Merlin—rapid analysis of dense genetic maps using sparse gene flow trees

Gonçalo R. Abecasis[1,2], Stacey S. Cherny[1], William O. Cookson[1] & Lon R. Cardon[1]

Published online: 3 December 2001, DOI: 10.1038/ng786

© 2002 Nature Publishing Group http://genetics.nature.com

Efforts to find disease genes using high-density single-nucleotide polymorphism (SNP) maps will produce data sets that exceed the limitations of current computational tools. Here we describe a new, efficient method for the analysis of dense genetic maps in pedigree data that provides extremely fast solutions to common problems such as allele-sharing analyses and haplotyping. We show that sparse binary trees represent patterns of gene flow in general pedigrees in a parsimonious manner, and derive a family of related algorithms for pedigree traversal. With these trees, exact likelihood calculations can be carried out efficiently for single markers or for multiple linked markers. Using an approximate multipoint calculation that ignores the unlikely possibility of a large number of recombinants further improves speed and provides accurate solutions in dense maps with thousands of markers. Our multipoint engine for rapid likelihood inference (Merlin) is a computer program that uses sparse inheritance trees for pedigree analysis; it performs rapid haplotyping, genotype error detection and affected pair linkage analyses and can handle more markers than other pedigree analysis packages.

Linkage and association studies routinely involve analyzing many markers in related individuals to determine phased haplotypes, test for cosegregation of disease and marker loci or identify problems in genotyping. The shift to dense SNP maps[1,2] poses new problems to pedigree analysis packages[3–9]. Packages based on the Elston-Stewart algorithm[10] can only handle a small number of markers and are not well suited to SNP maps. On the other hand, memory requirements for the Lander-Green algorithm[11] make analyzing hundreds or thousands of markers a severe challenge in all but the smallest pedigrees. Although Markov-Chain Monte-Carlo (MCMC) sampling methods[7,8,12] complement some of the deficiencies in these two approaches, as the number of tightly linked markers increases, it is difficult to guarantee their adequate convergence. Another unresolved issue is undetected genotyping error, which seriously hinders linkage and association studies[13,14]. As most SNP genotyping errors do not lead to mendelian inconsistencies[15], SNPs require specialized quality-control strategies.

The Lander-Green algorithm[11] considers each alternative gene flow pattern in a pedigree separately. Allele-sharing statistics for each set of observed phenotypes and likelihoods conditional on observed marker data are calculated and stored in memory[4]. Because the pattern of gene flow through a pedigree is fully specified by noting whether the grand-maternal or grand-paternal allele is transmitted in each meiosis, the results of these calculations are typically stored in a bit-indexed array (Fig. 1*a*), where each index bit indicates the outcome of one meiosis[4,9,16]. Binary trees provide another natural organization for results that depend on gene flow patterns. Each level in the tree represents one meiosis, and each branch corresponds to transmission of the grand-maternal or grand-paternal allele (Fig. 1*b*). Often, many alternative patterns of gene flow have the same outcome, and we reasoned that sparse binary trees might provide an efficient framework for pedigree analysis and extend the scope of the Lander-Green algorithm to very large data sets. These sparse trees are a reduced representation of the full binary tree, where gene flow patterns with identical outcomes are combined into symmetric and premature leaf nodes (Fig. 1*c*).

We first evaluated the performance of gene flow trees in single marker analyses using simulated replicates of pedigree D (Fig. 2), which includes 40 meioses. Usually the maternal or paternal origin of founder alleles cannot be discerned, and only 2^{32} representative outcomes must be considered[4]. If outcomes were enumerated in an array, this analysis would exceed the storage capacity of most modern workstations. In comparison, trees describing gene flow pattern likelihoods for SNP markers with equifrequent alleles and 20% missing data have a median size of less than 900 nodes, and are even smaller for more informative markers or smaller amounts of missing data (Table 1). This saves significant amounts of both storage and computing time, and similar savings result when allele-sharing statistics are calculated for most pedigrees.

