



Mendel's Accountant: A New Population Genetics Simulation Tool for Studying Mutation and Natural Selection

John Baumgardner, Ph.D., Institute for Creation Research, 1806 Royal Lane, Dallas, TX 75229

John Sanford, Ph.D., Department of Horticultural Science, NYSAES, Cornell University, Geneva, NY 14456

Wesley Brewer, Ph.D., Department of Computer Science, Handong University, Pohang, South Korea

Paul Gibson, Ph.D., Department of Plant, Soil, and Agricultural Systems, Southern Illinois University, Carbondale, IL 62901

Walter ReMine, M.S., 3003 Snelling Ave. N., St. Paul, MN 55113

Abstract

Mendel's Accountant (hereafter referred to as "Mendel") is a state-of-the-art forward-time population genetics model that tracks millions of individual mutations with their unique effects on fitness and unique location within the genome through large numbers of generations. It treats the process of natural selection in a precise way. It allows a user to choose values for a large number of parameters such as those specifying the mutation effect distribution, reproduction rate, population size, and variations in environmental conditions. Mendel is thus a versatile and capable research tool that can be applied to problems in human genetics, plant and animal breeding, and management of endangered species. With its user-friendly graphical user interface and its ability to run on laptop computers it can also be fruitfully employed in teaching genetics and genetic principles, even at a high school level. Mendel is freely available to users and can be downloaded from the web.

When biologically realistic parameters are selected, Mendel shows consistently that genetic deterioration is an inevitable outcome of the processes of mutation and natural selection. The primary reason is that most deleterious mutations are too subtle to be detected and eliminated by natural selection and therefore accumulate steadily generation after generation and inexorably degrade fitness.

Keywords

Genetic fitness, Genome, Mutation, Mutation effect, Population genetics, Natural selection, Numerical simulation

Introduction

The traditional approach that population geneticists have used for more than 75 years to understand how mutation and selection affect population dynamics is hand solution of analytical equations. However, recent advances in numerical simulation and the wide availability of low cost computational resources now make possible an alternative way to understand how populations change over time. Numerical simulation offers the ability to treat complex biological situations where an analytical solution would be cumbersome, if not impossible. Numerical simulation allows the study of the complex interactions of many biological factors simultaneously. This is generally not practical using traditional methods. The numerical approach provides great flexibility and allows a researcher or student to explore parameter space quite rapidly, without detailed knowledge of the many specialized mathematical techniques that underlie the classical theoretical approach.

At its most basic level, the task of modeling

mutation and selection in a population over many generations can be viewed as a bookkeeping problem in which random events play a major role. Mutations are continuously entering and leaving any population. When a new mutation arises, it may or may not be transmitted to an individual's progeny, depending on whether or not the chromosome segment carrying the mutation segregates into the gamete from which the progeny is derived. Generally speaking, mutations that occur near one another on the same chromosome are likely to be inherited together. Therefore, tracking mutation location in the genome is important if one desires to account for mutational linkage. In addition, in most higher organisms during meiosis there are about two crossovers per chromosome pair (Santiago & Cabellero, 2000). This random phenomenon of crossover also must be part of the simulation in order to treat linkage in a realistic manner.

Random mutations tend to differ greatly from one another in their effects on genotypic fitness. The fitness effect of a given mutation can be positive or

negative, can range from lethal to beneficial, and can vary from fully dominant to fully recessive. How the effects of multiple mutations (at different loci within the same individual) combine with one another (additively or multiplicatively) also influences the overall genetic fitness of an individual. The effectiveness of selection (that is, its power to alter individual mutation frequencies) is limited by the surplus population available, which in turn depends on the population's average fertility level. Selection efficiency is further limited by factors such as random fluctuations in environmental conditions. Generally speaking, reproduction in nature has a significant random component and is only partially correlated with the fitness of the genotype. All these variables influence actual genetic change over time and must be modeled accurately if a simulation is to be biologically relevant.

Approach

Although there are many programs for genetic data analysis, comparatively little effort has been devoted to software development for detailed simulation of the processes of mutation and selection (Balloux, 2001). Numerical strategies for population genetics modeling have been under discussion for several decades (Crosby, 1973; Fraser & Burnell, 1970), yet it is only recently that computing resources have become widely available to allow large realistic forward-time simulations. The forward-time approach offers the distinct advantage of being able to treat random mutations and natural selection under complex

mating/recombination scenarios.

