Genetic Epidemiology 8:29–46 (1991)

Efficient Computation of Patterned Covariance Matrix Mixed Models in Quantitative Segregation Analysis

Nicholas Schork

Division of Hypertension, Department of Medicine and Department of Statistics, University of Michigan, Ann Arbor, Michigan

The use of patterned covariance matrices in forming pedigree-based mixed models for quantitative traits is discussed. It is suggested that patterned covariance matrix models provide intuitive, theoretically appealing, and flexible genetic modeling devices for pedigree data. It is suggested further that the very great computational burden assumed in the implementation of covariance matrix-dependent mixed models can be overcome through the use of recent architectural breakthroughs in computing machinery. A brief and nontechnical overview of these architectures is offered, as are numerical and timing studies on various aspects of their use in evaluating mixed models. As the kinds of computers discussed in this paper are becoming more prevalent and easier to access and use, it is emphasized that it behooves geneticists to consider their use to combat needless approximation and time constraints necessitated by smaller, scalar computation oriented, machines.

Key words: segregation analysis, supercomputers, multivariate normal distribution, continuous phenotypes

INTRODUCTION

Quantitative segregation analysis, or the statistical inference of genetic mechanism from metrical or continuous data collected from genetic units such as nuclear families or extended pedigrees, has enjoyed a prominent place among many statisticians and biologists for some time. Early twentieth-century scientific luminaries Karl Pearson [1893,1895] and Ronald Fisher [1918] developed mathematical models that molded the earlier work of Mendel and others into an intuitive and testable framework which has become both the inspiration and theoretical foundation for the entire enterprise

Received for publication July 9, 1990; revision accepted October 22, 1990.

Address reprint requests to Nicholas Schork, Division of Hypertension, Department of Medicine, University of Michigan, R6592 Kresge I, Ann Arbor, MI 48109-0500.

of quantitative genetics. Despite a wealth of theoretical research extending the initial work of Pearson and Fisher, there remains at least one problem area for the modern quantitative segregation analyst: the *implementation* of models characterizing the simultaneous activity of monogenic (i.e., single locus, few alleles, etc.), polygenic (the combined force of many loci), and environmental determinants affecting quantitative phenotypes. The importance of such models cannot be overemphasized since quantitative traits are generally determined by a combination of genetic and environmental mechanisms. Current models for the effects of mixed genetic determinants on quantitative traits (hereafter referred to as ''mixed models'') make use of the intuitive and strong theoretical basis put forth by Pearson and Fisher, so the problems surrounding them really only involve their *application* or assessment with respect to data collected on a trait whose genetic determinants are in question. Many of these problems are almost entirely computational in nature.

The seminal work of Elston and Stewart [1971] exposed the computational burden tacit in the implementation of mixed models and offered an ingenious device to reduce this burden, but cautioned the reader that, even with the use of such a device, implementation of mixed models might not be feasible. Morton and MacLean [1974] derived the likelihood function for the first implementable mixed model. Since Morton and MacLean's function involved an analytically intractable integral, they suggested numerical methods for its evaluation. However, as estimation of the segregation parameters assumed in this function must be obtained through the numerical maximization of the likelihood function itself, Morton and MacLean's method basically involves the approximation of approximation—a task not only itself computationally burdensome but of questionable accuracy as well. In an important paper Ott [1979], building on the work of Elston and Stewart [1971] and Lange et al. [1976], derived an expression for the mixed model that involved no integrals. Unfortunately, Ott's formulation did involve summations that grew as 3^n, where n is the number of persons in a pedigree and, therefore, drove him to the conclusion that the implementation of the exact formulation of the mixed model would be virtually impossible for all but the smallest of pedigrees. Finally, Hasstedt [1982] incorporated an analytic approximation to integrals of the type that appeared in Morton and MacLean's equation into a model which has worked well in practice, but whose accuracy has not been adequately assessed.

The above discussion suggests that many efforts to reduce the complexity of mixed-model implementation through analytic, as opposed to programming or computer architecture oriented, means have been tried. The reason for this large amount of work on analytic, as opposed to computational, approaches stems from a pervasive attitude within the modern genetics community that can be summarized by the comments of Edwards [1982], who, in discussing the related analytic technique of linkage analysis, said of the complexity of the necessary likelihood functions, that

> . . . solutions are possible in practice, but attempts at developing algorithms are difficult and explicit procedures based on enumeration [of necessary aspects of relevant functions] too slow for computers, or for any theoretically possible computer. [p. 50].

Despite Edward's claim, it is surprising that no one has actually tried to use modern ''supercomputers'' (i.e., vector and parallel processor computers) to evaluate genetic models outside of sequence research given the very explicit computational demands of genetic modeling. This paper is meant to describe both experiences with, and the potential of, supercomputers in evaluating complex statistical genetic models. Though the focus of this paper is on Ott's [1979] method for evaluating mixed models for quantitative traits through the use of patterned covariance matrices, it should be understood that many of the techniques discussed are entirely applicable to qualitative trait segregation and linkage models. Thus, the paper should be of wide interest to the genetics community. In addition, though the experiments and ideas described herein involve relatively "compact" settings (e.g., nuclear families, univariate traits) similar methodologies and ideas are being applied by the author to models for multivariate traits, large extended predigree data, multiallelic and linked traits, skewed traits, and covariate-dependent traits, all of which, it is hoped, will be described in future work.

MIXED MODELS VIA PATTERNED COVARIANCE MATRICES

Ott's [1979] derivation of the mixed model uses a technique which has become a standard for modern theoretical and applied statisticians: the partitioning of the variability of a quantitative measure into identifiable sources or components. The "sources" in genetics applications include variability induced by activity at a single locus (monogenic activity), the combined effects of many loci, each of small effect (polygenic activity), and the environment. We shall recast Ott's derivation of the mixed model by focusing first on his method of partitioning polygenic and environmental sources of variation, then proceed to his method for modeling monogenic sources of variation, and finally consider the combination of polygenic, environmental, and monogenic sources of variation.

Following the lead of Fisher [1918] and Lange et al. [1976], Ott considered modeling the covariation among individuals in a given pedigree (e.g., mother–daughter covariation; grandfather–grandson covariation; uncle–niece covariation) through the use of a multivariate normal probability density function. Recognizing that any reasonable model would have to be consistent with the laws governing reproduction and genetic transmission (such as Mendel's), he partitioned an $n \times n$, where n is the number of people in the pedigree, covariance matrix Ω (whose rows and columns represent the pedigree members) of a multivariate normal function into theoretically reasonable components. One such partition can be written as

$$\Omega = 2\mathrm{A}\sigma_a^2 + D\sigma_d^2 + S\sigma_s^2 + E\sigma_e^2 \tag{1}$$

where σ_a^2 is the variance in the pedigree due to additive polygenic sources, σ_d^2 is the variance due to the effects of dominance occurring within polygenic sources, σ_s^2 is the variance due to shared household effects and σ_e^2 is the variance due to random environmental effects. The other terms in Eq. (1) are $n \times n$ matrices that specify the theoretical relationships that obtain between any two individuals, i and j, in the pedigree and a specific component. Thus, A is the kinship coefficient matrix as described by Lange et al. [1976], D is Jacquard's [1974] condensed coefficient of identity matrix, Δ_7, S is a matrix such that its (ij)th component is 1 if pedigree members i and j share the same household and 0 if they do not, and E is the identity matrix.

