

## (1) DATA PREPARATION COMMANDS

### LOAD MARKERS Command

Summary: Load marker-locus data  
Argument: <file name>

This command reads in the marker-locus data (allele frequencies for each genetic marker, frequency and penetrance information for the disease). The format of this file must be identical to the Linkage parameter file (output from the PREPLINK program). See the file linkloci.dat as an example of this file format or consult Linkage documentation for further help.

After 3 header lines (only the number of loci on line 1 and the marker order specified on line 3 are relevant and need to be changed), this file must begin with one (and only one) affectation locus describing the disease allele frequencies and penetrances.

GENEHUNTER-MODSCORE includes the functionalities of GENEHUNTER-IMPRINTING; in particular, it allows for a parametric (LOD or MOD score) analysis with imprinting disease models. For individuals who are heterozygous at the disease locus, two penetrance parameters (instead of only one parameter) need to be specified; one for paternal origin and one for maternal origin of the disease allele. The penetrance line should have four parameters and look as follows (with 'm' and '+' specifying the mutant and wild-type allele, respectively, and the paternally inherited allele listed first):

```
f+/+ fm/+ f+/m fm/m
```

In this case, 'imprinting on' needs to be entered before using the 'load markers' command. (If a standard nonimprinting disease model with three penetrance parameters is to be used, the 'imprinting' option should be left turned 'off' which is the default setting when GENEHUNTER-MODSCORE is initiated). See the example file linkloci.imp, as well as the help text for the 'imprinting' command. (Note that this version only allows for an analysis of autosomal loci. In case you want to perform an analysis for the X chromosome with imprinting, please use the xghi executable of GENEHUNTER-IMPRINTING version 1.3.)

Following the disease locus parameters, the information for each marker should be provided, as in the following example:

```
3 6 # D1S1234  
.20 .15 .15 .40 .05 .05
```

The 3 on the first line is obligatory, followed by the number of alleles for the marker. If desired a '#' followed by the name of the marker may be entered and this name will then appear on the Postscript output of the 'total' command and can be used to enter marker orders using the 'use' command. The second line for each marker simply contains the allele frequencies for alleles 1 through 6 in this case. In case that marker allele frequencies do not sum to 1.0 per marker, one should use the 'normalize allfreq' option. See 'help normalize allfreq' for information on how to normalize marker allele frequencies, so that they sum to 1.0 for each marker locus.

In the third to last line you indicate whether you would like to use sex-specific or sex-averaged recombination frequencies. If the first entry is '2', sex-specific recombination frequencies are used. In case that the first entry is '0', sex-averaged recombination frequencies are

employed. Map distances (interlocus distances in the marker order specified on line 3) may be entered on the second to last line in this file format. If sex-specific recombination frequencies should be used, two lines have to be entered, the first line with \*male\* and the second line with \*female\* map distances. (By using this order, consistency with Linkage format is maintained. On all other occasions, including the 'use' and 'read map' commands, GENEHUNTER-MODSCORE expects and prints female map distances and coordinates before the corresponding values for males.) Distances may be specified as either recombination fractions or centiMorgans, with the necessary assumption that if EVERY distance is less than 0.5, they are all assumed to be recombination fractions, otherwise (if ANY distance is greater than 0.5) they are interpreted as centiMorgan distances. Please note that, on line 3, marker loci are numbered 2 to n+1 since '1' is reserved for the disease locus which is ignored in the map order. If '1' (the disease locus) is specified as the first locus on line 3, it is required to start the interlocus-distance line(s) with a dummy recombination value, corresponding to the genetic distance between the disease locus ('1') and the first marker (e.g. '2'), which will always be ignored, followed by the recombination frequency between the 1st and 2nd marker, then 2nd and 3rd marker, and so on.

However, if you have specified that recombination frequencies should be read from a separate map file by the 'read map' command, the two lines with recombination frequencies can be omitted - and, if given, will be ignored. 'Read map' must be called before executing 'load markers'. More details regarding this feature can be found in the help text for the 'read map' command.

See 'help variance components' for information on how quantitative phenotype and covariate data should be specified in this file.

Please note that an analysis with sex-specific recombination fractions can also be performed in the context of the affected-sib-pair and QTL analysis capabilities (commands 'estimate', 'exclude', 'haseman elston', 'ml variance', 'no dom var', 'nonparametric', 'dump ibd' and 'variance components'). Here, the output only includes the sex-averaged genetic positions, i.e., the mean of the male and female coordinates. The corresponding male and female genetic positions can be obtained from the output of the 'scan pedigrees' command.

### **READ MAP Command**

Summary: Load map file with genetic positions  
Argument: <file name>

With this command you can specify that you want to read in marker distances from a publicly available map instead of supplying them yourself at the end of the linkage marker file which is read in by the 'load markers' command. The syntax is as follows:

```
read map <map-file>
```

This version of GENEHUNTER-MODSCORE includes four different map files: 'duffy.txt', 'marshfield.txt', 'nievergelt.txt', and 'rutgers.txt'. The deCODE map can be used as well. Please see the file INSTALL.ghm for instructions regarding how to create the corresponding map file. You can specify a different map file, but the format has to be the same as in the provided files. The 'read map' command will first try to open the map file with the specified name in the current directory. If this fails, the directory specified by the GHM\_DIR environment variable will be accessed to open a file with the name. GHM\_DIR should point to the directory where the GENEHUNTER-MODSCORE installation, including help and map files, are located. By this means, it is not necessary to copy these files to your working directory. An instruction regarding how to

set the GHM\_DIR environment variable with your operating system is given in the INSTALL.ghm file. In the case of an analysis with sex-averaged recombination fractions, the column with sex-averaged genetic positions will be used. When employing sex-specific recombination fractions, the program will only use the male and female positions. In this context, sex-averaged coordinates will be calculated internally as the arithmetic mean of the male and female positions. These averaged coordinates are used as the basis for reporting and plotting results, and they will also be written to the files 'user\_markers\_in\_map' and 'used\_map' (please see below) in addition to the male and female coordinates when a sex-specific map is used. Recombination frequencies which are given in the linkage marker file will be ignored when using a pre-defined map file. The 'read map' command must be executed before 'load markers'.

If the usage of sex-specific genetic distances has been turned on, the coordinates of additional ungenotyped markers given in the map file but not in the linkage marker file will be automatically used to determine the sex-specific genetic coordinates at which the linkage statistics should be calculated. In particular, using the coordinates of additional untyped markers allows for appropriately varying the female/male distance ratio even between two genotyped markers.

There will be four additional output files informing of the used marker positions and the used map:

'user\_markers\_in\_map': This file contains the markers, and their positions, which have been found in the linkage marker file as well as in the map file that you have specified. (The file will also be created when the 'read map' command is not used, containing the markers specified in the linkage marker file.)

'used\_map': This file contains the markers of the specified map file and their positions as they are used by the program. Here, in contrast to the 'user\_markers\_in\_map' file, markers which have not been specified in the linkage marker file are included as well.

'<map>\_omitted\_markers': This file contains a list of markers of the specified map which have been omitted because of incomplete information regarding the genetic positions. (For technical reasons, the list includes X-chromosomal markers of the map, since they do not have a valid male genetic position.)

'<map>\_inconsistent\_positions': Listed are consecutive markers if their positions are inconsistent, i.e., if the female or male genetic coordinate of a marker, in the sex-specific case, or otherwise the sex-averaged coordinate, is smaller than the corresponding coordinate of the previous marker. These markers are not omitted but their positions are corrected by setting the male and/or female position, or the averaged position, to the coordinate of the previous marker plus 0.0001 cM. If the male or female coordinate of a marker is identical to the coordinate of the preceding marker, it will be corrected as well, for technical reasons. However, since such markers do not represent inconsistencies in the narrow sense, they will not be listed in this file. You can inspect the corrected positions of all aforementioned markers in the 'used\_map' file.

If markers specified in the linkage marker file are not found in the map file, a list of these missing markers will be displayed in the standard GENEHUNTER-MODSCORE output. The missing markers will have to be inserted in the map file before continuing.

If the marker order, determined by line 3 of the linkage marker file, differs from the order in the map file, the program will report the first occurrence of such a mismatch and will subsequently quit. Please change the marker order, as defined in line 3, to be the same as in the genetic map file. (Note that line 3 of the linkage marker file appropriately affects the order of genotype columns in the pedigree

file as well, such that no changes are necessary in the pedigree file.)

If a marker name occurs more than once in the linkage marker file, the program will report this and quit. Please make sure that every marker name occurs only once in the linkage marker file.

It is important to note that, in the linkage marker file, the marker name must be preceded by a '#' in order to be recognized.

### USE Command

Summary:       Select the current map for analysis  
Argument:       <genetic map>  
Default:        displays the current map selected

The 'use' command is used to select the current map that the 'scan' command will operate on. It is called in the following manner when using sex-averaged recombination frequencies:

```
use <marker> <distance> <marker> <distance> <marker> ...
```

When using sex-specific recombination frequencies, the following syntax is used:

```
use <marker> <female distance> <male distance> <marker>  
  <female distance> <male distance> <marker> ...
```

Markers may be specified numerically (1 being the first listed in the marker locus file - the affectation locus does not count in this numbering scheme as it does in the Linkage parameter file) or by the names specified in the comment area for each marker. If recombination frequencies are specified in the Linkage parameter file, they will be entered automatically during the "load markers" step. Enter "use" without arguments to see what current linkage map has been entered. If there is no linkage map in the linkage parameter file, it is required to either read a map file with the "read map" command (prior to calling "load markers") or to enter a map using the "use" command before any analysis can take place.

With the "use" command, distances may be specified as either recombination-fractions or centiMorgans, with the necessary assumption that if EVERY distance is less than 0.5, they are all assumed to be recombination-fractions, otherwise (if ANY distance is greater than 0.5) they are interpreted as centiMorgan distances.

If a map file has already been read with the "read map" command, it is not possible to change the recombination frequencies between markers with the "use" command; in this case, "use" can be called only without arguments to display the current map.

## (2) GENEHUNTER-MODSCORE MAPPING COMMANDS

### SCAN PEDIGREES Command

Summary: Analyze pedigree data  
Argument: <file name>

The main analysis command in GENEHUNTER-MODSCORE is the "scan" command (with the "modscore" command executed later for a MOD-score analysis). For each pedigree found in the file indicated, the "scan" command will compute LOD scores and NPL sharing statistics at many positions in the genetic map (entered in the locus parameter file or via the "use" command). In addition, if the "count recs" option is turned on, observed recombinations will be displayed for each map interval at the end of the scan for each pedigree. This can be useful in highlighting likely positions of errors in the data.

The pedigree should be in the Linkage pedigree input format (before running MAKEPED or doing any preprocessing!). Each line of this file must have the following structure:

```
3 12 8 9 1 2 1 1 2 8 3 0 0 4 6 1 3 ... 4.10 0.374
(a) (b) (c) (d) (e) (f) (g) (h -----) (i -----)
```

- (a) pedigree name
- (b) individual ID #
- (c) father's ID #
- (d) mother's ID #
- (e) sex (1=MALE, 2=FEMALE)
- (f) affection status (1=UNAFFECTED, 2=AFFECTED)
- (g) liability class (OPTIONAL) - classes specified in marker data file
- (h) marker genotypes
- (i) phenotype/covariate data (OPTIONAL)

A 0 in any of the disease phenotype or marker genotype positions (as in the the genotypes for the third marker above) indicates missing data. See the file linkped.pre as an example.

A - in the phenotype/covariate data indicates missing data - NB:  
0 is a real value that a phenotype may take on and DOES NOT represent missing phenotype data

In this file format, you may enter as many pedigrees as you wish in a single file. If a pedigree is too large to be computed using a reasonable amount of time and memory, some individuals that provide less information will be discarded and warnings will be printed. Unaffected individuals with no descendants in the pedigree may be discarded with minimal loss of information and these will be the first eliminated should the pedigree be too large. See the "discard" option if you wish to utilize this speed-up in general.

The scan output of each pedigree consists of up to 7 columns of information as follows (depending on the setting of 'analysis type' and on the fact whether sex-specific or sex-averaged genetic distances are used):

- sex-averaged cM position in the scan
- female cM position (if a sex-specific map is used)
- male cM position (if a sex-specific map is used)
- LOD score (computed using the disease model given in the parameter file)
- NPL statistic
- exact computed significance (p-value) of the NPL statistic
- information content of the genotype data

The "total stat" command may be run after a successful "scan" to see the total scores for the entire data set. It is also possible to perform a MOD-score analysis with the "modscore" command.

\*\*\* IMPORTANT \*\*\*

Keep in mind when creating files that there must be a one-to-one correspondence (IN ORDER AND NUMBER) between the markers described in the body of the marker data file (i.e., order of the name and allele frequency definitions, without taking into account line 3 of the file) and the markers that have genotypes listed for them in the pedigree file. (Changing line 3 of the marker data file affects both the order of markers in the marker data file and the order of genotype columns in the pedigree file.)

### **TOTAL STAT Command**

Summary: Show total scores from a scan of multiple pedigrees  
Arguments: <'het'> <fixed-alpha>

The "total" command can only be used after a successful "scan" command of multiple pedigrees. It will display the same 7 columns of output as the "scan" command produced for each pedigree, only now the columns will display the combined values of each statistic (sum of LODscores, combined NPL score, average information content, and p-values of the raw NPL score total). In addition to the screen display of this information (if the "postscript output" option is turned on) postscript graphs of the total NPL statistic (stored in npl\_plot.ps), total LOD score (lod\_plot.ps), and total information content (info\_content.ps) will be created. Please note that, if a sex-specific genetic map is used, the sex-averaged coordinate calculated as the mean of the corresponding female and male genetic position will be used as x-axis coordinate in the postscript plots.

In addition two optional arguments may be entered. If the first argument is the word "het" then LODscores under heterogeneity will also be calculated alongside the regular LODscore sum. If a second numeric argument is provided after the word het, the LODscores under heterogeneity will be calculated assuming a fixed alpha (fraction of pedigrees linked - a number between 0.0 and 1.0). If this second argument is not provided, alpha will be allowed to vary until the HLOD is maximized.

In the context of a MOD-score analysis, the "total stat" command can be executed before and/or after the "modscore" command. If "total stat" is called before "modscore" (but after "scan pedigrees"), it will print the LOD scores obtained under the trait model provided in the locus datafile, as usual. If called after "modscore", "total stat" yields either the LOD-score for all considered genetic positions, obtained under the overall best-fitting model (with the option "modcalc global"), or the MOD score that has been calculated separately for each genetic position ("modcalc single").

The 'total stat' command must be called before running the 'calculate p value' command. For more details please see the 'calculate p value' help text.