Table 1 • Complexity of inheritance tree for pedigree D[a]

Missing genotypes	Info[b]	Total nodes			Leaf nodes[c]
		Mean	Median	95% C.I.	
four-allele marker with equifrequent alleles					
–	0.72	154.7	72	64–603	5.2
5%	0.68	245.2	122	64–1,166	9.9
10%	0.64	446.3	171	65–2,429	24.1
20%	0.55	1,747.4	405	69–15,943	107.3
50%	0.28	19,880.6	2,882	154–140,215	2,574.5
two-allele marker with equifrequent alleles					
–	0.42	706.0	151	57–5,447	66.9
5%	0.39	1,299.8	225	57–8,443	159.6
10%	0.36	2,157.7	329	61–15,361	148.9
20%	0.31	8,595.9	872	64–42,592	1,293.9
50%	0.14	55,639.1	4,477	135–383,407	9,173.5

[a]Pedigree D is represented in Fig. 2. [b]Average marker informativeness[4]. [c]The average number of leaf nodes, which correspond to full likelihood evaluations. These statistics summarize 1,000 replicates.

[1]*The Wellcome Trust Center for Human Genetics, University of Oxford, Oxford OX3 7BN, UK.* [2]*Present address: Center for Statistical Genetics, Department for Biostatistics, School of Public Health, 1420 Washington Heights, Ann Arbor, Michigan 48109-2029, USA. Correspondence should be addressed to G.R.A. (e-mail: goncalo@umich.edu).*

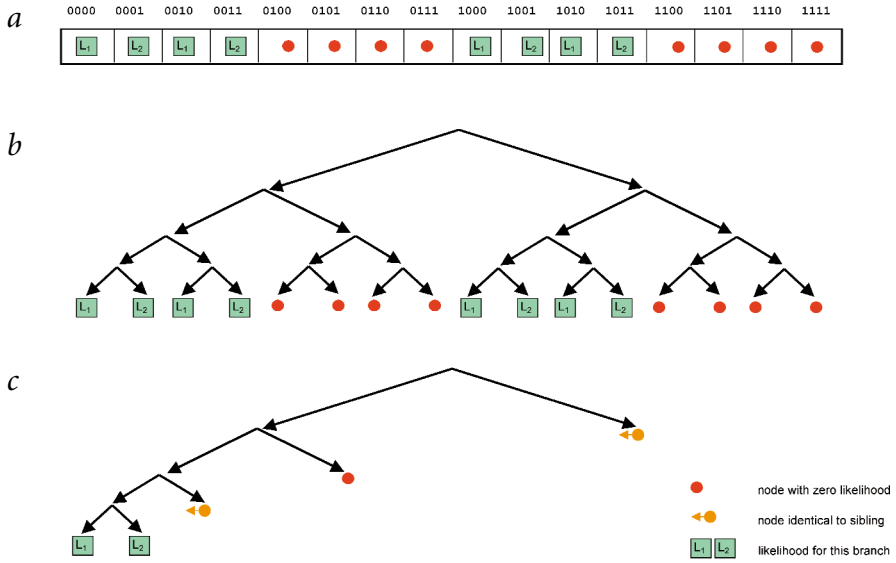

Fig. 1 Alternative representations of gene flow in a pedigree. ***a***, A bit-indexed array. The most common representation, this uses a sequence of binary digits, or an inheritance vector, to specify the outcome of each meiosis. Each of these sequences serves as an index into an array where the statistic of interest is stored. ***b***, A packed tree in which individual meioses are represented as new levels, and likelihoods or other statistics are evaluated for each leaf node. ***c***, A sparse tree, in which each branch (meiosis) is evaluated conditional on the outcome of preceding meioses. Evaluation stops early in sections producing invariant outcomes, resulting in premature leaf nodes (red circles). These occur, for example, when an impossible gene flow pattern is detected. Uninformative meioses produce symmetric nodes and further increase sparseness (orange circle with arrow). These occur, for example, when both parental alleles are indistinguishable.