To obtain maximum likelihood estimates (MLEs) of the variance components, σ^2_a, σ^2_d, σ^2_s, and σ^2_e, in Eq. (1) as well as the mean value, μ, of the trait under scrutiny, the likelihood equation [Edwards, 1972] involving pedigree data, x_n, must be maximized, this equation being:

$$L(\mu, \Omega | x_n) = \frac{1}{(2\pi)^{n/2} |\Omega|^{1/2}} e^{-1/2(x-\mu)'\Omega^{-1}(x-\mu)} \qquad (2)$$

where μ is a mean vector of length n whose components are (the same scalar mean) μ, Ω is the covariance matrix given in Eq. (1), and x_n is a vector of length n containing the trait values of the pedigree members. If data have been collected from N pedigrees, each with n_j ($j = 1, \ldots, N$) members, the appropriate likelihood equation would involve the joint density of the N pedigrees, which is given by the product (or sum of the logarithms) of the right-hand side of Eq. (2) evaluated for each pedigree.

Of importance in Eq. (2) is its departure from the paradigmatic form of the multivariate normal density function. The paradigmatic form of an n-variable multivariate normal density function implicates n means μ_i ($i = 1, \ldots, n$); that is, one mean for each variable (this is unlike Eq. (2) where there is a single mean for the trait that is common among the n pedigree members, i.e., variables). In addition, the paradigmatic form need not have a "pattern" to its covariance matrix, as does Eq. (2). These departures need to be emphasized since they will reemerge in the discussion of Ott's method of modeling monogenic sources of variation and may be important factors in the reliable numerical estimation and interpretation of the parameters themselves.

In describing Ott's method of modeling the effects of a single locus on a quantitative trait, we consider the simplest case: an autosomal locus with two alleles, A and a. For clarity, suppose the "a" allele is the "disease" allele or that allele associated with an effect of interest. This case has been given considerable attention in the literature (see, for instance, Morton [1967] and Elston [1981]). The two alleles produce three unique genotypes $g \in \{AA, Aa, aa\}$, each occurring in the population with the Hardy–Weinberg frequency, f_g; i.e., $f_{AA} = (1 - p)^2$, $f_{Aa} = 2p(1 - p)$, $f_{aa} = p^2$, where p is the frequency of the a allele. Associated with each genotype is a mean μ_g and variance σ^2_g implicated in a normal density function labeled the "penetrance" function, ϕ_g. The variance, σ^2_g, characterizes deviations of a trait value x from a mean value μ_g brought about by environmental or, possibly, polygenic effects. The penetrance function, when evaluated at a trait value, x, produces a result that is proportional to the probability that the trait value x would occur given that the person expressing x has genotype g:

$$\phi_g(x) = \frac{1}{\sqrt{2\pi\sigma^2_g}} e^{-1/2(x-\mu_g)^2/\sigma^2_g} \quad . \qquad (3)$$

Typically, it is assumed the effects of the three genotypes share a common variance, σ^2 (i.e., $\sigma^2{}_{AA} = \sigma^2{}_{Aa} = \sigma^2{}_{aa}$). The unconditional probability, Φ, of a trait value, x (that is, the probability derived without reference to a specific genotype), of a pedigree member whose parents are not part of the pedigree (termed a "founder") is thus given through the use of a normal mixture probability density function:

$$\Phi(x) = \sum_{g=1}^{3} f_g \cdot \phi_g(x) \tag{4}$$

where, for clarity, we have $g \in \{1 = AA, 2 = Aa, 3 = aa\}$. For offspring, the frequency parameters, f_g, are replaced with "transmission probabilities." Transmission probabilities reflect the probability that an offspring, k, has genotype g_k, given that his mother, m, and father, f, have genotypes g_m and g_f, respectively. These probabilities can be expressed as $\tau(g_k | g_m, g_f)$ and are discussed by Elston [1981]. Basically, if one assumes Mendelian inheritance for the trait, then the values for the τ's follow immediately: $\tau(AA | Aa, AA) = 1/2$, $\tau(aa | AA, AA) = 0$, $\tau(AA | AA, AA) = 1$, etc. In estimating the parameters p, μ_{AA}, μ_{Aa}, μ_{aa}, and σ^2, one can maximize the likelihood equation implicating the trait values of the pedigree members. For brevity's sake, we derive the likelihood equation only for a five-member nuclear family, which can be written as

$$L(p, \mu_{AA}, \mu_{Aa}, \mu_{aa}, \sigma^2 | x_1, x_2, x_3, x_4, x_5) = \sum_{g_1}\sum_{g_2}\sum_{g_3}\sum_{g_4}\sum_{g_5} f_{g_1}\phi_{g_1}(x_1) \cdot f_{g_2}\phi_{g_2}(x_2) \cdot \tau(g_3)\phi_{g_3}(x_3) \cdot \tau(g_4)\phi_{g_4}(x_4) \cdot \tau(g_5)\phi_{g_5}(x_5) \tag{5}$$

where g_i, $i \in \{1, 2, 3, 4, 5\}$, indexes the three possible genotypes (AA, Aa, aa) possessed by family member i, and where $\tau(g \cdot)$ is short for $\tau(g \cdot | g_1, g_2)$. The likelihood function for a larger pedigree simply involves further terms and summations associated with each member.

It is important to note that, as with the departure of Eq. (2) from the paradigmatic form of the multivariate density function, Eq. (4) and, ultimately, Eq. (5) depart from the paradigmatic form of the normal mixture distribution by placing restrictions on the "mixing weights" [Titterington et al., 1985] of the implicated mixture distribution. These restrictions stem from the Hardy–Weinberg condition for founders and the Mendelian transmission probabilities for offspring.