## MODCALC Command (abbreviation 'mc')

Summary: activate/deactivate MOD-score analysis  
Argument: <'global', 'single', or 'off'>  
Default: displays the current setting

GENEHUNTER-MODSCORE allows for a MOD-score analysis, in which parametric LOD scores are maximized over the parameters of the trait model, i.e., the penetrances and disease allele frequency. By this means, the trait-model parameter space is explored in an efficient way, and so researchers do not have to rely on a single trait model when performing a parametric linkage analysis. This can be of great help in the context of genetically complex traits, for which the disease model parameters are usually unknown prior to the analysis. Please note that, because of the additional maximization, MOD scores are inflated when compared to LOD scores that were calculated under a single trait model. Therefore, in the context of a MOD-score analysis, significance criteria for LOD scores cannot be applied without correction. GENEHUNTER-MODSCORE allows the user to perform Monte-Carlo simulations in order to calculate p values of MOD scores. This way, significance levels corresponding to certain LOD-score criteria can also be applied to MOD scores. Please see the help text for the 'calculate p value' command for further details, as well as the references mentioned in the INSTALL.ghm file.

When a MOD-score calculation should be performed, the user needs to activate the storage of inheritance-vector probabilities with the 'modcalc' command, for which there are two options, 'global' and 'single' (besides 'off'). This command must be executed before 'scan pedigrees', and should be complemented by the 'modscore' command which performs the actual maximization over trait models, after 'scan pedigrees'.

With 'modcalc global', the maximum of the LOD score over all assumed disease-locus positions along the marker map is determined for each trait model, and this maximum is maximized over different models. When using the 'global' option, the inheritance-vector distribution given the marker data (named P\_complete in the original GENEHUNTER reference, Kruglyak et al., Am J Hum Genet 58:1347-1363, 1996), obtained during 'scan pedigrees', needs to be stored in memory for all assumed disease-locus positions. During the 'modscore' command, all of these distributions will be reused for every tested trait model. Therefore, the memory requirements can be substantial for datasets with larger pedigrees. If the amount of main memory is not sufficient, the range of considered genetic positions can be restricted with the 'positions' command. When relying on swap memory with 'modcalc global', computational efficiency will drastically decrease, due to a large degree of paging.

With the 'modcalc single' option, a separate maximization over trait models will be done by the 'modscore' command for each assumed disease-locus position. This yields a MOD score, in conjunction with the penetrances and disease allele frequency of the best-fitting trait model, for every genetic position. The parameters can be regarded as an estimate of the genetic effect at a particular locus. In the case of two disease genes at separate loci on the same chromosome with markedly different trait-model parameters, only the locus with the stronger signal will be identified when using the 'global' option, but probably both of them will be found with 'modcalc single'. When using the 'single' option, the inheritance-vector probabilities need to be stored only for a single genetic position. On the other hand, computation-time demands will be higher than with 'modcalc global', since a separate round of maximization needs to be done for each genetic position. Here, if sufficient memory is available, time can be saved by storing the scoring-function vectors (i.e., the disease-locus likelihood) in memory for a certain number of trait models. The number of models to be stored is defined with the 'saved models' command. In addition, it is possible to restrict the range of genetic positions using the 'positions' command in order to reduce the computation time.

Details on reducing the computation-time demands through usage of a customized set of trait models for the maximization is given in the 'maximization' command section of this help.

It should be noted that restricting the genetic positions to be analyzed with the 'positions' command is preferable to restricting the number of markers taken into account with the 'use' command. When applying the 'positions' command to define a smaller range of genetic positions assumed for the disease locus, GENEHUNTER-MODSCORE will still extract the complete multipoint information by using all available markers. Therefore, this option should be used if enough memory is available to include the full marker set.

Please also see the help texts for the 'modscore', 'maximization', 'positions', 'saved models', and 'calculate p value' commands.

'modcalc' is 'off' when GENEHUNTER-MODSCORE is initiated.

### **ALGEBRAIC CALCULATION Command (abbreviation 'alg')**

Summary: activate/deactivate algebraic calculation mode  
Argument: <'on' or 'off'>  
Default: displays the current setting

The use of the algebraic algorithm significantly speeds up a parametric MOD-score analysis compared to the classic calculation mode implemented in former versions of GENEHUNTER-MODSCORE (e.g. version 3.0). The speed-up is especially pronounced when many sets of trait-model parameters are evaluated during the maximization. The algebraic algorithm reduces the effective number of inheritance vectors by collapsing them into classes with identical disease-locus-likelihood contribution. To this end, the disease-locus-likelihood contribution of each inheritance vector is stored in its algebraic form as a sum of products of penetrances and disease-allele frequencies. Inheritance vectors with the same disease-locus-likelihood contribution are joined together in an inheritance-vector class. This concept is even extended across pedigrees, such that the disease-locus-likelihood contributions of all inheritance vector classes can be numerically calculated in a single step for the entire dataset rather than having to recompute them many times. Finally, the result for a certain inheritance vector class is used for all pedigrees with inheritance vectors that are members of that particular class. Unlike the previous algorithm, the new algebraic algorithm neither requires peeling of nuclear families nor loop-breaking.

Please see the following publication for details:

Brugger M. and Strauch K.: Fast Linkage Analysis with MOD Scores Using Algebraic Calculation, Hum Hered 2014;78:179-194.

Use of the algebraic algorithm is only reasonable in conjunction with the 'modscore' command, especially with the 'modcalc single' option, for which a separate maximization over trait models is done for each assumed disease-locus position. Furthermore, usage of a dense grid of initially evaluated trait models ('maximization dense') is recommended, which leads to a more thorough maximization over trait-model parameters.

The new algebraic calculation mode can be readily combined with the 'calculate p value' option to determine empirical p values by performing simulations under the null hypothesis of no linkage. Hence, the evaluation of many sets of tested trait-model parameters during a MOD-score analysis in conjunction with empirical p-value calculation is now feasible within a reasonable amount of time.

The 'saved models' option will not be used with 'algebraic calculation' turned 'on' because the saving of scoring-function vectors does not lead to a further speed-up in the context of the algebraic algorithm.

For compatibility reasons, the user can switch to the classic calculation mode as implemented in former versions of GENEHUNTER-MODSCORE by specifying 'algebraic calculation off'.

'algebraic calculation' is 'on' when GENEHUNTER-MODSCORE is initiated. If the user wants to use the original algorithm instead, 'algebraic calculation' must be turned 'off' before 'scan pedigrees'.

### MAXIMIZATION Command

Summary:           customizes MOD score routine  
Argument:         <'standard', 'dense', 'grid', 'small', or 'user'>  
Default:           displays the current setting

With parametric linkage analysis, maximizing the likelihood for a given dataset involves a systematic variation of one or more parameters of the model. In a simple LOD-score analysis only the genetic position of the trait locus is varied, while the trait model parameters (penetrances and disease allele frequency) are fixed. With a MOD-score analysis, the penetrances and disease allele frequency are included in the maximization as well. This can be of great help in the context of genetically complex traits, for which the disease model parameters are usually unknown prior to the analysis. In most cases the resulting optimization problem cannot be solved by analytical means. Here the likelihood has to be calculated under different trait models in order to find the MOD score numerically. GENEHUNTER-MODSCORE performs this task in an efficient and customizable way.

The MOD-score calculation is started with the 'modscore' command which has to be executed after 'scan pedigrees' (with possibly 'total stat' or other commands between them). In addition, if MOD scores should be calculated, it is necessary to specify 'modcalc global' or 'modcalc single' prior to the 'scan pedigrees' command, in order to activate the storage of inheritance-vector probabilities. Please see the help text for the 'modcalc' and 'modscore' commands for details.

The 'maximization' command determines the way in which the disease-model parameters are varied when calculating the MOD score. With 'maximization standard' (i.e., the default setting), the MOD-score analysis is done in two steps. First, in addition to the trait model specified in the locus datafile, GENEHUNTER-MODSCORE tests a set of predefined trait models. In particular, the disease allele frequency is set to the values 0.01, 0.05, and 0.10; the phenocopy rate  $f_{+/+}$  is set to the values 0.001, 0.01, and 0.05; the other penetrances are set to the value of  $f_{+/+}$  as well as to 0.10, 0.50, and 0.95. Taking all possible combinations of these values forms the set of initially tested models. With the option 'penetrance restriction on' (which is the default), the heterozygote penetrances (two of them in case of imprinting) are constrained to be not smaller than  $f_{+/+}$  and not greater than the homozygous mutant penetrance  $f_{m/m}$ . This restriction can be turned off, e.g. in order to allow for overdominance or protective loci. With the choice of parameters given above, there are 81 initially tested trait models with 'imprinting off' and 'penetrance restriction on', 135 models with 'imprinting off' and 'penetrance restriction off', 261 models with 'imprinting on' and 'penetrance restriction on', and 567 models with 'imprinting on' and 'penetrance restriction off'. Additional user-defined trait models can be added to this first round with the 'model' command. In case of more than one liability class, or if the LOD score is below zero for all initial models, a further model with all penetrances equal for the currently optimized liability class is tested as well. Then, the LOD-score maximum over the initially tested models is used as the starting point for a fine maximization. Here, in order to reach the final MOD-score maximum, parameters are varied by steps of 0.01; if they fall

below 0.05, the step size is reduced to 0.005 or even to a smaller value. With the 'dimensions' command, the user can determine the number of parameters that are jointly varied during fine maximization. The default is 2; specifying a higher value leads to a more thorough maximization at the price of a larger number of disease-locus likelihood evaluations and, hence, more computing time. The penetrance restriction, if activated, applies to the fine maximization as well. The same holds for the restriction of the disease allele frequency (which by default cannot exceed 0.5), as defined by the 'allfreq restriction' command.

With 'maximization dense', a larger set of predefined trait models is tested before the fine maximization than with 'maximization standard'. Here the disease allele frequency is set to the values 0.01, 0.05, 0.10, and 0.50; the phenocopy rate  $f_{+/+}$  is set to the values 0.001, 0.01, 0.05, 0.50, 0.95, 0.99, and 0.999; the other penetrances are set to the value of  $f_{+/+}$  as well as to 0.05, 0.10, 0.50, 0.90, and 0.95. As before, taking all possible combinations of these values forms the set of initially tested models. The set obtained with 'maximization dense' is a superset of the models initially tested with 'maximization standard'. Due to the choice of grid points mentioned above, the set of models with 'maximization dense' is symmetric with respect to affection status, such that for every vector of penetrances ( $f_{+/+}$ ;  $f_{m/+}$ ;  $f_{+/m}$ ;  $f_{m/m}$ ) the complementary penetrance vector ( $1-f_{+/+}$ ;  $1-f_{m/+}$ ;  $1-f_{+/m}$ ;  $1-f_{m/m}$ ) is included in the set of models as well. Thus, the set of initially tested models does not change if the 'affected' phenotype is replaced by 'unaffected' and vice versa. There are 236 trait models with 'imprinting off' and 'penetrance restriction on', 848 models with 'imprinting off' and 'penetrance restriction off', 988 models with 'imprinting on' and 'penetrance restriction on', and 4928 models with 'imprinting on' and 'penetrance restriction off'. During fine maximization, the step size of 0.01 is reduced to 0.005 or even to a smaller value not only if a parameter falls below 0.05 (as is the case with 'maximization standard'), but also if a parameter exceeds 0.95, for symmetry reasons. Additional commands related to parameter optimization work in the same way as described above for the 'maximization standard' option.

For the MOD-score analysis, the inheritance-vector distribution given the marker data (named `P_complete` in the original GENEHUNTER reference, Kruglyak et al., *Am J Hum Genet* 58:1347-1363, 1996), obtained during 'scan pedigrees', depends on the genetic position, but not on the assumed trait model. Hence, it needs to be calculated just once. Only the scoring-function vectors, which contain the contribution of the disease locus to the likelihood, must be newly calculated for each trait model. Therefore, evaluation of the disease-locus likelihood becomes the predominant step in terms of computation time in the context of a MOD-score analysis. The core of GENEHUNTER-MODSCORE is a highly optimized engine for the calculation of the disease-locus likelihood, which is almost 6 times as fast as the original version. With 'modcalc global', the outer program loop is over trait models, and the inner loop is over the different positions assumed for the disease locus. The inheritance-vector distribution given the marker data is stored in memory for all pedigrees and assumed disease-locus positions, since the LOD score is evaluated at every position for each trait model.

With the 'modcalc single' option, the outer and inner loop are exchanged (now outer loop - positions, inner loop - trait models). Here, `P_complete` must be saved only for the current position. On the other hand, since the trait models need to be checked for every position, the scoring-function vectors will have to be evaluated more often than with 'modcalc global', and so the computation time roughly increases by a factor of the number of genetic positions. However, a scoring-function vector for given trait-model parameters does not depend on the genetic position. Thus, the scoring-function vectors can be saved in memory when calculated for the first time and recalled for a later position, which avoids time-consuming recomputation. In order to achieve this, GENEHUNTER-MODSCORE applies a smart model-saving scheme. Please see the help text for the 'saved models' command for details.

The model-saving option allows for MOD-score calculations even with larger pedigrees. In case that testing a large number of models is infeasible due to computation-time demands, GENEHUNTER-MODSCORE provides the opportunity to perform a MOD-score analysis only with predefined sets of trait models ('maximization grid', 'small', or 'user'), i.e., without a subsequent fine maximization. In this case, the set of trait models is identical for every genetic position, and so all of the scoring-function vectors can be calculated only once and stored in memory. This allows for a computationally efficient approximation of the MOD score for each locus. While 'maximization grid' uses the set of trait models mentioned above for the 'dense' option, 'small' performs a maximization with the same trait models as for the 'standard' option. With 'maximization user', only the user-defined trait models are employed, i.e., those models defined by the 'model' command (please see the corresponding help text). This allows the user to customize the type of maximization.

Although the gain in terms of computation time is especially relevant for 'modcalc single', all maximization options are available for both 'modcalc global' and 'single'. With a genetically complex trait, it is conceivable that two (or even more) disease loci are located on the same chromosome. In this case, using 'modcalc single' gives the opportunity to detect both of them. Therefore, unless the analysis is limited to a small genetic region, researchers are encouraged to perform their MOD-score analysis with 'modcalc single'.

For further information please have a look at the help texts of the commands 'modcalc', 'modscore', 'model', 'imprinting', 'positions', 'liability class', 'join liability classes', 'penetrance restriction', 'allfreq restriction', 'dimensions', 'saved models', and 'long mod output'.

When GENEHUNTER-MODSCORE is initiated, 'standard' is the default option for the maximization command.

### **MODSCORE Command**

Summary:        perform a MOD-score analysis  
No Arguments

GENEHUNTER-MODSCORE allows for a MOD-score analysis, in which parametric LOD scores are maximized over the parameters of the trait model, i.e., the penetrances and disease allele frequency. By this means, the trait-model parameter space is explored in an efficient way, and so researchers do not have to rely on a single trait model when performing a parametric linkage analysis. This can be of great help in the context of genetically complex traits, for which the disease model parameters are usually unknown prior to the analysis. Please note that, because of the additional maximization, MOD scores are inflated when compared to LOD scores that were calculated under a single trait model. Therefore, in the context of a MOD-score analysis, significance criteria for LOD scores cannot be applied without correction. GENEHUNTER-MODSCORE allows the user to perform Monte-Carlo simulations in order to calculate p values of MOD scores. This way, significance levels corresponding to certain LOD-score criteria can also be applied to MOD scores. Please see the help text for the 'calculate p value' command for further details, as well as the references mentioned in the INSTALL.ghm file.