We next examined whether multipoint calculations using the Lander-Green algorithm[11] could be carried out efficiently when sparse binary trees replace the traditional array organization of likelihoods. The Lander-Green algorithm uses a Markov chain to calculate likelihoods for all gene flow patterns at arbitrary chromosomal locations. The algorithm's rate-limiting step is the transition between informative markers, where probabilities conditional on all markers telomeric to a particular location are derived from conditional likelihoods at the previous marker. This step involves an expensive matrix multiplication operation that can benefit from the redundancies summarized in sparse gene flow trees. For very sparse trees that index a small number of nonzero likelihoods, we used this information in a sparse matrix–vector multiplication algorithm[17]. For trees that index a larger number of nonzero likelihoods, we used the Idury-Elston divide-and-conquer algorithm[18] that proceeds through successive bisections of inheritance space. In this case, the amount of computation is reduced whenever bisection of inheritance space results in identical sub-trees.

Table 2 compares the performance of sparse binary tree–based pedigree analysis using Merlin with two state-of-the-art programs implementing the Lander-Green algorithm, Allegro[9] and Genehunter[16]. The benchmark involved calculation of a nonparametric linkage statistic and pair-wise IBD probabilities for all relative pairs as well as the most likely haplotype path through a pedigree (Merlin, Allegro) or a fast approximation[16] to this solution (Genehunter). All three programs completed analysis of pedigree B, but Merlin performed fastest, whether analyses were required between markers (1 m 47 s with grand-parental genotypes, 1 m 17 s without) or at marker locations only (18 s with grand-parental genotypes, 25 s without). The three programs cope with the complexity of pedigree C differently: Genehunter treats meioses with known outcomes separately to reduce the number of possibilities stored in memory[16] and was successful only when grand-parental genotypes were available; Allegro swaps intermediate results to disk[9] and requires 20 GB of temporary storage to complete analyses; Merlin uses sparse binary trees to summarize redundancies in inheritance space and completed analysis in memory in less than two hours.

Table 2 • Comparative timings for Genehunter, Allegro and Merlin (with 5% missing data)

	All individuals in pedigree genotyped							
	A (×1000)		B		C		D	
Genehunter (v 2.1)								
exact	54 s	(38 s)	7 m 41 s	(37 s)	16 h 43 m 24 s	(18 m 16 s)	*	
Allegro (v 1.1b)								
exact	30 s	(18 s)	2 m 48 s	(2m17 s)	5 h 16 m 30 s	(3 h 54 m 13 s)	*	
Merlin (v 0.1)								
exact	16 s	(11 s)	1m 47 s	(18 s)	1 h 55 m 14 s	(13 m 55 s)	*	
approximation	22 s	(15 s)	4 s	(2 s)	36 s	(20 s)	1 m 28 s	(59 s)
	Top generation not genotyped							
	A (×1000)		B		C		D	
Genehunter (v 2.1)								
exact	1 m 03 s	(45 s)	8 m 25 s	(1 m 54 s)	*		*	
Allegro (v 1.1b)								
exact	30 s	(18 s)	1 m 29 s	(1 m 08 s)	1 h 29 m 11 s	(1 h 12 m 38 s)	*	
Merlin (v 0.1)								
exact	18 s	(13 s)	1 m 17 s	(25 s)	47 m 35 s	(15 m 50 s)	*	
approximation	27 s	(18 s)	11 s	(6 s)	2 m 38 s	(1 m 30 s)	3 m 45 s	(2 m 09 s)

Nonparametric linkage analysis, haplotyping and pair-wise IBD estimation were carried out at every marker location and halfway between consecutive markers (parentheses indicate analysis at marker locations only) on a 700 Mhz Pentium III with 2 GB of RAM (and 20 GB of disk for Allegro). Fifty microsatellite markers, with four equifrequent alleles, were simulated at 2-cM spacing. 'Approximation' indicates calculations assuming two or fewer recombinants between consecutive markers; all other rows refer to exact calculations. *Analysis could not be completed due to memory requirements.