Ott's model of the mixed effects of monogenic, polygenic, and environmental factors combines elements of Eqs. (2) and (5). The basic component of this combination centers around the summations used in Eq. (5) to express the possible genotype arrangements occurring among the pedigree members. These summations would generally encompass 3^n terms, where n is number of pedigree members, since every possible combination of the three genotypes over the n members would be implicated in the resulting "mixed" likelihood equation, much as they are in the single locus equation of (5). For each genotype arrangement (hereafter referred to as a "genotype vector", $\bar{g}$) one can assign a probability, $V_{\bar{g}}$ ($\bar{g} = 1, \ldots, n^3$), by multiplying the appropriate f_g and $\tau(g \cdot | g_m, g_f)$ terms. Thus, one possible genotype vector for a nuclear family consisting of two parents and three offspring is $\bar{g}^* = [1 = AA, 2 = Aa, 3 = AA, 4 = AA, 5 = Aa]$ with probability of occurrence $V_{\bar{g}^*} = (1 - p)^2 \cdot 2p(1 - p) \cdot 1/2 \cdot 1/2 \cdot 1/2$. By invoking Mendel's laws it becomes obvious that some genotype vectors will have probability $V_{\bar{g}} = 0$. For example, the genotype vector, $\bar{g}^- = [1 = AA, 2 = AA, 3 = AA, 4 = AA, 5 = Aa]$, has probability $V_{\bar{g}^-} = (1 - p)^2 \cdot (1 - p)^2 \cdot 1/2 \cdot 1/2 \cdot = 0 = 0$, since the mother $1 = AA$ and father $2 = AA$ genotypes will produce a gamete with the "a" allele (member $5 = Aa$) with probability 0. The number of "Mendelian" genotype vectors (i.e., those with probability of occurrence $V_{\bar{g}} > 0$) is $4 + 3^{n-2} + 2^n$ [Ott, 1979], which is considerably less than 3^n as n gets large, but can still be quite large. Lalouel et al. [1983], following the original line of thought given in Elston and Stewart [1971], consider treating the transmission probabilities as estimable parameters, the idea being that one could then compare, statistically, the model with the transmission probabilities fixed to their Mendelian values to one in which they were not so fixed in order to gauge the strength of the Mendelian assumption for the data. This strategy would obviously involve the consideration of all 3^n genotype vectors since no assumption of zero probability genotype vectors could be made (see Further Considerations in Mixed Model Evaluation for a discussion of issues pertaining to transmission probabilities and their estimation in mixed model settings).

The model assuming mixed monogenic, polygenic, and environmental effects implicates the parameters p, μ_{AA}, μ_{Aa}, μ_{aa}, σ_a^2, σ_d^2, σ_s^2, σ_e^2 previously described. The likelihood equation for this model for given pedigree trait values, x, becomes

$$L(p, \mu_{AA}, \mu_{Aa}, \mu_{aa}, \sigma_a^2, \sigma_d^2, \sigma_s^2, \sigma_e^2 \mid x) = \sum_{g=1}^{G(n)} V_{\bar{g}} \cdot \frac{1}{(2\pi)^{n/2} |\Omega|^{1/2}} e^{-1/2(x-\mu_{\bar{g}})'\Omega^{-1}(x-\mu_{\bar{g}})} \tag{6}$$

where x is the vector of trait values implicated in the pedigree, and $\mu_{\bar{g}}$ is a mean vector whose components are dictated by the arrangement of genotypes in the relevant genotype vector $\bar{g}$; for example, the mean vector associated with $\bar{g}^*$ above would be $\mu_{\bar{g}^*} = [\mu_{AA}, \mu_{Aa}, \mu_{AA}, \mu_{AA}, \mu_{Aa}]$, $G(n)$ refers to the number of genotype vectors to be used in the evaluation and as such depends on whether or not one wants to compute the "Mendelian" model [i.e., $G(n) = 4 + 3^{n-2} + 2^n$] or the "free transmission" model (i.e., $G[n] = 3^n$).

Note that Eq. (6) has inherited the constraints associated with Eqs. (2) and (5), as it represents a mixture of multivariate normal distributions whose mixing weights are restricted by the laws of Hardy and Weinberg and Mendel, whose mean vectors are restricted to different arrangements of at most three values, and whose covariance matrices must follow a specific pattern. It should be understood that extensions to these models are straightforward: one merely needs to add or delete covariance terms to create different variance models or extend the number of summations and mean vector components to allow for more alleles or loci.

SCALAR, VECTOR, AND PARALLEL NUMERICAL COMPUTATION

Computer Architectures

Consider the problem of adding two vectors, A and B, in a component-wise fashion, and storing the result in a vector, C. Familiar FORTRAN code for such a problem would be

```
DO 100 I = 1, N
100 C(I) = A(I) + B(I)
```

where the variable N is equal to the size of the vectors. A conventional scalar computer implementation of (7) would proceed by first fetching from memory the contents of components from vectors A and B and placing them in two different "registers," ultimately controlled by a "processor." The contents of these registers are then added together and the result is placed in another register. The content of this last register is then sent back to an appropriate memory location associated with components of the vector C. At this stage, the contents of the next two components from vectors A and B are fetched from storage, added together using registers, and sent back to memory. This process is repeated until all the components of A and B have been added together. The sequential or scalar-oriented nature of this computational process should be obvious.

A vector computer implementation of (7) is relatively straightforward. Here, the *scalar* registers, which are filled with relevant operands obtained from memory, are replaced by *vector* registers. Notice that all the components of vectors A and B are fetched from memory and placed in (vector) registers. The operations (here, addition) between the relevant components of vectors A and B are then done *simultaneously*, the results of which are placed in another vector register whose contents are then put back into memory locations associated with vector C. The time consuming, sequential fetch-register-add-return to memory process required for each of the N items in vectors A and B has been replaced by a single vector operation. Though something of an over-simplification, this description of vector computer operation does encapsulate the principal architectural advantages built into the design of vector computers. For an in-depth description of the use and features of vector computers, one should consult the excellent book by LeVesque and Williamson [1989].

The parallel computer implementation of code such as that in (7) would revolve around one very simple idea: parceling out subsets of relevant operands to different processors for simultaneous computation.

Of course, not every problem can be programmed in such a way that a vector computer would speed up its solution. Many problems involving "searches," sorting, or complicated functions are truly scalar in nature (i.e., sequential or order constrained) and would therefore not benefit from (or may actually be hindered by) implementation on a vector computer. In addition, not every program can make efficient use of parallel processors. For example, one may have a program that can only be broken down into subprograms, one of which takes longer to run than the others. Parallel implementation of such a program would result in some processors "waiting" (and hence not being utilized) for the longer subprogram to complete its tasks.

Mixed Model Algorithms

In this section we describe basic strategies one might use to take advantage of vector and parallel computers in evaluating mixed models. Since parallel implementation of mixed model evaluation is relatively straightforward, we discuss it first and then spend more time on vector computer implementation. Before this, however, we would like to draw attention to Table I which outlines some timing studies performed on various machines to assess the length of time it takes to compute one evaluation of the function given in Eq. (6) for various nuclear family sizes and numbers. The times given for the IBM 3090 parallel and vector processors were averages of 3 runs obtained using a program equipped with the computational strategies discussed below.

Two basic strategies exist for implementing patterned covariance matrix mixed models on parallel computers. The first involves parceling out $1/p$th, where p is the number of processors available, of the total families under study to each processor.