The MOD-score calculation is started with the 'modscore' command which must be executed after 'scan pedigrees' (with possibly 'total stat' or other commands between them). In addition, if MOD scores should be calculated, it is necessary to specify 'modcalc global' or 'modcalc single' prior to the 'scan pedigrees' command, in order to activate the storage of inheritance-vector probabilities. Please see the help text for the 'modcalc' command for details.

If 'imprinting' is turned 'on', GENEHUNTER-MODSCORE performs a MOD-score analysis under the assumption of trait models with four penetrances. By this means, like with the previous version GENEHUNTER-IMPRINTING, individuals who are heterozygous at the trait locus can be distinguished by the parent who transmitted the disease allele. This allows the user to adequately take imprinting into account, which is also called parent-of-origin effect. With 'imprinting off' (which is the default), standard trait models with three penetrances are used for the analysis.

The MOD score is calculated as follows. First, in addition to the trait model specified in the locus datafile and additional models defined by the user with the 'model' command, GENEHUNTER-MODSCORE tests a set of predefined trait models. Then, a fine maximization is performed (if 'maximization' is set to 'standard' or 'dense'), using the LOD-score maximum over the initially tested models as the starting point. Please see the help text for the 'maximization' command for details on the maximization procedure.

In conjunction with the MOD score, the analysis yields the penetrances and disease allele frequency of the best-fitting trait model. In order to obtain a more instructive representation of the trait model, the penetrances can be transformed into the dominance index D and (if 'imprinting' is 'on') the imprinting index I, defined as follows:

$$D = \frac{fm/+ + f+/m - f+/+ - fm/m}{fm/m - f+/+} \quad I = \frac{fm/+ - f+/m}{fm/m - f+/+}$$

Here, 'm' denotes the disease allele and '+' the wild-type allele, and in the heterozygous situation, the paternally inherited allele is listed first. These definitions ensure that D becomes 1 (fully dominant) if both heterozygote penetrances equal fm/m, -1 (fully recessive) if both heterozygote penetrances equal f+/+, and 0 (semidominant or additive) if the average of the two heterozygote penetrances is half-way between the two homozygote penetrances, irrespective of the values of f+/+ and fm/m. Likewise, it is ensured that I becomes 1 (complete maternal imprinting) or -1 (complete paternal imprinting) if one heterozygote penetrance equals f+/+ and the other equals fm/m, and that I=0 if both heterozygote penetrances are equal. D and I are constrained to take values between -1 and 1 only in case that the 'penetrance restriction' is 'on'; otherwise, their values can be <-1 or >1 as well. If f+/+ and fm/m are equal, D and I are undefined; in such cases, both will be arbitrarily set to zero. When using the 'modcalc global' option, the analysis yields the overall MOD score, together with the best-fitting model and the corresponding values of D and I. With 'modcalc single', the MOD score as well as penetrances and disease allele frequency of the best-fitting trait model, and the corresponding values of D and I, are obtained for every genetic position. These parameters can be regarded as an estimate of the genetic effect at a particular locus. Of course, the values of D and I are only of practical importance in genetic regions with high MOD scores. When the 'postscript' option is 'on', a PostScript plot of the MOD score, together with D and I, as a function of the assumed disease-locus position, is created with 'modcalc single'. This allows the user to immediately see which type of model (dominant, recessive, semidominant/additive, paternal or maternal imprinting) fits the data best for a certain genetic position.

If there is more than one liability class, the class for which the penetrances should be optimized can be selected with the 'liability class' command; the default is 1. Please note that the disease allele frequency applies to all liability classes. The penetrances reported by the 'modscore' command as well as the associated D and I (see above) apply only to the currently optimized liability class. This holds for the text output as well as for the PostScript plot. The 'join

liability classes' command can be used if the penetrances of two or more liability classes should be jointly varied; here, each penetrance is assumed to be the same for all joined liability classes. It is also possible to perform separate maximizations over penetrances of several liability classes, e.g. for males and females, individuals of different age, or different levels of risk due to environmental factors. This can be done by repeatedly executing the 'liability class' and 'modscore' commands for different liability classes within a single program run. Here, after the end of each 'modscore' round, the penetrances of the optimized liability class and the disease allele frequency are updated with the best-fitting values. By this means, they are reused for a subsequent MOD-score analysis in which the penetrances of a different liability class will be varied. With 'modcalc global', there is only one trait model that applies to all genetic positions. When using the 'modcalc single' option, the trait model parameters will be updated separately for each position, such that the best-fitting values for that particular position are reused for a following MOD-score calculation. It is even possible to perform several of these maximization rounds until the overall optimized MOD score converges. Such a highly explorative procedure clearly goes far beyond classical LOD-score analysis, and by scope and intent is a data-mining procedure.

With 'modcalc global', the 'modscore' command will always print the parameters of all tested models and the results during the course of the analysis. When using the 'modcalc single' mode, where a separate maximization is done for each genetic position, only the MOD score and parameters of the best-fitting model are printed for each position, as default ('long mod output off'). In order to obtain the output of all tested trait models for each genetic position, please turn 'long mod output on'. This will lead to files with length of several megabytes. If the set of genetic positions assumed for the disease locus has been restricted with the 'positions' command, a MOD score of 0 will be reported at the omitted genetic positions for technical reasons, in the text output as well as in the PostScript plots.

When 'display scores' is 'on' (default), the 'modscore' command with 'modcalc global' option will print the LOD scores for all pedigrees and genetic positions, obtained under the overall best-fitting model, at the end of the analysis. In case of 'modcalc single', the MOD-score contribution of each pedigree will be printed for every genetic position. This allows for the creation of a Gnuplot graph of the MOD score, displayed by the single family contributions, by applying the Perl script GH\_modview to the GENEHUNTER-MODSCORE output. If no output of the scores by pedigree is desired, 'display scores' should be turned 'off'.

The 'total stat' command, executed after 'modscore' with 'modcalc global', prints the LOD scores for all genetic positions under the overall best-fitting model, summed over all pedigrees. The output is together with the marker information content and possibly NPL score obtained during 'scan pedigrees'. In case of 'modcalc single', the MOD score, summed over all pedigrees, will be printed for each genetic position, together with marker information content and possibly NPL score. (In order to print the total LOD score obtained under the trait model specified in the locus datafile, the 'total stat' command should be executed between the 'scan pedigrees' and 'modscore' commands.) Please note that with 'modcalc global', strictly speaking, only the highest score over all genetic positions can be called 'MOD score', since the best-fitting model as the maximization result only applies to this position. When looking at a position that differs from the one that yields the overall highest score, a separate maximization may lead to a higher MOD score at that locus, under a different trait model. In the case of two disease genes at separate loci on the same chromosome with markedly different trait-model parameters, only the locus with the stronger signal will be identified when using the 'global' option, but probably both of them will be found with 'modcalc

single'.

Please see also the help texts of the related commands 'modcalc', 'maximization', 'model', 'imprinting', 'positions', 'liability class', 'join liability classes', 'penetrance restriction', 'allfreq restriction', 'dimensions', 'saved models', 'long mod output', and 'calculate p value'.

### **CALCULATE P VALUE Command (abbreviation 'cpv')**

Summary: calculates p-values for MOD or LOD scores  
Argument: <file name>

GENEHUNTER-MODSCORE allows for simulations under the null hypothesis of no linkage in order to determine empirical p-values for MOD scores or simple LOD scores calculated under a single model.

The p-value calculation is started with the 'calculate p value' command which must be executed after 'scan pedigrees' (and 'modscore' if a MOD-score analysis should be performed) with possibly 'total stat' or other commands between them. Furthermore, if p-values should be calculated, it is necessary to specify the desired 'number of replicates' prior to the 'scan pedigrees' command, in order to store some information regarding the real data set. Please see the help text for the 'number of replicates' command for details.

The Modus operandi for the simulation contains different steps, which are implemented to take place during one program run. First, the real data will be analyzed with the customary program options. Afterwards, for each replicate, the flow of alleles from one generation to the next is simulated with respect to the given marker allele frequencies and, in case of multipoint analysis, the recombination values for the inter-marker distances, using the pedigree structure of the real data sample. Then, the replicate is analyzed with the same program options as specified for the real data set. The last two steps will be repeated until the predefined number of n replicates have been simulated and analyzed (cf. the help text for 'sequential simulation' concerning an exception for the last step of this procedure). Finally, the empirical p value is reported.

With respect to the different analysis options (LOD vs. MOD, 'modcalc single' vs. 'modcalc global'), empirical p-values are reported in a different way. With a 'single-model' LOD-score analysis or a MOD-score analysis using 'modcalc single', empirical p-values are given pointwise for each considered position. These pointwise p-values may be judged according to the guidelines given by Lander and Kruglyak (Nature Genetics 11:241-247, 1995).

In addition to the pointwise results, a p-value for the best LOD or MOD score of the real data set is determined by counting the number of replicates with a score at least as high as the maximum real data result, irrespective of the locus at which the maximum occurs. Hence this p-value for the overall best score is 'region-wide' (or 'chromosome-wide') and not pointwise, and therefore it still needs to be corrected for the 'genome' (i.e., for all 23 chromosomes), not only in the context of whole genome scans. In case a MOD-score analysis is conducted using the 'modcalc global' option, only the region-wide p-value is reported.

As described with the 'maximization' option, a scoring-function vector for given trait-model parameters does not depend on the genetic position. Thus, the scoring-function vectors can be saved in memory when calculated for the first time and recalled for a later position. Since many trait models (at least all of the user-defined and predefined models) do not change from one replicate to another with the simulation routine, the stored scoring-functions are even reused for different replicates, which leads to an additional, substantial saving of computation time. It is

obvious that this speedup becomes more important with more time needed to calculate one single scoring function, that is, with increasing family size.

Along with the 'calculate p value' command a filename must be specified. In this file the overall maximum score for each replicate is stored, together with the pedigrees that had to be skipped during the analysis of each replicate due to uninformative markers. In addition to the results of the p-value calculation, the starting value for the random seed is reported as well.

There are some limitations regarding the usage of the 'calculate p value' command:

The p-value calculation only works with the 'analysis LOD' or 'analysis BOTH' option. In the latter case the NPL analysis will be deactivated during the calculation of p-values.

It is only possible to calculate p-values for MOD scores if, after calling 'modscore', no further changes have been done by the commands 'liability class', 'imprinting', 'dimensions', 'maximization', 'model', 'penetrance restriction', 'allfreq restriction', 'highest allfreq', and 'positions'. Calculation of p-values is also disabled if 'modscore' has been executed more than once, e.g. in case of separate maximizations over penetrances for different liability classes. If more than one liability class has been defined, it is possible to jointly vary the penetrances of two or more liability classes in a MOD-score analysis by using the 'join liability classes' command (which has to be executed between 'load markers' and 'scan pedigrees').

It is not possible to use any further analysis commands after completion of the p-value calculation.

Please see also the help texts of the related commands 'number of replicates', 'sequential simulation', 'store replicates', 'untyped founders', 'full information', 'simulate untyped', 'best position', 'set random seed', 'show distribution', 'liability class', and 'join liability classes'.

### **MODEL Command**

Summary: add a user-defined trait model to MOD-score analysis  
Arguments: <disease allele frequency> <3 or 4 penetrances>

For the maximization during a MOD-score analysis, a set of predefined trait models will be tested initially, before the fine maximization. Please see the help text for the 'modscore' command for details. With the 'model' command, a user-defined trait model can be added to the list of initially tested models. In case of 'imprinting on', the command and parameters should be entered in the following format:

```
model <p> <f+/+> <fm/+> <f+/m> <fm/m>
```

When 'imprinting' is 'off', the format should be as follows:

```
model <p> <f+/+> <fHet> <fm/m>
```

Here, 'p' denotes the disease allele frequency, and in the context of the penetrances (named 'f'), '+' stands for the wild-type allele and 'm' for the disease allele. In the heterozygous situation, with 'imprinting on', the paternally inherited allele is listed first; if 'imprinting' is 'off', the heterozygous genotype is named 'Het'.

Please see also the help text of the related command 'maximization'.

### **IMPRINTING Command**

Summary: activate/deactivate imprinting analysis  
Argument: <'on' or 'off'>  
Default: displays the current setting

Turning the 'imprinting' option on before a 'load markers' command causes GENEHUNTER-MODSCORE to use a disease model for parametric (LOD or MOD score) analysis that takes into account a parent-of-origin effect. For individuals who are heterozygous at the disease locus, two penetrance parameters (instead of only one parameter) need to be specified in the locus datafile; one for paternal origin and one for maternal origin of the disease allele. The penetrance line should have four parameters and look as follows (with 'm' and '+' specifying the mutant and wild-type allele, respectively, and the paternally inherited allele listed first):

f+/+ fm/+ f+/m fm/m

See the example file linkloci.imp. (Note that this version only allows for an analysis of autosomal loci. In case you want to perform an analysis for the X chromosome with imprinting, please use the xghi executable of GENEHUNTER-IMPRINTING version 1.3.)

If a standard nonimprinting disease model with three penetrance parameters is to be used, the 'imprinting' option should be left turned 'off' which is the default setting when GENEHUNTER-MODSCORE is initiated.

Once the 'load markers' command has been executed, the 'imprinting' option can only be changed from 'off' to 'on' (but not vice versa). Therefore, after having done a 'modscore' analysis with 'imprinting off', it is possible to turn 'imprinting on' within the same program run, and to perform an additional 'modscore' round that takes imprinting into account.

### **MOBIT SIMULATION Command (abbreviation 'mob')**

Summary: obtain empiric permutation p value for the MOBIT  
Argument: <'on' or 'off'>  
Default: displays the current setting

With GENEHUNTER-MODSCORE, a test for imprinting can be performed by looking at the difference between the MOD score accounting for imprinting ('imprinting' option 'on') and the standard nonimprinting MOD score ('imprinting' option 'off'), which can be calculated in a single program run. This test is called 'MOBIT' (MOD score Based Imprinting Test) and is thoroughly described in: Brugger M., Knapp M., and Strauch K.: Properties and Evaluation of the MOBIT - a novel Linkage-based Test Statistic and Quantification Method for Imprinting, SAGMB 2019;18(4). In the case of linkage, the null hypothesis of the MOBIT would be linkage, but no imprinting. The corresponding null distribution of the MOBIT should then follow a chi-squared distribution with 1 degree of freedom. However, the quality of the asymptotic null distribution is quite poor under boundary conditions, which are expected to be common when analyzing real datasets. In addition, the aforementioned asymptotic properties of the MOBIT do not hold in the absence of linkage. Therefore, calculation of empiric p values are highly recommended, which can be achieved by either performing 'ab initio' simulations of genotype data using the best-fitting nonimprinting model obtained from the MOD score analysis or by randomization of the origin of the parental alleles in offspring of every nuclear family within a given pedigree. The latter method (called 'perm' in the reference publication) can be used by employing the 'mobit simulation' command. The null hypothesis of the 'perm' method corresponds to an imprinting effect with expectation value 0, conditional on the linkage information of the real dataset. This means that genotypes remain unchanged using this

procedure. A more detailed explanation of the 'perm' method can be found in the Appendix of the abovementioned reference publication. Since the 'perm' method is most efficiently calculated using multiple CPUs, the GENEHUNTER-MODSCORE package contains a bash script ('run\_mobit\_permutations.sh') that can readily be used and adapted to run MOBIT simulations in parallel. It is of note that 'mobit simulation' must be set before loading pedigrees ('scan pedigrees' command) and cannot be used together with the 'calculate p value' command that calculates empiric p values for the linkage test. Moreover, the 'algebraic calculation' mode must be enabled (default), and the analysis type must be set to 'LOD' or 'BOTH'.