© 2002 Nature Publishing Group http://genetics.nature.com

© 2002 Nature Publishing Group http://genetics.nature.com

The number of catalogued SNP markers presents not only daunting computational challenges but also opportunities to develop solutions that account for reduced recombination between markers. Merlin can construct approximate solutions for dense maps where the probability of observing several recombinants between consecutive markers is close to zero by restricting analysis to gene flow patterns separated by a few recombination events or less. For the 'Approximation' row in Table 2, calculations assumed no more than two recombinants between consecutive markers. With this assumption, we could estimate haplotypes, lod scores and pair-wise IBD probabilities for pedigree D, and calculation speed improved for pedigrees B and C. In 1,000 realizations of pedigree B, Spearman's rank correlation coefficient between nonparametric Z-scores derived by this approximation and exact Z-scores was r_s=0.999, whereas the mean absolute difference between the two was less than 0.01. If no recombination between markers is assumed, Merlin can list all possible non-recombinant haplotypes in a pedigree with minimal computing requirements. These nonrecombinant haplotypes could be used to estimate founder haplotype frequencies[19] in measured haplotype analyses of candidate genes[20] or when microsatellite markers are replaced with clusters of SNPs.

Merlin also has the ability to conduct a sensitivity analysis of the multipoint likelihood. This analysis identifies genotypes that imply a recombination pattern that is not supported by neighboring markers and which are likely to be erroneous (Methods). Table 3 summarizes the performance of our error-detection strategy in a 1-cM SNP map, assuming all errors are hard-to-detect single-allele changes. Note that the majority of such errors do not produce mendelian inconsistencies, even when both parents are typed. The proportion of errors detected by Merlin increases with the number of genotyped individuals in each family, from approximately 16% with only two genotyped siblings to greater than 60% in families with four genotyped siblings, or more than 90% of errors when both parents and at least two offspring are genotyped. Note that even when only a minority of errors is detected, a significant amount of power can be recovered[13]. In general, performance improves with the number of genotyped individuals and increased map density. In sib pairs with no parents, performance is similar to previous methods[13].

Table 3 • Summary of unlikely genotype analyses in nuclear families

	Mendelian errors	Unlikely genotypes	Overall detection rate
No genotyped parents			
2 siblings	0.00	0.16	0.16
3 siblings	.00	.38	.38
4 siblings	.00	.61	.61
5 siblings	.00	.77	.77
One genotyped parent			
2 siblings	0.13	0.34	0.47
3 siblings	.13	.58	.71
4 siblings	.12	.72	.84
5 siblings	.12	.78	.91
Two genotyped parents			
2 siblings	0.37	0.56	0.93
3 siblings	.37	.56	.93
4 siblings	.38	.59	.97
5 siblings	.37	.60	.97

We simulated a map of 21 SNP markers with equal allele frequencies separated by 1 cM and introduced a single-allele change in the first offspring's middle-marker genotype. The table lists the proportion (in 10,000 replicates) of genotypes resulting in mendelian incompatibilities or flagged as unlikely, as well as an overall error detection rate. The proportion of mendelian errors is 0.0, 0.125 and 0.375 depending on whether zero, one or two parental genotypes are available, respectively. Additional variation in results is due to sampling error.

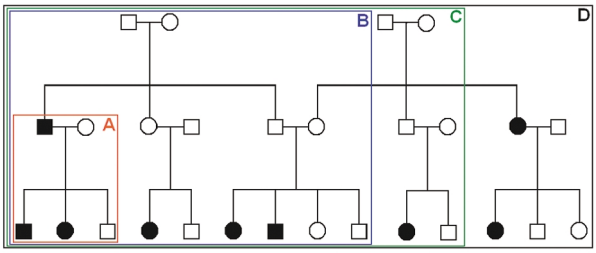

Fig. 2 Sample pedigrees used in simulations. Pedigree A is typical of affected sib-pair studies. Pedigree B, C and D are larger pedigrees used for benchmarking. Complexity (2*n*–*f*) is 4, 19, 25 and 32 bits for pedigrees A, B, C and D, respectively. If grandparents are not genotyped and male and female recombination fractions are assumed equal, complexity becomes 4, 18, 23 and 30 bits, respectively[9].