TABLE I. Elapsed Time (in Seconds) Needed on Various Machines to Compute One Iteration of a Patterned Covariance Matrix Mixed Model With a 2-Allele Codominant Major Locus, Additive Polygenic, and Environmental Effects for 50 and 200 Families of Sizes 4, 6, and 8*

	4		6		8	
	50	200	50	200	50	200
IBM XT mc	25.70	102.87	151.53	606.18	1187.76	4751.05
Everex 386	14.94	59.59	94.36	377.55	777.54	3102.24
IBM AT mc	11.64	46.68	71.78	287.40	575.39	2301.70
Sun 3/60	9.00	35.82	57.88	231.02	465.34	1855.62
Compaq 286 mc	8.51	34.10	52.45	209.87	420.78	1683.13
IBM PS/2 50 mc	7.63	30.37	45.86	183.45	364.43	1457.72
IBM PS/2 80 mc	3.46	13.78	20.43	81.78	159.61	638.28
Dell 310 mc	2.30	9.22	13.84	55.25	108.09	432.26
Dec 3100	0.22	0.85	1.19	4.84	10.42	40.19
IBM 3090 s	0.04	0.13	0.17	0.66	1.30	5.12
IBM 3090 2p	0.06	0.10	0.13	0.46	0.75	2.83
IBM 3090 4p	0.03	0.09	0.10	0.26	0.50	1.77
IBM 3090 6p	0.05	0.10	0.10	0.27	0.43	1.71
IBM 3090 v	0.03	0.11	0.08	0.34	0.50	2.08
IBM 3090 2vp	0.04	0.11	0.06	0.23	0.34	1.24
IBM 3090 4vp	0.05	0.06	0.06	0.16	0.21	0.76
IBM 3090 6vp	0.05	0.07	0.08	0.08	0.22	0.79

*mc, with math coprocessor; s, scalar mode; v, vector mode; 2p, 4p, 6p, number of processors.

Each processor is then responsible for computing log-likelihoods for each of the assigned families and computing their sum. Upon completion, the partial sums are then totalled to give the total log-likelihood. A second strategy would be to compute $G(n)$, the number of genotype vectors required for evaluation previously discussed and used in Eq. (6), for each family and parcel out $1/p$th of these $G(n)$ genotype vectors to each processor for evaluation. Upon completion of the evaluation of the $G(n)$ components of Eq. (6) on the different processors, the results are combined to form the log-likelihood of the relevant pedigree. This strategy is then repeated for each pedigree. The first strategy was used for the times reported on the parallel machines given in Table I.

Some comments about the times reported in Table I are in order. Ideally one would want to see a speed-up on the order of p for parallel algorithms over and above the scalar implementations of the same algorithm. That is, with two processors one would expect to see the program run in half as much time; with three processors, in one-third the time; etc. However, there is typically a ''cost'' associated with distributing tasks to different processors that, for small problems, may outweigh the speed-up due to the simultaneous processing performed. Another computational bottleneck for parallel machines stems from ''network noise'' due to multiusers. Though this bottleneck can be abolished by using a dedicated machine, it was most likely a source of speed reduction for our experiments on the much-used IBM 3090 housed at the Cornell National Supercomputer Facility reported in Table I. Thus, though the nonuniform or nonincremental speed-ups reported for the multiprocessor machines in Table I have an explanation, ideally (e.g., for problems with a large number of pedigrees or $G[n]$s) one would expect to see better performance gains.

Since our implementation of patterned covariance mixed models on a vector processor computer centers around the use of the ''string method,'' as opposed to the ''peel method,'' for pedigree model computation. The string method of pedigree likelihood evaluation involves spelling out all $G(n)$ components (or ''strings'') implicated in an equation like (6) and then merely summing them. The peel method, originally discussed by Elston and Stewart [1971] for nonmixed, major locus traits and whose algebraic basis was rederived by Cannings et al. [1978] for general settings, involves the use of conditional likelihoods to compute small segments (or ''peel groups''), possibly at the individual (i.e., pedigree member) level, of the pedigree likelihood individually and then combining them to form the total likelihood. Both methods have certain advantages. The peel method, by reducing the number of genotype vectors, $G(n)$, to $G(n^*)$ (where $n^* < n$) needed for a given peel-group evaluation, is much faster. However, by breaking up the pedigree one is necessarily sacrificing covariance terms between pedigree members. As a result, the peel method produces a likelihood value that is at best approximate [see Hasstedt, 1982]. For large pedigrees the string method may be impossible to compute, given the present state of computing machinery, so that strategies like that assumed in the peel method may have to be adopted. Of course, aspects of both methods can be combined (e.g., by using peel groups as large as possible and computing the likelihoods of these peel groups using the string method).

Our string-method oriented computational strategy for vector processors involves spelling out ''mean'' vectors relevant to each of the $G(n)$ components assumed in equations like (6) and then combining them to form giant matrix that can be used to evaluate the exponent(s) in Eq. (6). Since the pedigree covariance matrix, Ω (of Eq. 6), needs to be inverted only once for each pedigree (i.e., it does not change for each genotype vector) and the probability of each genotype vector can be computed prior to summation using estimated gene frequencies and Mendel's laws for offspring, the evaluation of (6) can be written to involve an explicitly vectorizable summation (or ''DO-LOOP'') over $G(n)$. That is, all the summed terms in (6) can be spelled out and stored in memory so that many of the component terms and operations needed for each of the $G(n)$ summands can be evaluated ''simultaneously'' through vectorization. The terms for the sum over $G(n)$ in (6) will then already be computed, making the summation trivial. Of course, the speed-up obtained using this strategy is dependent upon the size of the vectors used in the likelihood evaluation, and hence the size of the pedigree: if $G(n)$ is small, or the length of the mean vectors associated with each genotype arrangement (i.e., the number of pedigree members) is small, then the speed of the vector processor will not heavily outweigh the requisite fetches to memory needed to get each pedigree's data, nor will it outweigh the computational effort required for the covariance matrix formulation and inversion. This is reflected in Figure 1, where the speed of vectorized code on an IBM-3090 is given relative to optimimized scalar code for different pedigree-size likelihood evaluations. It clearly shows that speed increases only occur (outside of some noise-induced fluctuations) as a function of pedigree size and not number.

One problem with the implementation of the ''string'' method as discussed above is that it can be very memory intensive: if *all* $G(n)$ relevant genotype vector related structures (i.e., probabilities, mean vectors, etc.) must be spelled out and then run through the vector processor, one may easily run out of working (storage) computer memory to hold them all. To avoid this, one could work with subsets of the $G(n)$ genotype arrangements and try to balance the number of fetches to memory with memory size constraints to achieve some sort of efficiency. Figure 2 depicts relative speed

