MAPMAKER saves the results of most of its analyses, it is a simple and quick matter to again start up MAPMAKER on these data to quickly explore any small question which comes up.

```
91> quit
saving map data in file 'mouse.maps'... ok
saving three-point data in file 'mouse.3pt'... ok
```