We have described a representation of gene flow patterns based on sparse binary trees that extends the scope of the Lander-Green algorithm to much larger problems. We also describe a companion genotype-error detection strategy. As part of an analysis of chromosome-wide SNP haplotypes (E. Dawson *et al.*, manuscript submitted), we have haplotyped over 1,500 SNP markers in pedigrees of up to 24-bit complexity using less than 2 GB of memory and 500 MB of disk space. These analyses would require over 500 GB of storage using alternative methods. It is likely that additional opportunities exist for constructing approximate solutions based on sparse inheritance trees. For example, trees could be 'pruned' by removing unlikely gene flow patterns or, borrowing practices used in digital-signal processing, trees could be quantized by marking contiguous patterns of gene flow with similar likelihoods as effectively identical. Alternatively, trees could be manipulated within the reduced inheritance space described by Markianos *et al.*[16] to provide even faster exact solutions.

We have implemented our new approach in a C++ program that is freely available. Merlin carries out common single-point and multipoint analyses of pedigree data, including IBD and kinship calculations, nonparametric and variance component linkage analyses, error detection and information content mapping. For multipoint analyses in dense maps, Merlin allows the user to impose constraints on the number of recombinants between consecutive markers. Merlin estimates haplotypes by finding the most likely path of gene flow or by sampling paths of gene flow at all markers jointly. It can also list all possible nonrecombinant haplotypes within short regions. Finally, Merlin provides swap-file support for handling very large numbers of markers as well as gene-dropping simulations for estimating empirical significance levels.

Methods

Definitions. Consider a pedigree with f founders (individuals with no ancestors in the pedigree) and n nonfounders (their descendants). If each individual has either both parents or else none at all in the pedigree there will be $2n$ parent–offspring pairs. Each parent–offspring pair corresponds to a single meiotic event. For each chromosomal location ℓ, let the inheritance vector v_ℓ represent gene flow in a pedigree through a sequence of $2n$ binary digits, such that the i^{th} digit, $v_\ell(i)$, is 0 if the grand-maternal allele is transmitted in the meiosis connecting offspring o_i and parent p_i, and 1 otherwise. Define the index of the first digit in any such vector to be 1 so that i ranges between 1 and $2n$.

Sparse gene flow trees. A common strategy for analyzing pedigree data is to enumerate all possible inheritance vectors (v) and calculate likelihoods and/or linkage statistics for the pedigree conditional on each v. If vectors

© 2002 Nature Publishing Group http://genetics.nature.com

are inspected in a sequential fashion, the complexity of these problems is proportional to the number of possible inheritance vectors, 2^{2n}, and prohibitive for pedigrees of moderate size. Often, the calculations required (and their results) are similar for many different v, and smart algorithms can be constructed. For example, if inheritance vectors are enumerated in Gray code order[17,21], consecutive vectors differ by a single digit (meiosis) and the statistic of interest can be calculated with 2^{2n} partial updates (each update could be carried out using a previously described method[7]).

We propose that conducting calculations in the space of sparse gene flow trees reduces both computation and storage requirements for common genetic applications and provides an efficient approach for analyzing arbitrary pedigrees of moderate size. Sequential binary choices can be represented as binary trees where each bifurcation represents one digit. For inheritance vectors, each level in the tree represents an individual meiosis (Fig. 1*b*). These binary trees can be constructed using recursive functions, where each evaluation updates the statistic of interest using information contributed by fixing the outcome of the current meiosis and then creates two new branches that correspond to alternative outcomes for the following meiosis and are processed by recursion. This would require $2^{2n+1}-1$ partial updates, which is about twice as many as are required when enumerating inheritance vectors in Gray code order. For many applications, however, some meioses are not informative when evaluated conditional on the outcome of predecessor meioses.

In sparse inheritance trees, two additional types of node obviate unnecessary bifurcations in a standard binary tree: (i) symmetric nodes, where the statistic of interest yields identical outcomes on both branches of the tree and (ii) premature leaf nodes, where the statistic of interest yields identical outcomes for all subsequent branches of the tree (Fig. 1*c*). Note that each symmetric node has only one child tree, so that symmetry detected at meiosis i reduces the size of the tree by $2^{2n+1-i}-1$ nodes. A premature leaf node at meiosis i reduces the size of the tree by $2^{2n+1-i}-2$ nodes. In a recursive function, information accumulated up to meiosis i can be used to decide whether to evaluate and store the outcomes of the next meiosis separately (using a bifurcation node), together (in a symmetric node) or not at all (in a premature leaf node).