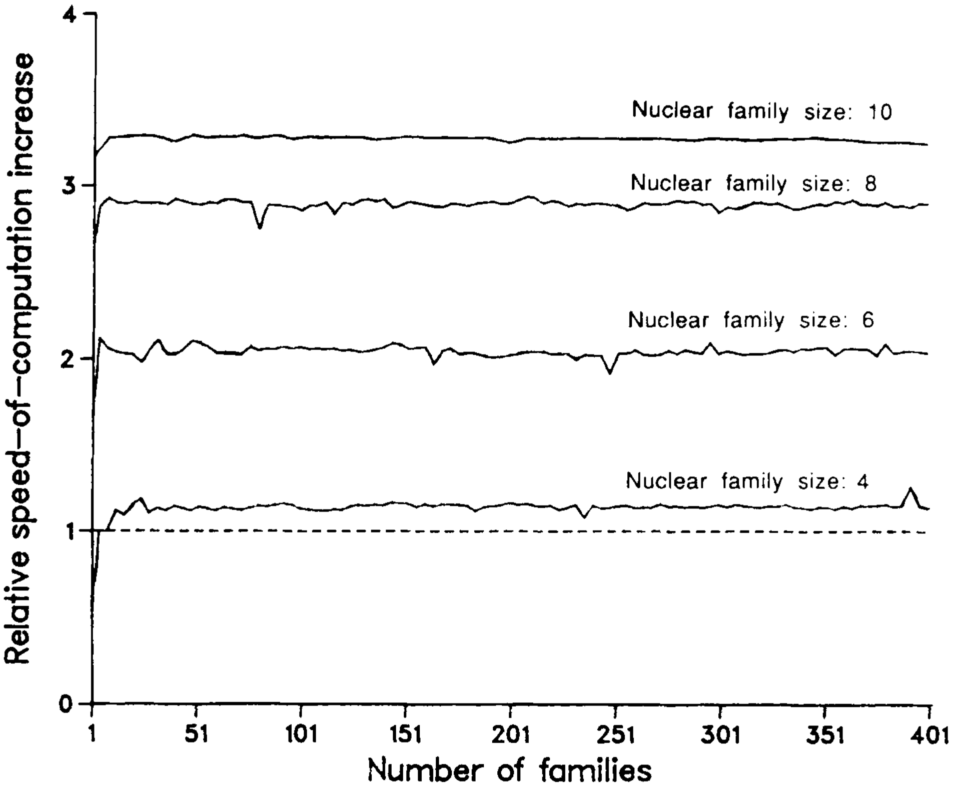

Fig. 1. Relative speed of computation increases for mixed model evaluation on a vector computer, assuming various nuclear family sizes and numbers. The dashed line represents time-identity, or virtual nonincrease in computational speed.

increases for mixed model evaluation associated with the use of different sizes of subsets of the $G(n)$ genotype arrangements for 100 nuclear families of size 8 (i.e., $G[n] = 4 + 729 + 256 = 989$). An initial subset size of 5 was used since computations using one genotype arrangement at a time were exceptionally slow. Figure 2 shows that for both a CRAY Y-MP and an IBM 3090 the decrease of computation time resulting from the use of larger and larger subsets of the $G(n)$ possible (Mendelian) arrangements quickly reaches a "saturation" point. Notice that though the largest speed-ups occur when running vectorized code, some speed-up does occur when running scalar code as well. This is intuitive since in using larger subsets of the $G(n)$ genotype arrangements, one is most likely reducing costly memory fetches and computational overhead that occurs as a result of spelling out the genotype arrangements individually and in a sequential manner. The saturation exhibited most likely results from "Amdahl's Law" [LeVesque and Williamson, 1989], which suggests that speed-up for vector and parallel programs is limited by that portion of the program (however small) that can *only* be done in scalar.

Two further comments about Figure 2 should be made. First, the speed-up reported is a *relative* one (i.e., relative to a computation assuming a subset of $G[n]$ size of 5). One should not get the impression that the IBM 3090 running in vector mode is

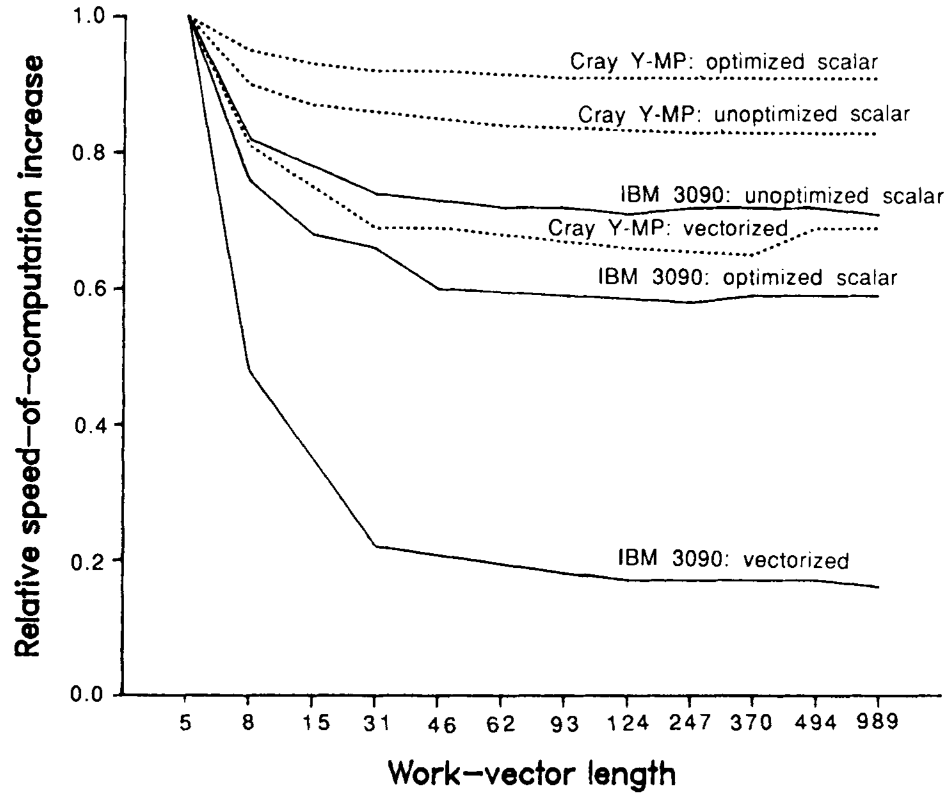

Fig. 2. Relative speed of computation increases for mixed-model evaluation on two different vector computers running in different modes, assuming different subset of total genotype arrangement (i.e., work-vector) sizes. Note that unlike Figure 1, speed-up is reflected in the *fraction* of time relative to computations involving a work-vector length of 5.

faster than the CRAY Y-MP running in vector mode. A uniprocessor Y-MP is generally three to five times faster than a uniprocessor IBM 3090 (see Dongarra [1989]). Second, the IBM 3090 is equipped with a special "memory cache" that can be used to great advantage by a programmer. Though we will not go into detail on cache architectures, it should suffice to say that a cache is a chunk of memory that resides between main computer memory and the registers to keep oft-used operands close to the registers for quick access. The presence of a cache impacted, no doubt, on the results depicted in Fig. 2.