Mendel represents an advance in forward-time simulations by incorporating several improvements over previous simulation tools:

(1) Mendel adds the ability to model mutations as having a continuous, natural distribution of mutation effects.

(2) Mendel allows a user-specified ratio of dominant to recessive mutations.

(3) Mendel uses an infinite sites model, where segregating mutations are distinct and their number is unlimited (or limited only slightly by computer capacities).

(4) Mendel incorporates the concept of heritability and accounts for environmental variance.

(5) Mendel uses realistic chromosome structure with realistic stochastic crossover and recombination, and a high number of linkage blocks (up to order 10^5). Users can specify the number of chromosome pairs.

(6) Mendel is tuned for speed-efficiency and memory usage to handle large populations and high mutation rates.

(7) Mendel allows control of genetic parameters via a graphical user interface (Figure 1), thereby allowing non-programmers to construct sophisticated simulations.

(8) Mendel provides several forms of graphical output, allowing the user to see the results as the simulation proceeds (Figure 2 shows one of the plots).

In addition, Mendel provides a variety of options for mating, bottleneck events, and population

The screenshot displays the web user interface for Mendel's Accountant. The title is "Mendel's Accountant" in a large blue font. Below the title, there is a note: "Note: using parameters from case01". A dropdown menu for "Choose parameter template..." is set to "case01".

The interface is divided into several sections:

- Control Panel (Left):** Includes buttons for "Start", "Clear", "Inputs", "Output", "List files", "Plot", "List cases", "Report", "More>>", and "Stop". A table shows the current case status:

Case	case01	<input type="radio"/>
Gen	4223 (84.5%)	
Fit	0.8323057	
Mutns	40024 D / 6F	
Time left	20.0 minutes	
Case	test02	<input type="radio"/>
- Basic parameters (Middle):** A list of seven parameters with input fields:
 - Case ID: case01
 - New mutations per offspring: 10.0
 - Fraction of mutations beneficial: 0.0001000
 - Offspring per female: 6.0
 - Population size: 1000
 - Generations: 5000
 - Advanced settings: Submit
- Advanced settings (Bottom):** A section titled "Advanced:: mutations. selection. population. computation" with three sub-sections:
 - Parameters shaping distribution of deleterious mutation effects:
 - genome size: 3.000e+09
 - fraction of mutations have "major effect": 0.0010
 - minimum mutation effect which is defined as "major": 0.100
 - Range and distribution of beneficial mutations:
 - maximal beneficial effect per mutation: 0.1000000

Figure 1. Web user interface of Mendel's Accountant showing a portion of the input window.