As binary trees and sparse binary trees are simply compact representations of inheritance vectors, common vector operators such as component-wise multiplication and addition retain their natural definitions.

Single-marker likelihoods. Let G_ℓ be a set of genotypes for a codominant marker at location ℓ. Although G_ℓ will usually not identify a unique pattern of gene flow at ℓ, it defines a likelihood for every inheritance vector v_ℓ. Briefly, this likelihood can be calculated by enumerating a set $A(G_\ell,v_\ell)$ including all founder allele states $a=[a_1, a_2, \ldots, a_{2f}]$ compatible with both v_ℓ and G_ℓ. Then the likelihood is $L(v_\ell|G_\ell) \propto 2^{-2n}\Sigma_a\Pi_i f(a_i)$, where $f(a_i)$ is the frequency of allele a_i. This likelihood assumes no errors in G_ℓ; later, we propose a strategy for identifying such errors.

Although G_ℓ defines a vector of likelihoods for all inheritance vectors at location ℓ ($\lambda_{\ell|G_\ell}$), the storage and CPU time requirements for defining $\lambda_{\ell|G_\ell}$ through exhaustive enumeration of all v_ℓ are prohibitive for pedigrees with more than approximately 30 meioses. Many v_ℓ imply identical founder allele sets A_ℓ and have equal likelihood, and many of these redundancies can be summarized in sparse binary trees using the following procedure:

When constructing a tree for locus ℓ, update A_ℓ and G_ℓ conditional on the outcomes of meioses $1\ldots i-1$ before evaluating meiosis i. Then define a premature leaf node if the likelihood for all nodes in this part of the tree is zero, that is, if $A_\ell=\varnothing$ (for example, if this set of meiotic outcomes requires allele sharing between individuals with different genotypes). Define a symmetric node if parent p_i is known to be homozygous or if offspring o_i and all its descendants are not genotyped, as, in these cases, A_ℓ is independent of the outcome of meiosis i. If symmetry or zero-likelihood nodes cannot be identified, add a branching node and proceed to evaluate meiosis $i+1$.

Full multipoint calculations. Let $G=[G_1, G_2, \ldots, G_k]$ define a set of codominant genotypes for k markers separated by recombination fractions $\theta=[\theta_{1,2}, \theta_{2,3}, \ldots, \theta_{k-1,k}]$. Assuming no interference, the likelihood of inheritance vectors at an arbitrary location ℓ conditional on all observed genotypes, $\lambda_{\ell|G,\theta}$, can be derived from the single marker likelihoods $\lambda_{\ell|G_\ell}$ at each location using a hidden Markov process[11].

Briefly, the likelihood for any inheritance vector v, $\lambda_{\ell|G,\theta}(v)$, can be factorized into a left conditional likelihood, $\lambda_{\ell|G_{1..\ell-1},\theta_{1-\ell}}(v)$, a right conditional likelihood, $\lambda_{\ell|G_{\ell+1..k},\theta_{\ell..k}}(v)$, and a single marker likelihood $\lambda_{\ell|G_\ell}(v)$. That is, $\lambda_{\ell|G,\theta}(v)=\lambda_{\ell|G_{1..\ell-1},\theta_{1..\ell}}(v)\ \lambda_{\ell|G_\ell}(v)\ \lambda_{\ell|G_{\ell+1..k},\theta_{\ell..k}}(v)$. Now define transition matrices $T_{\theta_{a,b}}$ for each pair of consecutive loci a and b, with elements $T_{\theta_{a,b}}(v,w)=\theta_{a,b}^{r(v,w)}(1-\theta_{a,b})^{2n-r(v,w)}$, where $r(v,w)$ is the number of differences between inheritance vectors v and w. Then left conditional likelihoods at locus ℓ, $\lambda_{\ell|G_{1..\ell-1},\theta_{1..\ell}}$ can be defined on the basis of left conditional likelihoods at locus $\ell-1$, $\lambda_{\ell-1|G_{1..\ell-1},\theta_{1..\ell-1}}$, as:

$\lambda_{\ell|G_{1..\ell},\theta_{1..\ell}}(v)=\lambda_{\ell|G_{1..\ell-1},\theta_{1..\ell}}(v)\lambda_{\ell|G_\ell}(v)$ and

$\lambda_{\ell|G_{1..\ell-1},\theta_{1..\ell}}=T_{\theta_{\ell-1,\ell}}\lambda_{\ell-1|G_{1..\ell-1},\theta_{1..\ell-1}}$.