THE IMPLEMENTATION OF MIXED MODELS

Proteus-G1: A Vector Designed Program

An experimental computer program, Proteus-G1, was designed and coded to evaluate patterned covariance matrix mixed models on an IBM 3090 with vector facility. This program uses the NPSOL optimization scheme ([Gill et al., 1986]—see section below) to compute maximum likelihood estimates of the segregation parameters. In order to test this program and gauge its efficiency and usefulness, we ran the program

TABLE II. Proportion of Models Chosen Based on the Akaike Information Criterion for the Data Sets Discussed in Konigsbeg et al. [1989]

	Analytical model			
Data set	Sporadic	Polygenic	Codominant	Mixed
Model 1 (sporadic)	10	0	0	0
Model 2a (polygenic)	0	9	1	0
Model 2b (polygenic)	0	9	0	1
Model 3 (codominant)	2	4	4	0
Model 4a (mixed)	0	1	2	7
Model 4b (mixed)	0	0	0	10

TABLE III. Average Parameter Estimates Obtained Across the 10 Replicate Data Sets Discussed in Konigsberg et al. [1989]

	h2	Gene frequency	Mean 1	Mean 2	Mean 3	Variance
Model 1 (sporadic)	—	—	20.02	—	—	18.25
Model 2a (polygenic)	0.57	—	21.01	—	—	26.93
Model 2b (polygenic)	0.60	—	−20.52	—	—	55.45
Model 3 (codominant)	—	0.80	19.52	25.56	31.00	49.84
Model 4a (mixed)	0.32	0.73	27.48	29.39	53.70	111.21
Model 4b (mixed)	0.36	0.68	14.98	23.83	33.14	10.55

TABLE IV. Average CPU Time in Minutes on an IBM 3090 With Vector Facility and Number of Function Evaluations (in Parentheses) Needed to Optimize Likelihoods for the Data Sets Described in Konigsberg et al. [1989]

	Sporadic	Polygenic	Codominant	Mixed
Model 1 (sporadic)	0.03 (32)	0.12 (149)	0.90 (156)	0.74 (127)
Model 2a (polygenic)	0.04 (43)	0.11 (139)	0.88 (147)	1.29 (209)
Model 2b (polygenic)	0.03 (39)	0.06 (79)	0.82 (149)	1.56 (273)
Model 3 (codominant)	0.05 (59)	0.10 (123)	1.30 (196)	1.36 (209)
Model 4a (mixed)	0.04 (52)	0.09 (108)	1.21 (212)	1.64 (287)
Model 4b (mixed)	0.04 (52)	0.06 (78)	0.87 (152)	0.87 (153)

on the data sets discussed in Konigsberg et al. [1989]. In order to obtain results comparable to those obtained by the programs investigated by Konigsberg et al., we evaluated sporadic, polygenic, codominant, and mixed models as described by Konigsberg et al. for each of the 10 replicate data sets assumed for each of the 6 transmission models discussed by them. Tables II and III show the models selected as the "best" using Proteus-G1 for each data set and the average parameter estimates obtained for the 6 models, respectively. These tables reflect a close agreement to the results obtained with PAP [Hasstedt and Cartwright, 1981] by Konigsberg et al. Table IV reports the average CPU time and number of function evaluations (including those used for numerical gradient evaluations) using Proteus-G1 for each of the 6 models. Table IV suggests that mixed model evaluation requires 30 times the computational effort it takes to evaluate sporadic models. To compute real time estimates for patterned covariance matrix mixed evaluation on different machines, one can merely multiply an average number of function evaluations entry in Table IV with the times given in Table I.

It should be kept in mind that scalar computer implemented patterned covariance models can be slow (compare Table VI of Konigsberg et al. [1989]) compared to other scalar computer implemented mixed models (e.g., the algebraic scheme used by Hasstedt [1982] or the regressive scheme of Bonney [1984]). This is because patterned covariance models require computations such as matrix inversions, inner product calculations, etc. that are costly computationally. The advantages of patterned covariance models are twofold: First, as previously emphasized, they are extremely flexible and intuitive; one can merely add or delete variance terms (e.g., Eq. 1) as the need arises without facing agonizing analytic reformulation. Second, also as previously emphasized, patterned covariance mixed model formulations are compatible with, and implementable on, machines that can greatly reduce their computational demands.

Parameter Estimation

As indicated earlier, estimation of the mixed model parameters involves maximization of Eq. (6). Since analytic derivatives for each parameter are (or would be) extremely difficult to obtain from (6), various numerical schemes were implemented and compared for their efficiency in determining MLEs of the mixed model parameters on 10 sample data sets. Since numerical schemes that do not rely on analytically derived derivative information are very slow by nature, it is imperative that the function to be evaluated is coded and evaluated in the fastest and most efficient manner possible. As emphasized throughout this paper, the necessary speed and efficiency can be achieved through the use of a supercomputer. The numerical schemes investigated included the scheme outlined by Box [1965], the simplex procedure of Nelder and Mead [1965], the quasi-Newton schemes inherent in the programs GEMINI [Lalouel, 1979] and QUASINEWTO (N.J. Schork, unpublished program), and the sequential quadratic programming design implemented in the NPSOL package [Gill et al., 1986]. GEMINI incorporates a sophisticated line search for step lengths (see Gill et al., [1981] for an explanation) but uses fixed length finite-difference intervals in approximating derivatives. QUASINEWTO, on the other hand, uses a simple linear search for step length determination but computes numerically optimal finite-difference interval lengths. Thus, these two programs were examined for the trade-offs between optimal numerical step length and derivative computations. Table V compares the efficiency of the various procedures in determining MLEs for 100 nuclear families of size 5 (i.e., with 3 offspring) with a vectorized segregation function code of the type previously described to evaluate the mixed model function. As expected, the routines that rely solely on function evaluations (the Box and simplex procedures) spend less time per function evaluation than the other routines (e.g., because basically all these routines have to do is evaluate the fast, vectorized segregation function and *not* invert Hessians, compute numerically stable step lengths, etc.). Also of interest is the intuitive finding that, though computation of optimal step-sizes (i.e., the GEMINI results, Table V) results in fewer function evaluations, the overhead required to perform the nonfunction evaluation related computations necessary for the optimal step-size calculations results in a larger time-per-function evaluation index. In addition, computing optimally estimated derivatives via finite-differencing (i.e., via QUASINEWTO, see Table V) necessarily takes a great many function evaluations. The NPSOL package is clearly the most efficient in computing MLEs, as it takes, on average $(35.2/1000) \cdot 179.9 = 6.3$ CPU (*not* real-

TABLE V. A Comparison of Some Optimization Programs Requiring Only Function Evaluations in Finding MLEs for the Mixed-Model Using Vectorized Code on 100 Nuclear Families of Size 5*

	Box		Simplex		Gemini		Q-newt		NPSOL	
Data set	T/F	Like	T/F	Like	T/F	Like	T/F	Like	T/F	Like
1	35.03	−667.53	34.89	−666.53	63.29	−666.52	36.31	−664.25	35.30	−666.52
2	34.85	−687.55	34.92	−687.55	40.19	−687.55	35.62	−687.55	35.37	−687.55
3	34.82	−699.07	34.86	−698.35	38.45	−698.33	37.25	−698.33	35.11	−698.33
4	34.88	−667.49	34.87	−668.91	38.09	−666.44	37.55	−666.44	35.30	−666.44
5	34.86	−667.32	34.84	−676.39	39.48	−676.34	37.04	−676.34	35.36	−676.34
6	34.85	−679.46	34.98	−679.46	69.03	−679.46	36.17	−679.46	35.12	−679.46
7	34.73	−672.05	35.01	−672.05	54.32	−672.05	36.45	−673.00	35.11	−672.05
8	34.74	−691.00	34.96	−691.00	41.46	−691.00	37.00	−691.00	35.21	−691.00
9	34.70	−677.83	35.02	−679.92	38.45	−679.92	37.10	−679.92	35.04	−679.92
10	34.81	−687.77	35.00	−687.77	62.21	−687.77	37.20	−687.77	35.09	−687.77
AVE	34.83/626.20		34.93/367.90		48.50/183.70		36.77/532.50		35.20/179.90	
G/S	6/3		5/2		8/1		8/2		8/1	