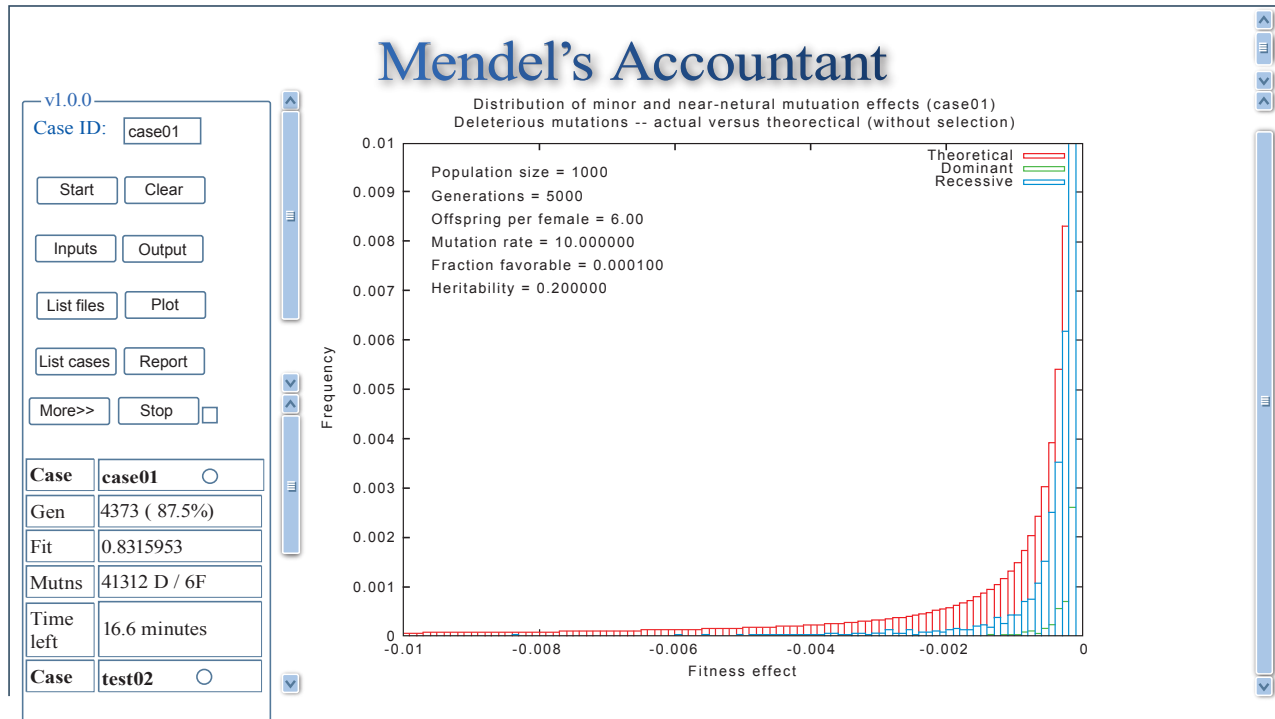


Figure 2. Web user interface of Mendel's Accountant showing one of the several output plots the program generates. This plot displays the distribution of deleterious mutations with respect to fitness effect. Red bars represent mutation distribution in the absence of selection. Blue and green bars represent actual accumulated recessive and dominant mutations, respectively, in the presence of selection. The two bars representing mutation classes with effects nearest zero extend beyond the vertical scale of the plot.

substructure. It is computationally efficient, allowing many problems of interest to be run on ordinary personal computers. In addition, because Mendel is parallelized with MPI (Message Passing Interface), it can exploit multiple processors to run: (a) multiple interacting heterogeneous tribes (b) multiple replications of a single case, or (c) a very large population comprised of sub-populations but with sufficient migration to maintain a high degree of genetic homogeneity.

Numerical Strategy

In each generation, Mendel first performs migration between tribes, then mating, then creation of offspring, with new mutations potentially introduced in each offspring's genome. Selection is applied as a final step to reduce the number of offspring that survive to reproduce in the succeeding generation. Although the overall structure is relatively straightforward, much care has been taken in representing and tracking the individual mutations, as we shall now discuss.

Representing and tracking mutations

In designing this numerical model, we endeavored to combine a high degree of biological realism with a high level of flexibility for investigating diverse population scenarios. To achieve this realism and flexibility, we choose to track, when desired, each

germ line mutation in every individual in each generation. We recognized that to track millions of individual mutations in a sizable population over many generations, efficient use of memory would be a critical issue – even with the large amount of memory commonly available on current generation computers. We therefore selected an approach that uses a single 32-bit (four-byte) integer to encode a mutation's fitness effect, its location in the genome, and whether it is dominant or recessive. Using this approach, given 1.6 gigabytes of memory on a single microprocessor, we can accommodate at any one time some 400 million mutations. If our maximum population size is, for example, 10,000, then the maximal number of mutations in any individual is 40,000. This implies that, at least in terms of memory, we can treat reasonably large cases using a single processor of the type found in many desktop computers today. In fact, typical laptop computers have sufficient memory to run many problems of interest with Mendel, especially in instructional contexts.

In terms of implementation, we use separate four-byte integer arrays to store favorable and deleterious mutations for all current members of the population. The sign of the integer is utilized to mark whether the mutation is dominant or recessive. The less significant part of the integer is used to encode the mutation's fitness effect, while the more significant

part is used to encode the mutation's location in the genome. The modulo function is employed to extract an integer from which the mutation's fitness effect can readily be computed, while a single multiplication yields the mutation's location in the genome in terms of the linkage subunit on which the mutation resides.

The mutations carried by each individual occur within its two versions of the haploid genome, one inherited from each of that individual's parents. Each haploid genome is divided into a user-specified number of linkage subunits. A linkage subunit represents a small physical interval along a chromosome within which no recombination occurs. In meiosis, one member of each linkage subunit pair from the two homologous chromosomes is randomly selected, with all its associated mutations, and is inherited by the gamete. If linkage is specified to be static, all linkage subunits are inherited independently of one another. However, if the user specifies dynamic linkage, many contiguous subunits that reside together on a larger portion of a chromosome are jointly transferred. In dynamic linkage, we assume that exactly two crossovers occur for each chromosome pair, with the random crossover locations constrained to lie at linkage subunit boundaries. Because crossover locations are random, they almost always occur at different points along each chromosome from one generation to the next.