#### **INCLUDE UNTYPED Command (abbreviation 'iu')**

Summary: include/exclude untyped persons with no kids  
Argument: <'on' or 'off'>  
Default: displays the current setting

With earlier GENEHUNTER versions, non-genotyped individuals with no children are always discarded. However, for a LOD or MOD-score analysis, individuals without marker genotypes but with available trait phenotype help to reconstruct their parents' trait-locus genotypes. Therefore, they do contribute to the LOD or MOD score - in some cases even to a substantial degree. For this reason, with GENEHUNTER-MODSCORE, such persons by default are included ('include untyped on'). If these individuals should nevertheless be excluded from the analysis, e.g. in order to save computation time or for compatibility with older versions, 'include untyped' needs to be turned 'off'.

#### **POSITIONS Command**

Summary: define genetic positions for the MOD-score analysis  
Arguments: <lowest position> <highest position>

As the default, all genetic positions assumed for the disease locus (as defined by the 'increment' and 'off end' commands) will be taken into account for a MOD-score analysis. The 'positions' command can be used to restrict the range of genetic positions. The first and second argument denote the lowest and highest disease-locus position for the MOD-score calculation. In case that a sex-specific map is used, these bounds apply to the sex-averaged coordinate, i.e., the mean of female and male genetic positions. Please note that, for technical reasons, a MOD score of 0 will be reported at the omitted genetic positions, in the text output as well as in the PostScript plots. A restriction of the range of genetic positions can be useful in order to save memory, with 'modcalc global', or in order to reduce the computation time, when using the 'modcalc single' option. Please also see the help text for the 'modcalc' command.

#### **LIABILITY CLASS Command (abbreviation 'lc')**

Summary: define liability class to be optimized (MOD score)  
Argument: <liability class>  
Default: displays the current setting

If there is more than one liability class, the class for which the penetrances should be optimized in a MOD-score analysis can be selected

with the 'liability class' command. Please note that the disease allele frequency applies to all liability classes. The penetrances reported by the 'modscore' command as well as the associated D and I (see above) apply only to the currently optimized liability class. This holds for the text output as well as for the PostScript plot. If the 'join liability classes' command is used to jointly vary the penetrances of two or more liability classes, the recoded class number should be specified as argument of the 'liability class' command.

By default, the penetrances of liability class 1 will be optimized.

#### **JOIN LIABILITY CLASSES Command (abbreviation 'jl')**

Summary: join liability classes for a MOD-score analysis  
Arguments: <numbers of liability classes to be joined>

If there is more than one liability class, the 'join liability classes' command can be used in order to jointly vary the penetrances of two or more liability classes in a MOD-score analysis. The classes, which are given as arguments of 'join liability classes', will be assigned a common recoded number. In case that the penetrances specified in the locus datafile differ between the liability classes to be joined, the penetrances of the liability class that is given as the first argument will be used for the joined liability class in the context of the initially tested model. To join a further set of liability classes, the recoded class numbers should be specified as arguments. The recoded number also needs to be used with the 'liability class' command to select the class for which the penetrances should be varied in the MOD-score analysis. In this context, during maximization, each penetrance is assumed to be the same for all joined liability classes. Please note that the liability classes in the pedigree file still need to reflect the initial definition in the locus datafile. The only purpose of this command is to allow for a MOD-score analysis in which the penetrances of several liability classes should be jointly varied, without need to recode the input files. The 'join liability classes' command must be executed between 'load markers' and 'scan pedigrees'.

#### **PENETRANCE RESTRICTION Command (abbreviation 'pr')**

Summary: activate/deactivate penetrance restriction (MOD)  
Argument: <'on' or 'off'>  
Default: displays the current setting

The 'penetrance restriction' affects the MOD-score calculation. If the option is 'on', the heterozygote penetrances (two of them in case of imprinting) are constrained to be not smaller than the homozygous wild-type penetrance  $f_{+/+}$  and not greater than the homozygous mutant penetrance  $f_{m/m}$ . This restriction can be turned 'off', e.g. in order to allow for overdominance or protective loci.

When GENEHUNTER-MODSCORE is started, 'penetrance restriction' is 'on'.

### **ALLFREQ RESTRICTION Command (abbreviation 'ar')**

Summary: activate/deactivate dis.allfreq. restriction (MOD)  
Argument: <'on' or 'off'>  
Default: displays the current setting

The 'allfreq restriction' affects the MOD-score calculation. If the option is 'on', the disease allele frequency is constrained to be not greater than the value specified by the 'highest allfreq' command (which defaults to 0.5). This restriction can be turned 'off' in order to leave the disease allele frequency unconstrained.

When GENEHUNTER-MODSCORE is started, 'allfreq restriction' is 'on'.

### **HIGHEST ALLFREQ Command (abbreviation 'ha')**

Summary: select upper disease allfreq. bound for MOD analysis  
Argument: <highest disease allele frequency>  
Default: displays the current value

The 'highest allfreq' affects the MOD-score calculation. If the option 'allfreq restriction' is 'on' (which is the default), the disease allele frequency is constrained to be not greater than the value specified by the 'highest allfreq' command.

When GENEHUNTER-MODSCORE is started, 'highest allfreq' is set to 0.5.

### **DIMENSIONS Command**

Summary: define number of parameters jointly varied (MOD)  
Argument: <number of parameters>  
Default: displays the current setting

With the 'dimensions' command, the user can determine the number of parameters that are jointly varied during the fine maximization of a MOD-score analysis. The argument should be a number between 1, which means that only one parameter is varied at a time, and 5, such that up to five parameters are varied together, i.e., the disease allele frequency and all four penetrances in case of 'imprinting on'. The default is 2; specifying a higher value leads to a more thorough maximization at the price of a larger number of disease-locus likelihood evaluations and, hence, more computing time. However, the demands should still be moderate if the dataset does not contain too large pedigrees.

### **SAVED MODELS Command**

Summary: define number of saved trait models (modcalc single)  
Argument: <number of trait models>  
Default: displays the current setting

The 'saved models' option is only used in the context of the classic calculation mode ('algebraic calculation off'). No model saving will be done with 'algebraic calculation' turned 'on' (which is the default) because the saving of scoring-function vectors, as described below, does not lead to a further speed-up in the context of the algebraic algorithm.

When using the 'modcalc single' option for an analysis with the 'modscore' command, a separate round of maximization is done for each genetic position. Thus, the scoring-function vectors have to be evaluated more often than with 'modcalc global', and the computation time roughly increases by a factor of the number of genetic positions. However, a scoring-function vector for given trait-model parameters does not depend on the genetic position. Thus, if sufficient memory is available, the scoring-function vectors can be saved when calculated for the first time and recalled for a later position, which avoids time-consuming recomputation. In order to achieve this, GENEHUNTER-MODSCORE applies a smart model-saving scheme. As the default, the scoring-function vectors of all initial models, i.e., predefined models and user-defined models, if there are any, will be stored in memory.

For the 'maximization standard' and 'small' option the numbers of initially tested models are as follows:

```
81 trait models with 'imprinting off' and 'penetrance restriction on',
135 trait models with 'imprinting off' and 'penetrance restriction off',
261 trait models with 'imprinting on' and 'penetrance restriction on',
567 trait models with 'imprinting on' and 'penetrance restriction off'.
```

For the 'maximization dense' and 'grid' option the numbers of initially tested models are as follows:

```
236 trait models with 'imprinting off' and 'penetrance restriction on',
848 trait models with 'imprinting off' and 'penetrance restriction off',
988 trait models with 'imprinting on' and 'penetrance restriction on',
4928 trait models with 'imprinting on' and 'penetrance restriction off'
```

- in any case, plus the number of user-defined trait models added with the 'model' command (and, in case of more than one liability class or if the maximum LOD over the initial models is below zero, a further model with all penetrances equal for the currently optimized liability class).

It is possible to change the number of models for which the scoring-function vectors will be stored in memory with the 'saved models' command. The argument should be a number between 0 (i.e., no model saving at all) and 32767. 'saved models -1' returns to the default of saving all initial models. The amount of memory needed per model crucially depends on the number and size of pedigrees in the dataset. If 'saved models' is called after 'scan pedigrees', the command will print the amount of memory needed, for the current dataset, to save the selected number of trait models. This allows for an adjustment of the number of saved models according to the total memory available on the system, before executing the 'modscore' command.

The initially tested models are stored with highest priority, since they are needed for every position, and thus the benefit will be maximal. If the number of 'saved models' is higher than the number of initially tested models, it is possible to save models tested during fine maximization as well (if 'maximization' is set to 'standard' or 'dense', since with the other options a fine maximization is not conducted). These models will generally not be the same for different positions. Still, it can be expected that the best-fitting trait-model parameters will be similar for consecutive positions, and hence, some of the fine-maximization models may have already been calculated before. It is more likely that a model checked during fine maximization has already been tested at one of the preceding positions, rather than several more positions ago. Therefore, when the number of models that can be stored is exceeded at some point, the program will drop the oldest fine-maximization model, which has been calculated for a position further back, and save the current model at its place (first-in-first-out strategy). This procedure increases the probability that a model checked during fine maximization is still available in memory from a previous calculation.

Please note that the initially tested models will never be overwritten, since they are always used for every position. (However, in case of several liability classes and when the 'modscore' command has already been executed before, the penetrances of the previously optimized liability class(es) will most likely not be the same for different genetic positions. Therefore, in that case, even the initial models will differ between genetic positions, and accordingly they are not stored in memory with higher priority than the models tested during fine maximization.)

For datasets with several large pedigrees, a substantial amount of memory can be required for the saving of the scoring-function vectors. However, independent of the used 'maximization' option, it is always worthwhile trying to save the scoring function at least for the initially tested models when using the 'modcalc single' option, because this cuts down the computation time by 30 to 50 percent or substantially more, depending on the chosen 'maximization' option. This is even the case if swap memory has to be used to a certain degree, since the paging activity is kept at a minimum.

When using the 'modscore' command in conjunction with the 'calculate p value' command, many trait models (at least all of the user-defined and predefined models) will not change from one replicate to another with the simulation routine. The stored scoring-functions can therefore even be reused for different replicates. In contrast to a MOD-score analysis without computing p-values, this holds also for the 'modcalc global' option. Hence, when p-values are calculated (and only in this case), scoring-functions will also be stored for 'modcalc global'.

#### **LONG MOD OUTPUT Command (abbreviation 'lm')**

Summary: activate/deactivate long output (modcalc single)  
Argument: <'on' or 'off'>  
Default: displays the current setting

When using the 'modcalc single' mode, a separate maximization is done for each genetic position when executing the 'modscore' command. As default, only the MOD score and parameters of the best-fitting model are printed for each position ('long mod output off'). In order to obtain the output of all tested trait models for each genetic position, please turn 'long mod output on'. This will lead to files with length of several megabytes.

With 'modcalc global', the 'modscore' command always prints the parameters of all tested models and the results during the course of the analysis, irrespective of the 'long mod output' option.

#### **NORMALIZE ALLFREQ Command (abbreviation 'naf')**

Summary: activate/deactivate marker allfreq normalization  
Argument: <'on', 'off' or 'old'>  
Default: displays the current setting

Turning the 'normalize allfreq' option 'on' induces GENEHUNTER-MODSCORE to normalize the marker allele frequencies so that they sum to 1.0 at each marker. This option should be used if marker allele frequencies specified in the locus datafile do not sum to 1.0 for all markers, since in some cases (e.g. when there are untyped founders) unnormalized marker allele frequencies can lead to wrong results.

With the default setting 'normalize allfreq off', a normalization is not

performed. In this case, the analysis will be stopped if the allele frequencies at a marker sum to a value  $\leq 0.9$  or  $\geq 1.1$ . Please check the values given in the locus datafile, and consider turning 'normalize allfreq on', if this happens.

If 'normalize allfreq' is set to 'old', the analysis will be performed even when marker allele frequencies sum to a value  $\leq 0.9$  or  $\geq 1.1$ , without a normalization. This is in accordance with previous program versions.

The 'normalize allfreq' option has to be specified before calling the 'load markers' command. Once the markers have been loaded, changes will be ignored.

#### **NUMBER OF REPLICATES Command (abbreviation 'nor')**

Summary: defines number of replicates used for cpv  
Argument: <'on' or 'off'>  
Default: displays the current setting

Calculation of empirical p-values involves simulating and analyzing a number of replicates n. This number is crucial for the accuracy of empirical p-values. If p-values are small, using an insufficient number of replicates leads to a decrease in accuracy. On the other hand, with larger p-values, using many replicates may unnecessarily increase the computation time.

Unfortunately, prior to the analysis one does not know the p-value - and thus, how many replicates will be necessary to obtain sufficient accuracy. However, given a certain significance level it is possible to determine the required number of replicates. For instance, if p-values around  $10^{-3}$  should be determined empirically, at least 10,000-50,000 replicates are needed (such that about 10 out of 10,000 replicates have a score at least as high as the real data set) - or larger numbers if smaller p-values should be accurately determined. With large pedigrees in the sample, it may sometimes be infeasible to compute a sufficient number of replicates. Even in that case one should try to compute at least a smaller number of replicates in order to obtain a rough guess of the p-value.

The number of replicates used to derive the empirical p-values defined by this option must be an integer greater than zero. Please keep in mind that the 'number of replicates' needs to be defined before executing the 'scan pedigrees' command.

For further details on efficient computation of p-values please see the help text for the 'sequential simulation' and 'best position' options. More general information on empirical p-value calculation is given in the help text of the 'calculate p value' command.

In case one does not wish to calculate p-values, 'number of replicates' should be '0' (which is the default setting) when 'scan pedigrees' is called.