At the first marker, the left conditional likelihoods are simply $\lambda_{1|G_1}$. Right conditional probabilities can be calculated through an analogous Markov process. The rate-limiting steps in this Markov process are the successive multiplications of transition matrices and likelihood vectors and sparse binary trees. If the number of nonzero likelihoods at two neighboring locations ℓ and $\ell-1$ is small, then sparse binary trees can be used as an index of nonzero likelihoods in sparse matrix–vector multiplication algorithms[17]. Alternatively, each convolution can be carried out by successive bisections of inheritance vector space[18]. These bisections correspond to recursively processing offspring trees for each node. Performance benefits, because the transforms of symmetric nodes and premature leaf nodes are also symmetric nodes and premature leaf nodes, respectively.

Multipoint analysis in dense maps. If all recombination fractions are small, it may be reasonable to assume that the probability of observing two or more recombinants between consecutive markers is effectively zero. This corresponds to setting all elements (v,w) of the transition matrix, $T_{\theta_{a,b}}$, to zero if $r(v,w)>1$ and produces a very sparse transition matrix. Then the Markov chain used to calculate left and right conditional probabilities can use sparse matrix-vector multiplication and considers only the most likely inheritance vectors. To extend this approximation to up to r recombination events in each step in the Markov chain and to allow efficient computation, we divide the interval between loci a and b into n equal segments and assume no more than one recombinant in each segment. Then we apply the original approximation r times using the correspondingly smaller recombination fraction. The expected number of recombinants between consecutive markers ℓ and $\ell+1$ in a pedigree is simply the product of the number of meioses, $2n$, and the recombination fraction, $\theta_{\ell,\ell+1}$; in general, these approximations are useful when $2n\theta_{\ell,\ell+1}<1$.

Nonparametric linkage statistics. Statistics that measure sharing among affected individuals can be scored using a strategy similar to that used for single-marker likelihoods. First, update the statistic of interest up to meiosis $i-1$. Then define either a symmetric node if meiosis i connects parent p_i to unaffected offspring o_i and all descendants of offspring o_i are also unaffected, or a premature leaf node if all further meioses lead to unaffected individuals only. Otherwise, add a branching node to the tree and proceed to meiosis $i+1$, evaluating the two possible outcomes of meiosis i.

Error detection. Erroneous genotypes can imply excessive and unlikely recombination events between tightly linked markers. To detect unlikely genotypes, we calculate the likelihood of observed genotypes conditional on all recombination fractions $L(G|\theta)$ and also assuming that all markers are unlinked, $L(G|\theta=\frac{1}{2})$. We then mark, in turn, each genotype g as unknown and updated these likelihoods to obtain $L(G\backslash g|\theta)$ and $L(G\backslash g|\theta=\frac{1}{2})$. If the information provided by g was consistent with neighboring markers, we expect that the ratio $r_{\text{linked}}=L(G\backslash g|\theta)/L(G|\theta)$ would be small compared to $r_{unlinked}=L(G\backslash g|\theta=\frac{1}{2})/L(G|\theta=\frac{1}{2})$. Genotypes that provide information inconsistent with neighboring markers, however, will cause the statistic $r=r_{\text{linked}}/r_{\text{unlinked}}$ to take unusually large values.

In the absence of errors, we expect that the probability of observing statistic r is less than $1/r$ for any pedigree structure and missing data pattern. There are two options for interpreting the statistic: select a large value of r as the threshold for flagging problem genotypes (such as $r>100$) or determine the precise relationship between r and false-positive rates through simulation. Genotypes where $r>40$ were flagged as errors (Table 3). This corresponds to a false-positive rate of <0.001 through simulation. The proportion of detected errors changed by less than 5% in all cases, if a threshold of $r>100$ was used.