It is useful here to explain the connection between the term "linkage subunit" and the term "haplotype" as normally used by geneticists. In Mendel, the full set of mutations carried by a given linkage subunit in a given individual constitutes a haplotype. Across a population, for each linkage subunit location in the genome there will normally arise multiple haplotypes, with each haplotype consisting of a different distinct set of mutations that are inherited together.

Finally, it is during gamete formation that the new mutations are added. Following the computational steps associated with mating and reproduction, the memory used to store the mutation information for the parenting generation is overwritten with the mutation information for the offspring.

From this brief description it should be clear that a basic aspect of the numerical code is the bookkeeping which tracks each individual mutation within each of the members of a population from one generation to the next. Mendel has been designed to make efficient use of available memory to be able to track extreme numbers of mutations. Mendel was also designed to limit the amount of computation required so as to enhance execution speed.

Prescribing fitness effects of mutations

Because of the nature of genomic information and the many ways mutations can alter it, mutations vary

in their influence on the organism from occasionally beneficial to almost neutral to lethal. The realism of any population genetics model depends critically on the manner in which mutations are assumed to alter fitness. Selecting a distribution of mutational effect that matches biological reality is therefore a crucially important issue. The ability to represent effects that vary over a wide range of amplitude is especially important to be able to treat nearly neutral mutations in a proper manner. This generally requires the range of the distribution function to span many orders of magnitude. Since nearly neutral mutations occur at vastly higher frequencies than do mutations that have large impacts on fitness, previous investigators have employed exponential distributions (Kimura, 1979) that yield large numbers of small-effect mutations and small numbers of mutations with large effect.

To provide users of Mendel even more flexibility in specifying the fitness effect distribution, we have chosen to use a form of the Weibull function (NIST/SEMATECH e-Handbook of Statistical Methods) that is a generalization of the more usual exponential function. Our function, expressed by Equation (1), maps a random number x , drawn from a set of uniformly distributed random numbers, to a fitness effect $d(x)$ for a given random mutation as follows:

$$d(x) = d_{sf} \exp(-ax^\gamma), 0 \leq x \leq 1 \quad (1)$$

Here d_{sf} is the scale factor which is equal to the extreme value which $d(x)$ assumes when $x=0$. We allow this scale factor to have two separate values, one for deleterious mutations and the other for favorable ones. These scale factors are defined relative to the initial fitness value assumed for the population before we introduce new mutations. In Mendel, we assume this initial fitness value to be 1.0. For deleterious mutations, since lethal mutations exist, we choose $d_{sf_del} = -1$. For favorable mutations, we allow the user to specify the (positive) scale factor d_{sf_fav} . Normally, this would be a small value (for example, 0.001 to 0.1), since it is only in very special situations that a single beneficial mutation would have a very large effect.

The parameters a and γ , both positive real numbers, determine the shape of the fitness effect distribution. We apply the same values of a and γ to both favorable and deleterious mutations. The parameter a determines the minimum absolute values for $d(x)$, realized when $x=1$. We choose to make the minimum absolute value of $d(x)$ to be the inverse of the haploid genome size G (measured in number of nucleotides) by choosing $a = \log_e(G)$. For example, for the human genome, $G = 3 \times 10^9$, which means that for the case of deleterious mutations, $d(1) = -1/G = -3 \times 10^{-10}$. For large genomes, this minimum value therefore becomes very close to zero. For organisms with smaller genomes

such as yeast, which has a value for G on the order of 10^7 (Goffeau et al., 1996), the minimum absolute effect is larger. This is consistent with the expectation that each nucleotide in a smaller genome on average must play a greater relative role in the organism's fitness.

The second parameter γ can be viewed as controlling the fraction of mutations that have a large absolute fitness effect. Instead of specifying γ directly, we select two quantities that are more intuitive and together define γ . The first is θ , a threshold value that defines a “high-impact mutation.” The second is q , the fraction of mutations that exceed this threshold in their effect. For example, a user can first define a high-impact mutation as one that results in 10% or more change in fitness ($\theta=0.1$) relative to the scale factor and then specify that 0.001 of all mutations ($q=0.001$) be in this category. Inside the program the value of γ is computed that satisfies these requirements. We reiterate that Mendel uses the same value for γ , and thus the same values for θ and q , for both favorable and deleterious mutations. Figure 3 shows the effect of the parameter q on the shape of the distribution of fitness effect. Note that for each of the cases displayed the large majority of mutations are nearly neutral, that is, they have very small effects. Since a mutation's effect on fitness can be measured experimentally only if it is sufficiently large, our strategy for parameterizing the fitness effect distribution in terms of the high-impact mutations provides a means for the Mendel user to relate the numerical model input more directly to available data regarding the actual measurable frequencies of mutations in a given biological context.