## SEQUENTIAL SIMULATION Command (abbreviation 'seq')

Summary: activate/deactivate sequential simulation mode  
Argument: <'0' or integer greater than zero>  
Default: displays the current setting

Typically empirical p-values are computed by using a fixed number  $n$  of replicates (for further information see 'number of replicates' option) and counting the replicates  $r$  that surpass the test statistic of the real data set. Besag and Clifford (Biometrika 78:301-304, 1991) proposed a method that makes use of fixing a target number,  $h$ , of replicates that have to exceed the observed test statistic. By this approach, replicates of the data are simulated sequentially until this target number has been reached. The estimated p-value is  $h/n$ , where now  $n$  is random and  $h$  is fixed. When the empirical p-value is large (i.e., when the study results are nonsignificant),  $n$  is small, and the process finishes quickly. When the p-value is small,  $n$  is larger, and more time is invested.

Upon activation of the 'sequential simulation' option, with the target number  $h$  of replicates given as argument, the procedure proposed by Besag and Clifford will be followed. The number  $n$  determined by the 'number of replicates' option fixes the maximum number of replicates used for deriving the empirical p-values. This is necessary to make sure that the computation ends within a reasonable amount of time, even for very small p-values.

When using the 'modcalc single' option, the calculation at a particular genetic locus will be stopped after the fixed number  $h$  of replicates, that exceed the test statistic of the real data set at that locus, is reached. Furthermore, with 'modcalc single' the p-value for the overall maximum MOD score (over all loci) of the real data will not be provided with 'sequential simulation' activated. For details on the 'region-wide' p-value for the overall maximum MOD score please see the help text of the 'calculate p value' command.

In case that 'sequential simulation' is used together with 'modcalc global', replicates will be simulated until the overall maximum MOD score (over all loci) in the real data set is exceeded by the overall maximum score of the specified number  $h$  of replicates. Please keep in mind that this p-value is still 'region-wide', and further corrections have to be done for the assessment of genome-wide significance in the context of a whole-genome scan.

Is a 'single-model' LOD-score analysis performed with the 'sequential simulation' option activated, the simulation process will proceed as long as the overall real data maximum score is exceeded by  $h$  replicates and the real data score is exceeded by  $h$  replicates at each of the positions.

'Sequential simulation' can be deactivated with the argument '0'. Any integer lower than zero will lead to a fixed  $h$  of '50', any integer greater than zero will be used as fixed  $h$ .

'Sequential simulation' is off (= '0') when GENEHUNTER-MODSCORE is started.

Please see also the help texts of the related commands 'number of replicates', 'calculate p value', 'modcalc', and 'best position'.

### **STORE REPLICATES Command (abbreviation 'str')**

Summary: activate/deactivate storage of replicate information  
Argument: <'pre', 'both' or 'off'>  
Default: displays the current setting

The replicates generated to determine empirical p-values for MOD and LOD scores can be saved to files in Linkage (pre-Makeped) format by activating the 'store replicates pre' option, so that they can be reused at a later time. In order to gain some information on the results for each replicate as well, please use the 'store replicates both' option. In this case, in addition to the pedigree file (with extension \*.pre) a second file (with extension \*.out) containing the results for the respective replicates will be created. The files are named repli<number> plus the file extension with <number> denoting the index of a replicate, ranging from 1 to n (see 'number of replicates' help text regarding the specification of n).

Please keep in mind that generating the aforementioned files for a large number of replicates requires sufficient amount of free disk space in your working directory.

'Store replicates' is 'off' when GENEHUNTER-MODSCORE is started.

### **UNTYPED FOUNDERS Command (abbreviation 'ufo')**

Summary: activate/deactivate usage of untyped founders (cpv)  
Argument: <'on' or 'off'>  
Default: displays the current setting

The simulation routine of GENEHUNTER-MODSCORE generates replicates with respect to the 'genotype patterns' of the real data set. This means that a genotype that is missing in the real data set ('0 0') for a certain individual and marker will remain a missing genotype in the replicates, and equivalent for the 'non-zero' genotypes. In some cases one may want to use replicates with founders that are completely untyped for all markers. With 'untyped founders on' the simulated alleles for the founders are set to zero after the non-founders' genotypes have been derived.

Please see also the help texts of the related commands 'full information', 'calculate p value', and 'simulate untyped'.

'Untyped founders' is 'off' when GENEHUNTER-MODSCORE is started.

### **FULL INFORMATION Command (abbreviation 'fin')**

Summary: activate/deactivate fully informative markers (cpv)  
Argument: <'on' or 'off'>  
Default: displays the current setting

The simulation of founder genotypes is usually conditional on the number of marker alleles and the allele frequencies. These parameters determine the information content at a specific marker. For instance, if one or more founders are homozygous, information content at that particular marker is reduced. This will be automatically taken into account when calculating empirical p-values for real data sets with GENEHUNTER-MODSCORE, provided that realistic marker allele frequencies have been specified in the locus datafile. However if one wishes to generate replicates with fully informative markers, 'full information' has to be activated. In this case, the alleles for the founders in each

family are assigned from 1 to n. If the 'untyped founders' option is deactivated and the 'simulate untyped' option is activated, this assures fully informative markers for every simulated replicate.

Please see also the help texts of the related commands 'untyped founders', 'calculate p value', and 'simulate untyped'.

'Full information' is 'off' when GENEHUNTER-MODSCORE is started.

#### **SIMULATE UNTYPED Command (abbreviation 'sun')**

Summary: activate/deactivate simulation of missing genotypes  
Argument: <'on' or 'off'>  
Default: displays the current setting

The simulation routine of GENEHUNTER-MODSCORE generates replicates with respect to the 'genotype patterns' of the real data set. This means that a genotype that is missing in the real data set ('0 0') for a certain individual and marker will remain a missing genotype in the replicates, and equivalent for the 'non-zero' genotypes. In order to generate replicates with no missing genotypes, the 'simulate untyped' option should be turned 'on'. With this setting, genotypes will be given for every marker, including those individuals that are fully or partly ungenotyped in the real data set.

Please see also the help texts of the related commands 'full information', 'calculate p value', and 'untyped founders'.

'Simulate untyped' is 'off' when GENEHUNTER-MODSCORE is started.

#### **BEST POSITION Command (abbreviation 'bep')**

Summary: perform p-value calc. for the best position only  
Argument: <'on' or 'off'>  
Default: displays the current setting

Especially in the context of a 'modcalc single' MOD-score analysis, computation time is a limiting factor for the calculation of empirical p-values with a sufficient number of replicates. There are two options implemented in GENEHUNTER-MODSCORE to restrict the computation time needed for calculating p-values. The first one is the 'sequential simulation' option (for more information please see the corresponding help text). The second one is the 'best position' option. When activated, a pointwise p-value will only be calculated for the position with the overall maximum MOD score in the real data set, ignoring all other positions. If there are two or more positions with the (same) highest score in the real data analysis, only the first position will be considered. In case the 'best position' option is activated, no 'region-wide' p-value will be calculated for the overall maximum score of the real data set.

It is only possible to use 'best position' with 'modcalc single'.  
'Best position' is 'off' when GENEHUNTER-MODSCORE is started.

### SET RANDOM SEED Command (abbreviation 'srs')

Summary: sets the random seed for the Mersenne Twister  
Argument: <' -1 or integer that is greater or equal to 0 >  
Default: displays the current setting

This command must be executed before any commands that use random numbers, for example 'calculate p value'. Otherwise, random numbers generated before the 'set random seed' command will be seeded using the system time.

The calculation of empirical p-values requires the usage of a pseudo-random number generator (in case of GENEHUNTER-MODSCORE, the Mersenne Twister is employed). In this context, the random seed is a number used to initialize the pseudo-random number generator. With the same random seed, two different runs lead to the same series of pseudo-random numbers and therefore to identical results. Normally this behaviour is avoided by using the system time as a more or less 'random' random seed; this procedure assures different pseudo-random numbers for two consecutive runs. If one wishes, for example, to compare the results of an analysis with and without 'imprinting', the 'set random seed' option can be used to perform both runs using the same set of simulated replicates, i.e., with genotypes being identical for both simulations.

With 'set random seed -1' (which is the default), the random seed is derived from the system time. Any other integer lower than zero will be ignored, any integer greater than or equal to zero will be used as the random seed.

### SHOW DISTRIBUTION Command (abbreviation 'sdi')

Summary: activate/deactivate displaying distribution info  
Argument: <'on' or 'off' >  
Default: displays the current setting

Sometimes it is not only of interest how many replicates exceeded the score of the real data set, but also the overall distribution of the results for the replicates. Activating the 'show distribution' option provides a brief summary of the score distribution for each genetic position, using intervals of width 0.1, starting with [0.00; 0.10[. The first column of the summary gives the number of replicates with a score in the particular interval, the next column gives the number of replicates exceeding the lower boundary of the interval. Finally the p-value for the lower boundary is provided.

If the 'best position' option is used in order to limit computation time (please see corresponding help text), and the 'show distribution' option is activated, researchers may use the score distribution at the position with the highest linkage peak to obtain an idea about p-values of scores at other locations, without having to perform an analysis of all replicates for every genetic position. This is only possible with 'modcalc single' or a standard LOD-score analysis (see also below). Please note, however, that p-values derived in this way will only be a rough estimate, since the distribution of linkage scores at a locus depends on genetic distances to flanking markers as well as on marker information content.

Simulated p-values for all positions (i.e., if the 'best position' option is deactivated) will be provided only with 'modcalc single' or a standard LOD-score analysis. In case 'show distribution' is used together with 'modcalc global', the distribution is only given for the overall best MOD score in the real data set (i.e., the maximum over all positions). Hence, p values corresponding to the boundaries of MOD-score intervals can only be used to judge other region-wide MOD

scores (e.g. maxima on different chromosomes), but not scores at specific positions in the real data set.

It is not possible to use the 'show distribution' option together with the 'sequential simulation' option. 'Show distribution' is 'off' when GENEHUNTER-MODSCORE is started.

#### **SINGLE POINT Command**

Summary: activate/deactivate single-point analysis  
Argument: <'on' or 'off'>  
Default: displays the current setting

Turning the 'single point' option on instructs subsequent 'scan' and 'total' commands to calculate and display single-point LOD and NPL scores for each marker in the data set individually rather than the usual multi-point analysis. The position of the disease locus is assumed to be identical to the marker position (i.e., the recombination fraction is fixed to zero), using single marker information only. This command will ignore the linkage map set with the 'use' command and will not produce haplotype output or recombination counts for obvious reasons.

'Single point' is 'off' when GENEHUNTER-MODSCORE is initiated, and it must remain 'off' for a MOD-score analysis and a p-value calculation.

#### **COUNT RECS Command**

Summary: turn recombination counting on  
Argument: <'on' or 'off'>  
Default: displays the current setting

Turning this option on activates the recombination-counting mechanism in the "scan" command. After each pedigree is scanned, the observed recombinations (and resulting distances) are shown for each map interval alongside the actual distance of the interval. When there are significantly more recombinants than expected in an interval or set of intervals, this can often indicate an error or errors in the genotype data.

At the end of the scan of multiple pedigrees, the overall count of recombinants in each interval is displayed along with the expected value for the entire data set. Recombination counts significantly higher than expected here can be an indication of a marker that is error-prone over multiple pedigrees or of an error in the entered genetic map (either in order or distance).

When an analysis with sex-specific genetic distances is performed, the sex-specific values will be appropriately used for calculation of the numbers of expected and observed recombinations. These numbers will be reported, for each inter-marker interval, combined for male and female meioses.

'Count recs' is OFF when GENEHUNTER-MODSCORE is started.

### HAPLOTYPE Command

Summary: determine likely haplotypes for individuals  
Argument: <'on' or 'off'>  
Default: displays the current setting

When the 'haplotype' option is turned on, the 'scan' command will report the most likely inferences made regarding the haplotypes of the individuals in each pedigree. The haplotypes for founders will be displayed on the screen and the haplotypes for all individuals analyzed will be stored in a file called haplo.dump. In addition, if the 'postscript output' option is 'on', the entire pedigree (with haplotypes and recombinations indicated) will be drawn in a postscript file suitable for printing and displaying.

The haplotypes displayed represent the maximum-likelihood set of inheritance vectors that explain the data. After all markers have been scanned in a pedigree the most likely path through all of the markers is recreated - thus yielding the most likely pattern of inheritance at each marker and likely positions of recombinants. Among nearby markers that show no recombination, these haplotypes are usually unambiguous, but in cases where recombinants are present (especially in small sibships of 2 or 3 individuals), the haplotypes may be imperfect and simply represent the most likely choice out of several valid choices. For example, the most likely position of recombinants is shown in the PostScript output but other placements may be possible but simply less likely due to considerations of map interval size and allele frequency at certain markers.

Haplotypes can be invaluable tools both analytically (in searching for shared genomic regions of distantly related affected individuals and indicating linkage disequilibrium between markers) and practically (in searching for errors in genotyping which usually manifest themselves as excessive obligate recombination in an individual or pedigree). In cases where two original parents are both untyped for all loci, haplotypes will be displayed for them as usual but it must be noted that the assignments could be reversed (i.e., the two haplotypes assigned to the original father could actually belong to the original mother and vice-versa).

N.B. - at this time the drawing code is not yet complete and while nearly complete, certain pedigree structures (such as those containing marriage loops, inbreeding loops, or individuals with many spouses) may not always be drawn properly. Refer to the results in the haplo.dump file if it appears the pedigree has not been drawn properly.

The MaxProb haplotyping approach (please see the 'haplotype method' command for details) appropriately takes sex-specific genetic distances into account when specified for an analysis, whereas the Viterbi algorithm will always rely on the sex-averaged coordinates.

'Haplotype' is OFF when GENEHUNTER-MODSCORE is started.

### DISCARD Command

Summary: eliminate less informative individuals  
Argument: <'on' or 'off'>  
Default: displays the current setting

As noted in the "scan" command, some larger pedigrees can be quite time consuming to analyze. To speed this up, some less informative individuals can be discarded without significant loss of information. When the "discard" option is turned on, unaffected individuals that

have no descendents in the pedigree and have informative parents (i.e., genotyped) are discarded from analysis. This will alter results somewhat (LOD scores more than NPL statistics since the unaffected individuals are not considered in NPL statistics which measure the degree of sharing among affected individuals) and should only be used if you are interested in obtaining a fast approximation of the results or if your pedigrees are extremely large and cannot be fully analyzed by GENEHUNTER-MODSCORE.

#### **MAX BITS Command (abbreviation 'mb')**

Summary: determine how large a pedigree may be analyzed  
Argument: <number of bits>  
Default: displays the current setting

Because of the time and memory requirements of the mapping algorithms in GENEHUNTER-MODSCORE, a maximum pedigree size must be set to keep the computations within the ability of the computer it is running on. The memory and time required increase exponentially with the number of bits in the inheritance vector (number of meioses being examined). This number is  $2N - F$  where  $F$  is the number of founders in the pedigree and  $N$  is the number of non-founders. For example, a pedigree consisting of two parents and their 4 children would have a size =  $2N - F = 6$ . Entirely uninformative individuals such as individuals in the last generation of a pedigree that are ungenotyped are not included in this figure as they will not be analyzed.