Merlin. Merlin is available with source code at http://bioinformatics.well.ox.ac.uk/Merlin and http://www.sph.umich.edu/csg/abecasis/Merlin.

Acknowledgments

This work was supported by the Wellcome Trust through a Prize Studentship (G.R.A.) Senior Research Fellowship (W.O.C.) and a Principal Research Fellowship (L.R.C.), and by the National Eye Institute (S.S.C. and L.R.C.).

Received 21 August; accepted 22 October 2001.

1. Mullikin, J.C. *et al*. An SNP map of human chromosome 22. *Nature* **407**, 516–520 (2000).
2. Altshuler, D. *et al*. An SNP map of the human genome generated by reduced representation shotgun sequencing. *Nature* **407**, 513–516 (2000).
3. Lathrop, G.M., Lalouel, J.M., Julier, C. & Ott, J. Multilocus linkage analysis in humans: detection of linkage and estimation of recombination. *Am. J. Hum. Genet.* **37**, 482–498 (1985).
4. Kruglyak, L. & Lander, E.S. Complete multipoint sib-pair analysis of qualitative and quantitative traits. *Am. J. Hum. Genet.* **57**, 439–454 (1995).
5. O'Connell, J.R. & Weeks, D.E. The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. *Nature Genet.* **11**, 402–408 (1995).
6. Cottingham, R.W. Jr, Idury, R.M. & Schaffer, A.A. Faster sequential genetic linkage computations. *Am. J. Hum. Genet.* **53**, 252–263 (1993).
7. Sobel, E. & Lange, K. Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics. *Am. J. Hum. Genet.* **58**, 1323–1337 (1996).
8. Heath, S.C. Markov chain Monte Carlo segregation and linkage analysis for oligogenic models. *Am. J. Hum. Genet.* **61**, 748–760 (1997).
9. Gudbjartsson, D.F., Jonasson, K., Frigge, M.L. & Kong, A. Allegro, a new computer program for multipoint linkage analysis. *Nature Genet.* **25**, 12–13 (2000).
10. Elston, R.C. & Stewart, J. A general model for the genetic analysis of pedigree data. *Hum. Hered.* **21**, 523–542 (1971).
11. Lander, E.S. & Green, P. Construction of multilocus genetic linkage maps in humans. *Proc. Natl Acad. Sci. USA* **84**, 2363–2367 (1987).
12. Guo, S.W. & Thompson, E.A. A Monte Carlo method for combined segregation and linkage analysis. *Am. J. Hum. Genet.* **51**, 1111–1126 (1992).
13. Douglas, J.A., Boehnke, M. & Lange, K. A multipoint method for detecting genotyping errors and mutations in sibling-pair linkage data. *Am. J. Hum. Genet.* **66**, 1287–1297 (2000).
14. Abecasis, G.R., Cherny, S.S. & Cardon, L.R. The impact of genotyping error on linkage and association analysis of quantitative traits. *Eur. J. Hum. Genet.* **9**, 130–134 (2001).
15. Gordon, D., Heath, S.C. & Ott, J. True pedigree errors more frequent than apparent errors for single nucleotide polymorphisms. *Hum. Hered.* **49**, 65–70 (1999).
16. Markianos, K., Daly, M.J. & Kruglyak, L. Efficient multipoint linkage analysis through reduction of inheritance space. *Am. J. Hum. Genet.* **68**, 963–977 (2001).
17. Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. *Numerical Recipes in C*. (Cambridge University Press, New York, 1992).
18. Idury, R.M. & Elston, R.C. A faster and more general hidden Markov model algorithm for multipoint likelihood calculations. *Hum. Hered.* **47**, 197–202 (1997).
19. Excoffier, L. & Slatkin, M. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol.* **12**, 921–927 (1995).
20. Keavney, B. *et al*. Measured haplotype analysis of the angiotensin-I converting enzyme gene. *Hum. Mol. Genet.* **7**, 1745–1751 (1998).
21. Gray, F. Pulse Code Communication. in *Patent 2,632,058* (USA, 1953).

© 2002 Nature Publishing Group http://genetics.nature.com