*Box, Box's modified simplex; Simplex, Nelder/Mead Simplex; Gemini, a quasi-Newton program; Q-Newt, a quasi-Newton program; NPSOL, a sequential quadratic program; T/F, time per function evaluation (in CPU milliseconds); Like, maximum likelihood achieved; AVE, average T/F ratio/average number of function evaluations; G/S, number of simulations in which the program found the greatest/smallest likelihood value (includes ties only for greatest values).

time) seconds on an IBM 3090 with vector facility to find them. Of course it is possible to restructure optimization algorithms to run faster on supercomputers. For instance, it is entirely possible to vectorize Hessian matrix inversion for some quasi-Newton schemes. The intent of the comparisons outlined here was to investigate the properties of *extant*, inherently scalar, optimization algorithms that make use of the refinements in the Likelihood equation computation previously outlined and not to make claims about inherent deficiencies of particular optimization strategies.

Further Considerations in Mixed Model Evaluation

The numeric nature of the computation of likelihoods and parameter estimates assumed in equations like (6) invites a number of practical problems apart from computational time or computer memory demands, in that the accuracy of the results defies easy assessment. Simulation studies like the one carried out by Konigsberg et al. [1989] and the one discussed in Computer Architectures are by nature indirect. One way of directly assessing the accuracy of mixed-model schemes involves simulated data, but works though the possible genotype arrangements a pedigree might have. Thus, one can simulate data and thereby know at once the genotype possessed by each simulated individual. By comparing these known genotypes with those predicted by a mixed-model scheme, one can effectively gauge the accuracy of the mixed-model scheme in question. Of course, the resulting accuracy indicator depends to a great extent on the genotype prediction scheme used. We chose the following prediction scheme for its ease of implementation and intuitive appeal: simply choose the genotype vector with the greatest likelihood and assign individuals genotypes using this vector. That is, after the mixed-model parameters have been reliably estimated, choose the genotype arrangement associated with the largest term in the sum (of $G[n]$ components) in an equation like (6). This strategy has the advantage that it predicts individual geno-

TABLE VI. Average Proportion of Genotypes Predicted Correctly Given Different Segregation Parameter Configurations and Family Data Structures, for 100 Simulations for Each Setting*

			Families (offspring)			
p	sep	H	50 (3)	50 (4)	100 (3)	Column average
.25	1	.5	.476	.477	.642	.532
.25	1	.25	.509	.624	.601	.578
.25	2	.5	.847	.738	.823	.803
.25	2	.25	.766	.579	.840	.728
.05	1	.5	.485	.892	.781	.719
.05	1	.25	.490	.875	.877	.747
.05	2	.5	.850	.901	.897	.883
.05	2	.25	.870	.821	.899	.863
Row average			.662	.735	.795	

*p, allele frequency; sep, separation between mean genotype effects (assuming codominance); H, polygenic heritability.

types using data from the entire pedigree. Using this strategy, we performed a simple Monte Carlo study of the accuracy of Proteus-G1. We generated 100 data sets following certain mixed-model configurations assuming two codominant alleles, computed parameter estimates for each, and then compared predicted and actual genotypes. The results are given in Table VI. It is obvious from Table VI that greater separation between mean genotype effects results in better prediction. It is also obvious, and intuitively reasonable, that larger samples result in parameter estimates with better predictive power, and that lesser polygenic "noise" (i.e., smaller heritabilities) results in better monogenotype prediction. Thus, genotype prediction-based assessments of accuracy may be useful for the evaluation of mixed-model programs, especially when approximations are used, as in Hasstedt [1982] and Bonney [1984].

Transmission probabilities play an important role in the statistical inference of genetic mechanism for qualitative traits, so it is not surprising that they should be considered useful for quantitative traits as well. Despite their theoretical and intuitive appeal, however, the estimation of transmission probabilities, when estimated alone or along with other parameters assumed in mixed models, is notoriously difficult (or at least our experiences have indicated this) and invites opposition to their use. An important question is then whether or not one could do without them; or rather, whether one can just work with mixed models with transmission probabilities "fixed" to their Mendelian values. In order to address one aspect of this question we performed a small Monte Carlo experiment. We generated 100 nuclear family data sets following Mendelian patterns with different allele frequencies and mean genotype effect separations, but assumed a heritability of 0.25 throughout. For these same allele frequency, separation, and heritability settings we generated nuclear family data following *equal* transmission probability patterns. For each type of data we computed the number of times a Mendelian mixed model and/or a single locus model without polygenes better characterized the data than a polygenic or sporadic model based on Akaike's Information Criterion [1971] and Schwarz's more conservative Bayesian Criterion [1978]. It should be kept in mind that *none* of the models compared actually involved the estimation of transmission probabilities: the sporadic and polygenic models simply require $p = 0$

TABLE VII. Proportion of Mendelian Single-Locus 2-Allele Codominant Mixed (Total Mixed and Nonmixed in Parentheses) Segregation Models Accepted Over Sporadic and Purely Polygenic Models Using Two Different Model Selection Devices for Mendelian and Equal Transmission Probability Data*

		SBC		AIC	
p	sep.	Mend.	Equal	Mend.	Equal
.25	1	.00 (.07)	.00 (.03)	.29 (.64)	.10 (.36)
.25	2	.34 (.98)	.00 (.09)	.73 (1.0)	.06 (.42)
.05	1	.00 (.02)	.00 (.03)	.23 (.46)	.08 (.48)
.05	2	.30 (.71)	.09 (.29)	.80 (.95)	.50 (.78)

**p*, allele frequency; sep, separation between mean monogenotype effects; SBC, Schwarz's Bayesian criterion; AIC, Akaike's information criterion; Mend., Mendelian transmission data; Equal, equal transmission probability data.

and the major locus and mixed models *assumed* Mendelian transmission and Hardy–Weinberg equilibrium. The results are outlined in Table VII, and suggest that although the likelihood of accepting a Mendelian major gene model is reduced when the Mendelian transmission assumption is violated, this reduction is not large enough to obviate the need for transmission probability assessment in the special case of equal transmission probability. Similar results were obtained in the case study of qualitative traits by McGuffin and Huckle [1990]. We are working on the problem of whether or not one needs to estimate the transmission probabilities, or merely needs to evaluate models with them fixed to certain values (other than Mendelian) to provide sufficient power against non-Mendelian data types. It should be emphasized that transmission probability estimation can be easily implemented in vectorized or parallelized code.