On most workstations, setting the value to 19 or 20 will be a reasonable limit. If pedigrees exceed the size that may be computed under the current 'max bits' setting, individuals may be dropped or the pedigree may be skipped (depending on the setting of 'skip large' - see below). The default setting of 'max bits' is 20.

#### **SKIP LARGE Command**

Summary: determine how large pedigrees are dealt with  
Argument: <'on' or 'off'>  
Default: displays the current setting

Because of the memory and time limitations described in the 'max bits' section, certain pedigrees may not be able to be computed. In this instance a warning message is displayed and one of two things will happen:

- if 'skip large' is ON - the pedigree will be skipped over entirely and the computation will continue with the next pedigree in the data set
- if 'skip large' is OFF - pedigree individuals will be trimmed off until the pedigree is small enough to be analyzed within the current setting of 'max bits'. This trimming is done such that the maximum amount of linkage information is retained - the first individuals to be eliminated will be unaffected individuals at the bottom of the pedigree as these individuals add very little to the NPL statistic (which measures sharing among affected individuals) and will affect the LOD score somewhat depending on the proposed penetrance of the disease allele.

In either case, it is recommended that for very large pedigrees (where a large number of individuals are not being analyzed) you consider

dividing the pedigree into two or more reasonably sized pedigrees that can be analyzed in full.

### **ANALYSIS Command**

Summary: select what type of linkage analysis to perform  
Argument: <'NPL', 'LOD', or 'BOTH'>  
Default: displays the current setting

The 'analysis' command allows the user to select the method of linkage analysis employed by the scan command. One may select one of three options:

NPL: the 'scan' and 'total' commands will produce only the non-parametric sharing statistics

LOD: the 'scan' and 'total' commands will produce only parametric LOD scores based on the model specified in the locus information file

BOTH: both NPL and LOD scores will be produced

The 'analysis' option is set to BOTH when GENEHUNTER-MODSCORE is started. When a MOD-score analysis should be performed, 'analysis' must either be set to LOD or BOTH.

### **SCORE Command**

Summary: select NPL scoring function  
Argument: <'pairs', 'all', or 'hom'>  
Default: displays the current setting

The 'score' command allows the user to select the NPL scoring function to be used during analysis with the 'scan' command. These functions offer a measurement of the degree of sharing among affected individuals and are not dependent on the specific model proposed for the disease as the parametric LOD score is. The statistic reported will represent the deviation from Mendelian expectation observed and will roughly follow the normal distribution.

The 'pairs' function computes a score based on the degree of sharing among all pairs of affected individuals in a pedigree. This statistic is similar to those used in non-parametric sib-pair or APM analyses.

The 'all' function examines all individuals simultaneously and assigns a higher score when more of them share the same allele by descent. It is our experience in extensive simulations and analysis of real pedigree data that the 'all' statistic provides a more powerful test.

NEW: The 'hom' function is similar to 'all', i.e., all affected individuals are examined simultaneously. Whereas the 'all' function chooses one out of two alleles for each of k affected individuals and calculates the average score over all possible choices, the 'hom' function looks at both alleles at the same time. The score is then calculated as the 'all' score for these 2k alleles, with no need for taking the average over many choices. In other words, each of the distinct founder alleles contributes a factor, i.e., the factorial of its frequency within the set of the affected individuals' alleles, to the overall score. Compared to 'all', the 'hom' function allows much faster NPL score calculation, which is especially convenient when analyzing larger pedigrees.

### **POSTSCRIPT OUTPUT Command (abbreviation 'ps')**

Summary: activate Postscript graphing capability  
Argument: <'on' or 'off'>  
Default: displays the current setting

When the "postscript output" option is turned on, the "total stat" command will prompt the user for filenames in which to store postscript graphs for total LOD score, total NPL statistic, and total information content. These files are ready for printing on any Postscript printer and can be displayed by many screen browsers such as Ghostscript. In addition, if the 'haplotype' option is 'on', the scan command will produce pedigree drawings with most likely haplotypes of original individuals and most likely placements of recombinations. Furthermore, when a MOD-score analysis is performed under the "modcalc single" option, a postscript graph is created that shows the MOD score in conjunction with the dominance index D and imprinting index I for the best-fitting model at every genetic position. Please see the "modscore" command for details.

If a sex-specific genetic map is used, the sex-averaged coordinate calculated as the mean of the corresponding female and male genetic position will be used as x-axis coordinate in the Postscript plots.

### **LETTERS Command**

Summary: controls allele display in Postscript output  
Argument: <'on' or 'off'>  
Default: displays the current setting

When the 'haplotype' and 'postscript' options are both turned on, the 'scan' command produces postscript pedigree drawings with the most likely haplotypes of original individuals displayed. If 'letters' is on, these haplotypes will be drawn as letters representing the founder chromosome inherited rather than the numeric genotypes themselves. Upon startup of Genehunter, 'letters' is off and these drawings will display the actual alleles inherited.

### **DRAWING SCALE Command (abbreviation 'ds')**

Summary: set scale of Postscript 'total' drawings

The 'drawing scale' command allows the user to select the type of scaling used to draw the total NPL, LOD, and information content pictures during the 'total' command, as well as the MOD-score plots when using the 'modcalc single' option. The two options are to have the genetic map (along the x-axis) fill the page, or to set a constant numeric scale in dots per cM (always reflecting the sex-averaged coordinate). The latter option may be used if you are interested in having the same scale used among different runs of GENEHUNTER-MODSCORE for later comparison of output. There are roughly 650 dots available for drawing so a good choice for scale would be roughly 650/(length of largest chromosome). By default, the Postscript drawings will fill the page.

### **TITLE Command**

Summary: set title of PostScript plots  
No Arguments

By default, the pedigree filename is used as the title when scores are plotted as a function of the genetic position in a PostScript graph. The 'title' command can be used to specify a different title. In order to allow for strings that include whitespace, the title is not given as a command argument; rather, the user will be prompted to enter the title on a separate line after calling the 'title' command. If only <Return> is entered here, the pedigree filename will be used as the title of the graph.

### **OFF END Command**

Summary: Select how far to compute scores beyond ends of map  
Argument: <distance>  
Default: displays the current value

This command controls how far before the first marker and after the last marker in a map scores will be calculated. For example, if off-end is set to 10.0, then subsequent scan commands will begin calculating scores 10 cM before the first marker and continue stepping through until 10 cM after the last marker. The default value of 'off end' is 0.0 cM. Calling 'off end' with no arguments causes GENEHUNTER-MODSCORE to report the current value.

Distances may be specified as either recombination-fractions or centiMorgans, with the necessary assumption that any distance below 0.5 is assumed to be a recombination-fraction and any greater than or equal to 0.5 is assumed to be in centiMorgans.

With sex-specific genetic coordinates, the 'off end' distance parameter is interpreted as a sex-averaged coordinate (i.e., the mean of the female and male positions); however, sex-specific genetic distances are still used. If a map file has been specified by the 'read map' command, the female/male ratio of genetic distances in the off-end range (i.e., before the first and after the last typed marker) is calculated from additional untyped markers, located in these regions, that can be found in the map file. If the off-end range reaches beyond the genetic region with marker coverage (as found in the map file), the overall female/male distance ratio for the entire chromosome (as determined by the first and last marker for that chromosome given in the map file) will be used for these outer portions of the off-end range. In case that no map file has been specified, the overall female/male distance ratio, as determined by the first and last marker given in the locus datafile, will be employed for the entire off-end range at both ends of the marker group.

### **INCREMENT Command**

Summary: Choose the scan step size  
Arguments: <'distance' or 'step'> <number>

If 'increment distance 2.0' is entered, the 'scan' command will calculate LODscores and NPL statistics every 2.0 cM throughout the genetic map selected (regardless of the position of markers in that map) as follows (in this example the off end distance is set to 6.0 cM):

-6.0 (6 cM before the first marker), -4.0, -2.0, 0.0 (the position of the first marker), 2.0, 4.0, ...etc...until 6.0 cM after the last locus.

If 'increment step 5' is selected, the scan command will calculate scores at 5 equally spaced positions between each marker. For example, with a three-locus map with 10 and 15 cM intervals and 'off-end' set to 5.0 cM, maps will be computed at the following positions:

-5.0, -4.0, -3.0, -2.0, -1.0 (equally spaced in the 5cM before the first marker)  
0.0, 2.0, 4.0, 6.0, 8.0 (equally spaced in the 10 cM interval)  
10.0, 13.0, 16.0, 19.0, 22.0 (equally spaced in the 15 cM interval)  
25.0, 26.0, 27.0, 28.0, 29.0, 30.0 (equally spaced in the 5cM after the map)

The default value of 'increment' is 'step 5'. Calling 'increment' with no arguments causes GENEHUNTER-MODSCORE to report the current value.

Note that the first ('distance') method is not guaranteed to hit every marker position and should be considered inferior to the second ('step') method, which will compute a map at every marker position.

With sex-specific genetic coordinates, if the 'increment distance' method is used, the distance parameter is interpreted as a sex-averaged coordinate (i.e., the mean of the female and male positions); however, sex-specific genetic distances are still employed. With both the 'step' and 'distance' method, the coordinates of additional ungenotyped markers given in a map file (specified by the 'read map' command) will be automatically used to determine the sex-specific genetic coordinates at which the linkage statistics should be calculated. In particular, using the coordinates of additional untyped markers allows for appropriately varying the female/male distance ratio even between two genotyped markers.

#### **MAP FUNCTION Command**

Summary: Choose a cM <-> rec-frac conversion function  
Argument: <'haldane' or 'kosambi'>  
Default: displays the current value

This command controls which mapping function is used to convert centiMorgans to recombination-fractions and back again both in the input and output of the program and in the internal calculations. Currently only Haldane and Kosambi map functions are available. The default 'map function' is Haldane.

#### **UNITS Command**

Summary: Choose whether scan output is in cM or rec-frac  
Argument: <'cM' or 'rec-frac'>  
Default: displays the current setting

The 'units' command enables the user to select whether the output from the 'scan' command appears in recombination-fractions (rf) or centiMorgan distance (cM). The conversion function for centiMorgans to recombination fractions can be set using the 'map function' command. When GENEHUNTER-MODSCORE is started up, Haldane centiMorgans are selected as output units.

### **DISPLAY SCORES Command**

Summary: activate screen display of scores and haplotypes  
Argument: <'on' or 'off'>  
Default: displays the current setting

If 'display scores' is 'on', GENEHUNTER-MODSCORE will display the scores and haplotypes for each pedigree ('scan pedigrees' and 'modscore'). 'Display scores' is ON when GENEHUNTER-MODSCORE is started.

### **COMPUTE SHARING Command (abbreviation 'cs')**

Summary: turn IBD matrix storage on/off  
Argument: <'on' or 'off'>  
Default: displays the current setting

While scanning a pedigree, if 'compute sharing' is turned 'on', the program accumulates the IBD inheritance probabilities (IBD matrix) for all pairs of relatives in the pedigree. The IBD matrix is used for the computation of TDT scores, sib pair statistics and variance components analysis. If such analyses are not required, the user should turn off storage of the IBD matrix using the command compute sharing off. In large collections of highly informative pedigrees, storage of the IBD matrix is the dominant memory requirement.

Please note: 'compute sharing off' is the default setting for GENEHUNTER-MODSCORE; it is appropriate if only LOD, NPL, and MOD scores should be computed, and saves both computation time and memory.

### **DUMP REQUIREMENTS Command**

Summary: estimate memory usage instead of scanning pedigree  
Argument: <'on' or 'off'>  
Default: displays the current setting

This command sets a flag that modifies the behavior of the program. When the flag is set, scanning a pedigree simply reports the memory requirements. The researcher can add/subtract pedigree members or modify the marker map and get immediate feedback before proceeding with the full analysis. Time requirements are not reported, but in the absence of inbreeding loops, it is safe to assume that time and memory requirements scale proportionally. We report only memory allocation that scales exponentially with pedigree size and we do not report requirements for storage of the IBD matrix. Please note: this feature does not reflect the computational requirements of a MOD-score analysis or p-value calculation with GENEHUNTER-MODSCORE. The amount of additional memory required for model-vector saving in a MOD-score analysis can be determined by executing the 'saved models' command after 'scan pedigrees'. Here, the 'dump requirements' option should be left 'off'. In case of p-value calculation the 'dump requirements' option will be ignored.

### HAPLOTYPE METHOD Command

Summary: choose algorithm used for haplotyping  
Argument: <'Viterbi' or 'MaxProb'>  
Default: displays the current setting

There are two standard solutions to haplotype reconstruction. One is based on selecting the most likely inheritance vector at each locus and the other (Viterbi algorithm) selects the most likely set of vectors considering all loci simultaneously. While the second solution has theoretical appeal because it finds a global maximum, in practice both methods yield similar results; especially when used for the analysis of pedigrees with high information content. Currently only the first approach (MaxProb) takes advantage of space reduction. With this approach, haplotype reconstruction adds a few percent to the overall computational time and does not add to the memory requirements.

The MaxProb haplotyping approach appropriately takes sex-specific genetic distances into account when specified for an analysis, whereas the Viterbi algorithm will always rely on the sex-averaged coordinates.

### (3) SIBS QUALITATIVE TRAIT MAPPING COMMANDS

Commands to map loci using affectation status.

#### ESTIMATE Command

Summary: maximum likelihood estimate of IBD sharing  
No Arguments

Usage -- To run the command just type 'estimate'; no arguments are needed. GENEHUNTER(-MODSCORE) will first ask you if you want to analyze your data under the assumption of no dominance variance, or under the assumption of dominance variance where Holmans' triangle is applied:

```
analyze under the assumption of no dominance variance? y/n [n]
```

The default is to perform the analysis with the assumption of dominance variance.

GENEHUNTER(-MODSCORE) will then query you for the filenames to store the text and postscript output respectively. You will be alerted if either of the chosen filenames already exist.

Output -- The text file consists of the columns:  
position <z0> <z1> <z2> <loglike>

where the z-values are the calculated maximum likelihood proportions. At the end of the text output is a time-stamped summary of the session settings when the analysis was run. The first postscript output file is a graph of position vs. loglike and the second is a plot of how the maximum likelihood sharing proportions change across the region. The marker names are given along the x-axis and the distance examined in the analysis is given at end of the x-axis (this may be larger than the map distance if you have specified an off-end distance).

Background -- 'estimate' scans the selected map region and identifies regions of significant excess allele sharing.

Note that the LOD score is never negative, because the maximum likelihood solution for z0, z1, and z2 can never be worse than the Mendelian segregation expectation.

#### EXCLUDE Command

Summary: exclusion mapping  
Arguments: <relative risk ratio hypotheses>

Usage -->  
command line:

```
npl:6> exclude
```

You are then given the option of inputting a set of z's or relative risk values (the input queries will be different depending on whether you want to analyze your data under the assumption of no dominance variance or not).