Details of encoding the genomic location and fitness effect of a mutation

In the preceding section we mentioned that a single four-byte integer is used to encode a mutation's type, its fitness effect, and its location in the genome. Some readers might like to know how we do this. First, as we have already mentioned, whether the mutation is dominant or recessive is encoded in the sign of the integer. Next, we choose an integer-valued modulus μ given by $2^{31} - 1 = 2,147,483,647$ (which is the largest value a four-byte integer can assume) divided by λ , the number of linkage subunits. For example, if λ is 2,000, then we choose $\mu = 1,073,741$. If we let the symbol σ be either 1 or -1 to denote whether the mutation is dominant or recessive and let m be the integer-valued mutational index used to represent the mutation, then our encoding formula for m is given by $m = \sigma[(l-1)\mu + \mu x]$, where l is the index of the linkage subunit on which the mutation occurs and x is the value of the random number, with $0.0 \leq x \leq 1.0$, that specifies the mutation's fitness effect. We apply the modulo function with modulus μ to the absolute value of m to recover x . We divide the absolute value of m by μ and use the int function to recover l .

The mutation indices just described are stored in ascending numerical order for each of the two versions of the haploid genome in each individual. This allows us to be able to readily test whether a given mutation is homozygous, that is, whether or not that particular mutation occurs on both copies of the individual's chromosomes. When a new mutation is introduced, the existing mutation

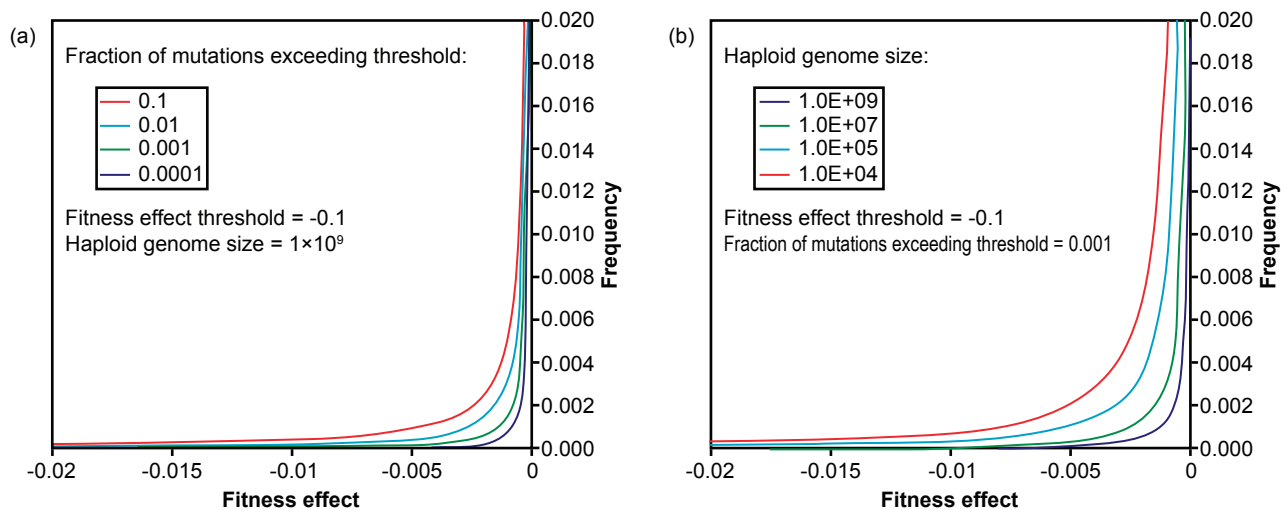


Figure 3. (a) Response of the fitness-effect distribution function to changes in the fraction of “high impact” mutations (0.0001 to 0.1). (b) Response of the fitness-effect distribution function to changes in the specified haploid genome size (number of nucleotides = 1×10^4 to 1×10^9). The graphs display only a small portion of these distributions, excluding the larger effect mutations (that extend off the scale to the left) as well as most mutations that have nearly zero effect (whose distributions plot beyond the top of the vertical scale). The vertical scale is the number of mutations per unit fitness effect, normalized to the maximum values.