It is also important to emphasize that the questions of accuracy and transmission probability use posed in this section had not been addressed before simply *because* of their very great computational demand. These demands were lightened in our case because we used a supercomputer. Of course there is no reason to stop supercomputer use at the power study or computational efficiency level. Supercomputers can provide efficient environments for testing hypotheses (see Schork and Hardwick [1990]; Schork [1990]; Schork and Schork [1989]) or the modeling of truly multivariate traits, such as the hyperkinetic state described in Schork et al. [1990]). As such, supercomputers have a potential in statistical genetics that warrants serious attention.

CONCLUSION

In a recent editorial Ott [1990] justifiably suggested that extant segregation modeling devices are not powerful enough to sort out genetic mechanisms responsible for complex traits, and that therefore geneticists should look more to linkage or gene marker-based strategies for infering genetic mechanism in traits with unknown genetic etiologies. This lack of power stems, in part, from computational and estimation difficulties that linkage methods do not possess. On the other hand, it is becoming more and more obvious that the pardigm and tools assumed in many linkage-based and "bottom-up" strategies are unequivocally limited (see, for example, Friedmann [1990] or Sing [1990]). Thus, there appears to be an impasse: good genetic research tools exist, but only for a small class of traits. Supercomputers and efficient computational schemes may provide a way out of this impasse, in that they may allow both the reliable construction and implementation of more complex, and hence more realistic, genetic modeling devices.

ACKNOWLEDGMENTS

Pamela Minick helped type the manuscript. Lyle Konigsberg supplied test data sets generated using simulation programs written for Genetic Analysis Workshop 1. Robb Muirhead and Mark Snow helped establish the necessarily large computer accounts. Len Harding taught me about supercomputers. Stevo Julius helped instill in me an appreciation for the human medical sciences. Alan Weder and Michael Boehnke taught me genetics. Finally, my father, M.A. Schork, was a constant source of inspiration and an outlet for my obscure and often obtuse questions.

REFERENCES

Akaike H (1974): A new look at statistical model identification. IEEE Transact Automatic Control 19:716–723.

Bonney GE (1984): On the statistical determination of major gene mechanisms in continuous human traits: Regressive models. Am J Med Genet 18:731–749.

Box MJ (1965): A new method of constrained optimization and a comparison with other methods. Comput J 8:42–52.

Cannings C, Thompson EH, Skolnick MH (1978): Probability functions on complex pedigrees. Adv Appl Prob 10:26–61.

Dongarra JJ (1989): Performance of various computers using standard linear equations software. Argonne National Laboratory Report MSCO-TM-23m, October.

Edwards AWF (1972): "Likelihood." London: Cambridge University Press.

Edwards J (1982): The user of computers. Cytogenet Cell Genet 39:43–51.

Elston RC (1981): Segregation Analysis. In Harris H, Hirshhorn K (eds): "Advances in Human Genetics." New York: Plenum: 63–120.

Elston RC, Stewart J (1971): A general model for the genetic analysis of pedigree data. Hum Hered 21:323–342.

Fisher RA (1918): The correlation between relatives on the supposition of Mendelian inheritance. Transac Soc Edinburgh 52:399–433.

Friedmann (1990): The human genome—some implications of extensive "reverse genetic" medicine. Am J Hum Genet 46:407–414.

Gill PE, Murray W, Saunders MA, Wright MH (1984): User's guide for SOL/NPSOL: A Fortran package for non-linear programming. Stanford University, Department of Operations Research, Report SOL 83-1.

Gill PE, Murray W, Wright MH (1981): "Practical Optimization." New York: Academic Press.

Hasstedt SJ (1982): A mixed-model likelihood approximation on large pedigrees. Comput Biomed Res 15:295–307.

Hasstedt SJ, Cartwight PE (1981): PAP: Pedigree analysis package. Technical report no. 13, Department of Biophysics and Computing, University of Utah, Salt Lake City.

Jacquard A (1974): The Genetic Structure of Populations. New York: Springer.

Konigsberg L, Kammerer C, MacCluer J (1989): Segregation analysis of quantitative traits in nuclear families: Comparison of three program packages. Genet Epidemiol 6:713–726.

Lalouel JM (1979): GEMINI—A computer program for optimization of general nonlinear functions. University of Utah, Department of Biophysics and Computing, Technical Report 4.

Lalouel JM, Rao DC, Morton NE, Elston RC (1983): A unified model for complex segregation analysis. Am J Hum Genet 26:484–503.

Lange K, Westlake J, Spence MA (1976): Extensions to pedigree analysis. III. Variance components by the scoring method. Ann Hum Genet 39:485–491.

LeVesque JM, Williamson JW (1989): "A Guidebook to FORTRAN on Supercomputers." New York: Academic Press.

McGuffin P, Huckle P (1990): Simulation of Mendelism revisited: The recessive gene for attending medical school. Am J Hum Genet 46:994–999.

Morton NE (1967): The detection of major genes under additive continuous variation. Am J Hum Genet 19:23–34.

Morton NE, MacLean CJ (1974): Analysis of family resemblance. II. Complex segregation of quantitative traits. Am J Hum Genet 26:489–503.

Nelder JA , Mead R (1965): A simplex method for function minimization. Comput J 7:308–313.

Ott J (1979): Maximum likelihood estimation by counting methods under polygenic and mixed models in human pedigrees. Am J Hum Genet 31:161–175.

Ott J (1990): Cutting a Gordian knot in the linkage analysis of complex human traits. Am J Hum Genet 46:219–221.

Pearson K (1893): Contributions to the mathematical theory of evolution. I. On the dissection of asymmetrical frequency curves. Phil Transact A185:71–110.

Pearson K (1895): Contributions to the mathematical theory of evolution: II. Skew variation in homogenous material. Phil Transact A186:343–414.

Schork NJ (1990): Bootstrapping likelihood ratios in quantitative genetics. ''The Proceedings of the 214th Meeting of the Institute of Mathematical Statistics: Special Topics Conference on the Bootstrap.'' New York: John Wiley & Sons, in press.

Schork NJ, Hardwick JP (1990): Supercomputer-intensive multivariable randomization tests. ''The Proceedings of the 22nd Symposium on the Interface of Computing Science and Statistics.'' New York: John Wiley & Sons, in press.

Schork NJ, Schork MA (1989): Testing separate families of segregation hypotheses: Bootstrap methods. Am J Hum Genet 45:803–813.

Schork NJ, Weder AB, Schork MA, Bassett DR, Julius S (1990): Disease entities, mixed multi-normal distributions, and the role of the hyperkinetic state in the pathogenesis of hypertension. Statist Med 9:301–314.

Schwarz G (1978): Estimating the dimension of a model. Ann Statist 6:461–464.

Sing C (1990): Review of Douglas Falconer's *Introduction to Quantitative Genetics*, 3rd edition. Am J Hum Genet 46:1231–1233.

Titterington DM, Smith AFM, Makov UE (1985): ''Statistical Analysis of Finite Mixture Distributions.'' London: John Wiley & Sons.

Edited by G.P. Vogler and D.C. Rao