Output --> The text file consists of tabbed columns in the format:

```
position z2-1 z2-2 z2-3 ... etc.
```

(You should be able to use this file as input to a plotting program if you don't have access to a postscript printer.) At the end of the text file is a time-stamped summary of the session settings. The postscript file consists of multiple y-axis LOD score plots for each relative risk value/set of z's and gives the distance examined in the analysis at end of the x-axis (this may be larger than the map distance if you have specified an off-end distance). A horizontal dashed line is drawn at the traditional exclusion criterion of  $Z < -2$ .

Background --> Exclusion mapping is used to identify and exclude regions unlikely to have a major effect on the trait you are mapping. GENEHUNTER(-MODSCORE) does this by comparing the likelihood of the observed sharing proportion of 0, 1 and 2 alleles between affected sibs ( $z_0, z_1, z_2$ ), to the likelihood under the Mendelian expectation of  $a_0=1/4$ ,  $a_1=1/2$ , and  $a_2=1/4$ . When using GENEHUNTER(-MODSCORE) under the assumption of no dominance the sharing proportions are given by:

$$\begin{aligned} z_0 &= a_0/L_s \\ z_1 &= a_1 \\ z_2 &= a_2((2L_s-1)/L_s) \end{aligned}$$

where  $L_s = \lambda_{\text{sub-S}}$ , the relative risk ratio for a sib, defined as:

$$\frac{\text{prevalence of the trait in siblings of affected individuals}}{\text{prevalence of the trait in the population at large}}$$

Note that  $L_s = 1$  when there is no observed difference in prevalence of sibs vs the population ( $z_0=a_0$ ,  $z_1=a_1$ ,  $z_2=a_2$  and  $\text{LOD} = 0$ ). If  $L_s < 1$ , it would imply that there was some protective advantage in having an affected sib. Since neither of these cases are interesting and/or reasonable, only  $L_s$  values  $> 1$  are allowed. (The no dominance variance assumption allows us to simplify the sharing proportions above to the one variable  $L_s$ . With dominance variance  $L_s = L_o$  where  $L_o = \text{relative risk ratio for an offspring}$ , and  $L_m-1=2(L_s-1)$  where  $L_m$  is the relative risk ratio for a monozygotic twin.)

The likelihood under Bayes theorem is:

$L(\text{pos}) = (z_0*p_0+z_1*p_1+z_2*p_2) / (a_0*p_0+a_1*p_1+a_2*p_2)$   
and the LOD score is calculated by summing  $\log_{10}(L(\text{pos}))$  across pedigrees for each position.

The relations for  $z_0$ ,  $z_1$ , and  $z_2$  above hold if multiple loci are involved in the trait, provided that the loci interact multiplicatively and the lambda values are defined as the component of the relative risk attributable to the locus.

More details on the analytical method are present in the publication

#### (4) SIBS QUANTITATIVE TRAIT LOCI (QTL) MAPPING COMMANDS

Commands to map loci using numerical phenotype scorings.

##### HASEMAN ELSTON Command

Summary:        traditional & EM Haseman-Elston analysis  
No Arguments

Usage -- After typing the command you will be queried as to which phenotype you want to analyze (if you have loaded more than one), and then queried for files to store the text output for the traditional haseman-elston and EM haseman-elston analyses, as well as the filename for the postscript output.

Output -- The traditional and EM haseman-elston output files have the columns:

    <position>        <beta>        <LOD>        <t>

At the bottom of each of these text output file is a time-stamped summary of the session variables when the command was run. This summary will also list which phenotype was selected and in the case of the EM algorithm, the convergence limit that was used. The postscript output file has a plot of both the traditional and EM results.

Note1: The EM algorithm has been found to have very rare instabilities in large intervals between markers; if there is a sudden peak in the EM plot make sure a similarly shaped peak also appears in the traditional haseman-elston results. (The nonparametric method does not have these instabilities either and can also be used to verify your results.)

Note2: In order to run this command you must have selected more than two pedigrees/pairs -- which shouldn't be a problem since it won't be very significant using any less!

##### ML VARIANCE Command

Summary:        maximum likelihood QTL variance estimation  
No Arguments

Usage -- After typing the command you will be queried as to which phenotype you want to analyze (if you have loaded more than one), and then queried for a files to store the text and postscript output.

Output -- The text output file has the format:

    <position>        <LOD>        <sigsq0>        <sigsql>        <sigsq2>

At the bottom of each of these text output file is a time-stamped summary of the session variables when the command was run. This summary will also list which phenotype was selected and the convergence limit that was used. The postscript output file is a plot of position vs. LOD.

Note: This EM-based algorithm has been found to have very rare instabilities in large intervals between markers; if there is a sudden peak in the plot you can verify it by checking it against the results of the nonparametric method, which is not subject to the same instabilities.

### NO DOM VAR Command

Summary:       maxlike QTL variance est. under no-dominance assmp.  
No Arguments

Usage -- After typing the command you will be queried as to which phenotype you want to analyze (if you have loaded more than one), and then queried for a files to store the text and postscript output.

Output -- The text output file has the format:

    <position>       <LOD>       <sigsq0>       <sigsq1>       <sigsq2>

At the bottom of each of these text output file is a time-stamped summary of the session variables when the command was run. This summary will also list which phenotype was selected and the convergence limit that was used. The postscript output file is a plot of position vs. LOD.

Note: This EM-based algorithm has been found to have very rare instabilities in large intervals between markers; if there is a sudden peak in the plot you can verify it by checking it against the results of the nonparametric method, which is not subject to the same instabilities.

### NONPARAMETRIC Command

Summary:       non-parametric QTL analysis  
No Arguments

Usage -- After typing the command you will be queried as to which phenotype you want to analyze (if you have loaded more than one), and then queried for a files to store the text and postscript output.

Output -- The text output file has the format:

    <position>       <Z-score>

At the bottom of each of these text output file is a time-stamped summary of the session variables when the command was run. This summary will also list which phenotype was selected and the convergence limit that was used. The postscript output file is a plot of position vs. Z-score.

## (5) OTHER SIBS COMMANDS

### PAIRS USED Command

Summary: select what pair combinations will be used  
No Arguments

If you have loaded more than two sibs in any of your sibships this command allows you to include the extra sibs in the analysis commands (all sibs are automatically included for phase information if parents are missing). When using 'all pairs', each pair is considered as an independent pedigree but a weight (2/num\_affecteds) is factored in to counteract inflation of significance due to the statistical dependence among these pairs.

Simply type 'pairs used' and indicate which pair setting you would like to use:

```
sibpair:1> pairs used
```

the current pair setting is: \*first affected/phenotyped sibpair only\*

Possible pair options:

1. First pair of affected/phenotyped sibs
2. All independent pairs of affected/phenotyped sibs\*
3. All pairs of affected/phenotyped sibs\*
4. All pairs pf affected/phenotyped sibs-UNWEIGHTED

Enter the index of the analysis you want to use [1]: 2

\*"independent" pairs of sibs are created by taking the first sib paired with sibs 2...n (for a three-sib sibship this will mean the sharing for pairs 1-2 & 1-3 will be computed). Therefore, the results can be different if you rearrange the order of the sibship. "all" pairs are created by taking the first sib paired with sibs 2...n, the second sib paired with 3...n, etc. For a four-sib sibship this means the sharing for pairs 1-2, 1-3, 1-4, 2-3, 2-4 and 3-4 will be computed. The sibs are considered as part of a whole family when inheritance vectors are determined and then each pair is treated as a essentially a separate pedigree for the purposes of analysis.

You DO NOT need to re-scan for a change in the pair setting to take effect.

The default is to use the first pair of affected/phenotyped sibs.

### DUMP IBD Command

Summary: dump the ibd distribution to a text file  
No Arguments

This command allows you to output the calculated likelihood of sharing 0, 1 or 2 alleles for each relative pair within each pedigree, possibly for use in another program. (You will be queried for the filename to store it in.)

The output format is:

```
<pos> <pedigree> <indiv1-indiv2> <priorz0> <priorz1> <priorz2> <z0> <z1> <z2>
```

This command has been expanded from the original MAPMAKER/SIBS command to include all non-founder relative pairs (regardless of relationship or affected status).

When using sex-specific recombination fractions, these will be appropriately taken into account for calculation of the ibd distribution - as with all sib-pair and QTL analysis capabilities. Here, the output only includes the sex-averaged genetic positions, i.e., the mean of the male and female coordinates. The corresponding male and female genetic positions can be obtained from the output of the 'scan pedigrees' command.

## (6) VARIANCE COMPONENTS

### VARIANCE COMPONENTS Command

Summary: run variance components analysis  
No Arguments

This command looks for evidence of quantitative trait loci (QTLs). At each scan position, the program determines whether a significant amount of the variance in a quantitative trait can be attributed to a QTL at that position. Specifically, it calculates maximum likelihood values for the mean trait value (separately for each sex, if desired), additive and dominance variance components for the QTL, additive and dominance variance components for other, unlinked loci, and an environmental variance component. One or both of the dominance components can be optionally excluded. In addition, the program can incorporate covariate effects by estimating the regression of the trait value on a given covariate value. The significance of the QTL effects is tested by comparing the maximum likelihood model with another one in which the QTL variance components are constrained to equal zero. The likelihood ratio of the two models is used to calculate a LOD score which can be compared to a chi-squared distribution as in classical methods of QTL analysis.

#### **\*Data preparation\***

Phenotype values should be included in the pedigree file, after the genotype values for each individual. Covariates should be listed immediately after the phenotypes. Multiple values may be entered, up to the numbers given by the constants MAX\_PHENOTYPES and MAX\_COVARIATES in npl.h. Each phenotype should be indicated in the map file by a single line reading "0 2" followed by five empty lines (These are needed to maintain consistency with the LINKAGE file format. The data expected by LINKAGE in these lines are not used by Genehunter and can be excluded). Each covariate should be indicated with a single line reading "4 0". In addition, the total number of loci (the first number on the first line of the map file) should include the number of phenotypes and covariates, as well as the number of markers and qualitative traits.

#### **\*Running the program\***

When the "variance components" command is entered, the user is prompted for names for the output files and is then asked whether to include dominance variance components for the unlinked loci and for the QTL. The user is then given the option of entering starting estimates for the model's parameters, rather than letting the program come up with it's own estimates. This option is provided because the program's ability to converge on the maximum likelihood values is sometimes sensitive to the starting guesses. Trying out different starting values and seeing whether the same result is obtained provides a check on the correctness of the results. This should probably be done with all analyses, but is especially needed if the program is yielding odd results, such as negative or unrealistically high LOD scores. If manual input is chosen, the program first displays the total variance of the trait value being examined, as well as the mean trait value (separately for males and females if this option has been chosen). These figures can be helpful in choosing starting values.

#### **\*Output\***

The output file shows a LOD score for each scan position, along with estimates of the means, variance components, and covariate regression coefficients for that position. The corresponding estimates for the null model are also reported. Because the program can sometimes fail to converge on an estimate for some positions, the output indicates for each position whether convergence occurred. When it does not occur, the output shows the estimates for the last position which did converge. If

the program frequently fails to converge, it may be necessary to raise the maximum number of iterations allowed in the estimating algorithm (MAXITS in the file varcom.c). If postscript output is switched on, the program also produces two graphics files. One contains a plot of LOD score versus position, and the other a plot of the proportion of total phenotypic variance accounted for by each component of the maximum likelihood model, versus position.

### **SET STARTING VALUES Command**

Summary: choose method for initial estimates of parameters  
No Arguments

This command determines how the variance components command makes its initial parameter estimates. To use, enter "set starting values" and choose the desired option:

```
npl:1> set starting values
```

Genehunter currently uses a constant fraction of total phenotypic variance.

Possible starting values:

1. ML estimate from adjacent position
  2. Constant fraction of total phenotypic variance
- Enter the index of the start values you want to use [2]:

The first method simply divides the total trait variance evenly among the variance components of the model. Thus it uses the same initial values for each position. The second method does this for the first position, but thereafter uses the maximum likelihood values for the last position. The second method is generally faster, because adjacent positions usually have similar maximum likelihood estimates, hence the algorithm requires fewer iterations to converge when it starts near its destination. However, the method can sometimes prevent convergence on the true maximum likelihood estimate, instead settling on a local maximum near the maximum likelihood values of the adjacent position. The first method, slower but more reliable, is the default.

### **MEANS BY SEX Command**

Summary: choose whether to estimate means by sex  
No Arguments

This command determines whether the variance components command estimates means separately by sex. To use, simply enter "means by sex" and indicate the desired setting:

```
npl:1> means by sex
```

Genehunter currently estimates male and female means separately.

1. Estimate a single mean
  2. Estimate male and female means separately
- Enter the index of the option you want to use [2]:

The default setting of [2] should improve the method's power to detect linkage when sex actually has an effect on the trait's value. Setting [1] can slightly speed things up when no such effects are thought to exist, or when insufficient data exist for one sex.

## (7) TDT COMMANDS

GENEHUNTER(-MODSCORE) now contains a standard implementation of the transmission disequilibrium test (TDT) along with several extensions for using missing data, multiple loci, and estimating significance via simulation.

### TDT Command

Summary: standard single locus TDT  
Argument: <file name>

The 'tdt' command performs the traditional Transmission Disequilibrium Test (Spielman, McGinnis, and Ewens, Am J Hum Genet. 1993 Mar;52(3):506-16.) on the Linkage-style pedigree file specified as the argument. Transmissions from homozygous parents are not counted (the obligately provide a transmitted and untransmitted copy of the same allele) and cases where one parent is missing are used only when the genotyped parent and the proband are both distinct heterozygotes (Curtis and Sham, Am J Hum Genet. 1995 Mar;56(3):811-2.) In addition, the transmissions and non-transmissions are stored for use by multi-locus TDT commands (tdt2, tdt3, tdt4).

The case where the both parents and the proband have the same heterozygous genotype are counted (as a transmission and non-transmission of each allele) but are not stored for use in the multi-locus test. As noted by Dudbridge, et al. (Am J Hum Genet. 2000 Jun; 66(6):2009-2012), the elimination of such cases may lead to a slight upward bias in type I error in multiple locus TDTs and results in transmitted/untransmitted ratios that clearly overestimate the strength of the underlying gene effect. An option exists ("dhskip on") which eliminates all cases in which the two parents are identically heterozygous (whether or not the offspring can reconstruct phase) and use of this option results in a conservative test but a robust estimate of gene effect as estimated by transmission ratio. The correction recommended by Dudbridge, et al., can be trivially reconstructed from the results of multiple-locus TDT with this option on or off - their recommendation is to count each case in which unreconstructed heterozygotes are recovered by a homozygous offspring as one transmission rather than two. With dhskip off these cases are counted twice and with it on they are not counted, therefore adding half the difference to the result with dhskip on produces the recommended test.

### TDT2 Command

Summary: two locus TDT  
Argument: <offset between markers to examine>  
Default: analyze adjacent markers (offset=1)

The 'tdt2' command computes the two-locus version of the TDT. The identical rules for counting transmissions and non-transmissions are employed and as in the standard single marker TDT. If an offset is provided as an argument, the analysis will be done on pairs of markers as follows (1 and 1+offset, 2 and 2+offset, 3 and 3+offset, etc.). By default, offset is set to 1 so with no argument specified, 'tdt2' will produce a two-locus TDT test for marker pairs in map order (1 and 2, 2 and 3, 3 and 4, etc.) By nature, this model assumes there is no recombination between adjacent markers (or at least not a significant amount) which would interfere with the detection of potential founder haplotypes. Therefore it is probably most useful on closely spaced markers and/or in more recently founded populations. See note regarding phase reconstruction in 'help tdt'. This command is only available after the 'tdt' analysis of a pedigree file.

### TDT3 Command

Summary: three locus TDT

Computes a three-locus TDT (see 'tdt2' for a more details about multi-locus TDTs). This command is only available after the 'tdt' analysis of a pedigree file.

### TDT4 Command

Summary: four locus TDT

Computes a four-locus TDT (see 'tdt2' for a more details about multi-locus TDTs). This command is only available after the 'tdt' analysis of a pedigree file.

### PERM1 Command

Summary: permutation test for determining TDT significance  
Argument: <number of simulations>

Since the standard application of the TDT usually involves the analysis of numerous alleles at numerous markers, a significant correction is required to interpret the significance of any one result. Treating the num\_markers x num\_alleles tests as independent is extremely conservative since a) the tests of the alleles at each marker are not independent and b) there will be very rare alleles which will penalize an additional degree of freedom without any chance of providing results of interest. A better test of significance is provided by a permutation method in the 'perm1' command as follows:

- \* create a new data set by taking each pair of transmitted and untransmitted alleles and arbitrarily (at p=0.50) reversing the assignment of which was transmitted
- \* tally and store the results of the TDT for this new data set
- \* repeat 1000 or more (the number of simulations is indicated in the argument to 'perm1'), comparing each simulated data set to the actual results observed in the real data set

After the simulations are completed, a report indicating

- \* how many of the permuted data sets had a higher maximum value and
- \* how many of the permuted data sets had more results above certain thresholds (.01, .001)

is displayed providing a better estimate of the significance of the observed data. This command is only available after the 'tdt' analysis of a pedigree file.

#### PERMUTATION SUMMARY:

12 of 1000 simulations had a larger maximum value than the real best (15.42)  
48 of 1000 simulations had as many tests (22) exceeding p=.01  
19 of 1000 simulations had as many tests (3) exceeding p=.001

### PERM2 Command

Summary: permutation test for determining TDT significance  
Argument: <number of simulations>

This test performs the same permutation test as in 'perm1' but instead examines all permutations of all two locus haplotypes formed by adjacent markers and markers separated by 1 in the current map order. The results can be interpreted as in the 'perm1' command. This command is only available after the 'tdt' analysis of a pedigree file.

### DHSKIP Command

Summary: treatment of identically heterozygous parents  
Argument: <'on' or 'off'>  
Default: displays the current setting

The elimination of cases in which haplotypes cannot be reconstructed results in multiple-locus TDTs with subtle flaws. First, the transmission ratio (which is used to estimate gene effect) is an overestimate since cases in which one copy of a haplotype was transmitted and one was untransmitted are not counted. Secondly, as pointed out by Dudbridge, et al. (Am J Hum Genet. 2000 Jun; 66(6):2009-2012), the elimination of such cases may lead to a slight upward bias in type I error in multiple locus TDTs. Turning dhskip on before the tdt command computes TDTs excluding cases in which both parents are identically heterozygous (providing a conservative test and robust estimation of gene effect). See "help tdt" for more information about use of this command.

## (8) ADDITIONAL COMMANDS

There are several basic features which GENEHUNTER-MODSCORE provides to make the program more friendly and useful. These include on-line help ('help'), the ability to record session output ('photo'), and the ability to accept input from a batch file ('run').

### HELP Command (abbreviation '?')

Summary: GENEHUNTER-MODSCORE on-line help facility  
Argument: <command or topic>

'Help' displays on-line help information for GENEHUNTER-MODSCORE commands and features. Typing 'help' alone produces a list of available topics and commands. For a general description of a numbered topic, type 'help <number>', where <number> is the displayed number of the topic. For help on a more specific command or feature, type 'help <name>', for example:

```
npl:1> help haplotype
```

The on-line help is an exact duplicate of the Postscript reference manual (ghm.ps) which accompanies the distribution.

### PHOTO Command

Summary: record the output of a session in a file  
Argument: <file name>

The "photo" command is used to save a copy of the current GENEHUNTER-MODSCORE session (input and output) in a text file. If you type "photo <file name>", for example,

```
npl:1> photo sample.out
```

all input and output from that point on will be copied into the specified file (here, the file named "sample.out"). Typing "photo off" or quitting GENEHUNTER-MODSCORE terminates this process and closes the photo file. The default extension for a transcript file is ".out". The 'photo' command will append program output to the specified file, so output from several sessions may be collected in the same file if desired.

### RUN Command

Summary: lets GENEHUNTER-MODSCORE take input from a file  
Argument: <file name>

The "run" command instructs GENEHUNTER-MODSCORE to take a series of commands from any text file. This file should contain lines of commands and other input just as they would be typed into GENEHUNTER-MODSCORE interactively.

For example, you might want to use a 'run' file to save setup commands for loading your data:

```
load markers test.loci
increment step 5
postscript on
```

```
count recs on
haplotype off
```

and could be run with the command

```
npl:1> run setup.in
```

where 'setup.in' is the name of the file containing the 5 lines of commands above. This feature is especially useful for providing input to GENEHUNTER-MODSCORE during long runs on data files with many pedigrees which you may wish to let run overnight or at least without any user input.

### **SYSTEM Command**

Summary: execute a command under the operating system  
Argument: <system command>

The 'system' command is used to temporarily interrupt GENEHUNTER-MODSCORE and start up a new command interpreter from the operating system. Commands which are normally typed to the operating system may then be issued. You can return to GENEHUNTER-MODSCORE by typing 'exit' or control-D in most operating systems. If an argument is supplied to 'system', the argument is interpreted just as a normal command issued to the operating system. For example:

```
npl:4> system lp results.out
```

would execute the printing command on your operating system and then return control immediately to GENEHUNTER-MODSCORE.

### **CHANGE DIRECTORY Command (abbreviation 'cd')**

Summary: change the current directory  
Argument: <new directory>

The 'cd' command works essentially the same way it does under Unix. By default, all files are read or written from the current directory unless specified otherwise.

### **TIME Command**

Summary: display the current time  
No Arguments

Display the current time from the system clock.

### **QUIT Command (abbreviation 'q')**

Summary: exit session  
No Arguments

Assures that the program exits properly.

**GENEHUNTER-MODSCORE 3.1.1 COMMAND REFERENCE:**

(1)	DATA PREPARATION COMMANDS .....	1
	LOAD MARKERS Command .....	1
	READ MAP Command .....	2
	USE Command .....	4
(2)	GENEHUNTER-MODSCORE MAPPING COMMANDS .....	5
	SCAN PEDIGREES Command .....	5
	TOTAL STAT Command .....	6
	MODCALC Command .....	7
	ALGEBRAIC CALCULATION Command .....	8
	MAXIMIZATION Command .....	9
	MODSCORE Command .....	11
	CALCULATE P VALUE Command .....	14
	MODEL Command .....	15
	IMPRINTING Command .....	16
	MOBIT SIMULATION Command .....	16
	INCLUDE UNTYPED Command .....	17
	POSITIONS Command .....	17
	LIABILITY CLASS Command .....	17
	JOIN LIABILITY CLASSES Command .....	18
	PENETRANCE RESTRICTION Command .....	18
	ALLFREQ RESTRICTION Command .....	19
	HIGHEST ALLFREQ Command .....	19
	DIMENSIONS Command .....	19
	SAVED MODELS Command .....	19
	LONG MOD OUTPUT Command .....	21
	NORMALIZE ALLFREQ Command .....	21
	NUMBER OF REPLICATES Command .....	22
	SEQUENTIAL SIMULATION Command .....	23
	STORE REPLICATES Command .....	24
	UNTYPED FOUNDERS Command .....	24
	FULL INFORMATION Command .....	24
	SIMULATE UNTYPED Command .....	25
	BEST POSITION Command .....	25
	SET RANDOM SEED Command .....	26
	SHOW DISTRIBUTION Command .....	26
	SINGLE POINT Command .....	27
	COUNT RECS Command .....	27
	HAPLOTYPE Command .....	28
	DISCARD Command .....	28
	MAX BITS Command .....	29
	SKIP LARGE Command .....	29
	ANALYSIS Command .....	30
	SCORE Command .....	30
	POSTSCRIPT OUTPUT Command .....	31
	LETTERS Command .....	31
	DRAWING SCALE Command .....	31
	TITLE Command .....	32
	OFF END Command .....	32
	INCREMENT Command .....	32
	MAP FUNCTION Command .....	33
	UNITS Command .....	33
	DISPLAY SCORES Command .....	34
	COMPUTE SHARING Command .....	34
	DUMP REQUIREMENTS Command .....	34
	HAPLOTYPE METHOD Command .....	35
(3)	SIBS QUALITATIVE TRAIT MAPPING COMMANDS .....	36
	ESTIMATE Command .....	36
	EXCLUDE Command .....	36
(4)	SIBS QUANTITATIVE TRAIT LOCI (QTL) MAPPING COMMANDS .....	38
	HASEMAN ELSTON Command .....	38
	ML VARIANCE Command .....	38

NO DOM VAR Command .....	39
NONPARAMETRIC Command .....	39
(5) OTHER SIBS COMMANDS .....	40
PAIRS USED Command .....	40
DUMP IBD Command .....	40
(6) VARIANCE COMPONENTS .....	42
VARIANCE COMPONENTS Command .....	42
SET STARTING VALUES Command .....	43
MEANS BY SEX Command .....	43
(7) TDT COMMANDS .....	44
TDT Command .....	44
TDT2 Command .....	44
TDT3 Command .....	45
TDT4 Command .....	45
PERM1 Command .....	45
PERM2 Command .....	46
DHSKIP Command .....	46
(8) ADDITIONAL COMMANDS .....	47
HELP Command .....	47
PHOTO Command .....	47
RUN Command .....	47
SYSTEM Command .....	48
CHANGE DIRECTORY Command .....	48
TIME Command .....	48
QUIT Command .....	48

## GENEHUNTER-MODSCORE 3.1.1 QUICK REFERENCE:

### (1) DATA PREPARATION COMMANDS

load markers.....Load marker-locus data  
read map.....Load map file with genetic positions  
use.....Select the current map for analysis

### (2) GENEHUNTER-MODSCORE MAPPING COMMANDS

scan pedigrees.....Analyze pedigree data  
total stat.....Show total scores from a scan of multiple pedigrees  
modcalc.....activate/deactivate MOD-score analysis  
algebraic calculation....activate/deactivate algebraic calculation mode  
maximization.....customizes MOD score routine  
modscore.....perform a MOD-score analysis  
calculate p value.....calculates p-values for MOD or LOD scores  
model.....add a user-defined trait model to MOD-score analysis  
imprinting.....activate/deactivate imprinting analysis  
mobit simulation.....obtain empiric permutation p value for the MOBIT  
include untyped.....include/exclude untyped persons with no kids  
positions.....define genetic positions for the MOD-score analysis  
liability class.....define liability class to be optimized (MOD score)  
join liability classes....join liability classes for a MOD-score analysis  
penetrance restriction...activate/deactivate penetrance restriction (MOD)  
allfreq restriction.....activate/deactivate dis.allfreq. restriction (MOD)  
highest allfreq.....select upper disease allfreq. bound for MOD analysis  
dimensions.....define number of parameters jointly varied (MOD)  
saved models.....define number of saved trait models (modcalc single)  
long mod output.....activate/deactivate long output (modcalc single)  
normalize allfreq.....activate/deactivate marker allfreq normalization  
number of replicates.....defines number of replicates used for cpv  
sequential simulation....activate/deactivate sequential simulation mode  
store replicates.....activate/deactivate storage of replicate information  
untyped founders.....activate/deactivate usage of untyped founders (cpv)  
full information.....activate/deactivate fully informative markers (cpv)  
simulate untyped.....activate/deactivate simulation of missing genotypes  
best position.....perform p-value calc. for the best position only  
set random seed.....sets the random seed for the Mersenne Twister  
show distribution.....activate/deactivate displaying distribution info  
single point.....activate/deactivate single-point analysis  
count recs.....turn recombination counting on  
haplotype.....determine likely haplotypes for individuals  
discard.....eliminate less informative individuals  
max bits.....determine how large a pedigree may be analyzed  
skip large.....determine how large pedigrees are dealt with  
analysis.....select what type of linkage analysis to perform  
score.....select NPL scoring function  
postscript output.....activate Postscript graphing capability  
letters.....controls allele display in Postscript output  
drawing scale.....set scale of Postscript 'total' drawings  
title.....set title of PostScript plots  
off end.....Select how far to compute scores beyond ends of map  
increment.....Choose the scan step size  
map function.....Choose a cM <-> rec-frac conversion function  
units.....Choose whether scan output is in cM or rec-frac  
display scores.....activate screen display of scores and haplotypes  
compute sharing.....turn IBD matrix storage on/off  
dump requirements.....estimate memory usage instead of scanning pedigree  
haplotype method.....choose algorithm used for haplotyping

### (3) SIBS QUALITATIVE TRAIT MAPPING COMMANDS

estimate.....maximum likelihood estimate of IBD sharing  
exclude.....exclusion mapping

### (4) SIBS QUANTITATIVE TRAIT LOCI (QTL) MAPPING COMMANDS

haseman elston.....traditional & EM Haseman-Elston analysis  
ml variance.....maximum likelihood QTL variance estimation

no dom var.....maxlike QTL variance est. under no-dominance assmp.  
nonparametric.....non-parametric QTL analysis

(5) OTHER SIBS COMMANDS

pairs used.....select what pair combinations will be used  
dump ibd.....dump the ibd distribution to a text file

(6) VARIANCE COMPONENTS

variance components.....run variance components analysis  
set starting values.....choose method for initial estimates of parameters  
means by sex.....choose whether to estimate means by sex

(7) TDT COMMANDS

tdt.....standard single locus TDT  
tdt2.....two locus TDT  
tdt3.....three locus TDT  
tdt4.....four locus TDT  
perm1.....permutation test for determining TDT significance  
perm2.....permutation test for determining TDT significance  
dhskip.....treatment of identically heterozygous parents

(8) ADDITIONAL COMMANDS

help.....GENEHUNTER-MODSCORE on-line help facility  
photo.....record the output of a session in a file  
run.....lets GENEHUNTER-MODSCORE take input from a file  
system.....execute a command under the operating system  
change directory.....change the current directory  
time.....display the current time  
quit.....exit session

\* = reference information only - not a command