indices are shifted within memory so that the index of the new mutation can be inserted in the appropriate location. Identifying homozygous mutations in a given individual involves scanning the two haploid genomes in numerical order and searching for matches. The user specifies both the proportion of mutations that are recessive and, for both recessive and dominant mutations, the fraction of the full (homozygous) effect to be expressed when the mutation is heterozygous. Mutations are assumed to be heterozygous unless found to be homozygous. In the latter case, an appropriate adjustment is applied. Since by definition a mutation that is recessive has a heterozygous effect less than half of the homozygous effect and a dominant mutation has a heterozygous effect greater than half, the adjustment will be positive for a recessive mutation and negative for a dominant mutation, accounting for the fact that the heterozygous effect has already been applied twice.

To calculate total fitness, Mendel offers three options for combining the effects of all the mutations within an individual. One, referred to as multiplicative fitness, multiplies together individual fitness effects of the form $(1-d_i)$ for all mutations, where d_i is the fitness effect associated with mutation i . A second option, referred to as additive fitness, simply sums the fitness effects d_i from all the mutations and subtracts this total from one. The third option is specifying the proportion of multiplicative effect, the remainder being additive.

To reduce the number of times the fitness effect function needs to be computed from the stored mutation index m , Mendel allocates a separate array to track the cumulative heterozygous fitness effects from all the mutations associated with each linkage subunit in each version of the haploid genome in each individual. When a new mutation is added in a zygote, its heterozygous fitness effect is incorporated into the composite fitness effect of the linkage subunit on which it occurs. Apart from certain diagnostic analyses, performed infrequently, this is the only time the fitness function needs to be evaluated, except in the infrequent cases of homozygosity, where an adjustment must be applied. Because linkage subunits are assumed to pass intact from parents to zygote, all the fitness information needed to describe the heterozygous fitness effects of all the mutations in a given linkage subunit is carried in a single number from this array. This number, along with the list of mutation indices for the linkage subunit, is transferred from parent to zygote. Homozygous effects are computed and added once the zygote is formed. In addition to reducing the number of times the fitness function needs to be computed, another benefit of this array is that, if desired, the mutation indices for very

low impact mutations need not be stored and tracked at all. The user may specify a fitness effect threshold, below which mutation indices themselves are not stored or tracked. Mendel accounts for the fitness effects of these very low impact mutations by incorporating their effect into the cumulative fitness value stored in the linkage subunit fitness array. Choosing a fitness effect tracking threshold of 0.000001, for example, typically results in about 70% reduction in storage and 30% less computation compared with tracking all the mutations (using a tracking threshold of zero). The drawback of this feature is that it does not account for the rare instances of homozygosity among these extremely low impact mutations. However, this error is negligible in most circumstances.

Mating and tribes

Mendel is presently limited to sexually reproducing diploid organisms. The default mode for mating is random pairing of selected individuals and monogamy. Alternatively, for certain organisms such as plants, the user can specify a fraction of self-mating (self-fertilization). In addition, Mendel offers the option of partitioning a population into a specified number of sub-populations (either homogenous or heterogeneous), which represent mating sub-groups. Mating occurs only among individuals within these sub-populations, or tribes, except that tribes can exchange, via migration, a specified number of individuals with neighboring tribes at specified generation intervals. Random monogamous mating is performed within each tribe following exchanges with the neighboring tribes.

Currently, Mendel offers three options for modeling the migration of individuals between tribes: (1) a one-way stepping-stone model, (2) a two-way stepping-stone model, and (3) an island model. These three migration models are illustrated in Figure 4 for the case of four tribes. The one-way stepping-stone model passes a user-specified number of individuals to only one neighboring tribe (in this case, the next process in the process list). The two-way stepping-stone model passes individuals to the two neighbors located on either side of the sending tribe. The island model passes individuals to every other tribe. In the two-way stepping-stone model, specifying one individual for inter-tribe migration implies that one individual will be sent to each neighboring process, such that a total of two individuals are sent from each tribe. Similarly, in the case of the island model, if the user specifies the number of migrating individuals to be one, each tribe will pass one individual to every other tribe, meaning that $NP-1$ individuals are sent out from each tribe, where NP is the total number of processes or tribes. It can be noted that for the case of two tribes, all three models perform migration

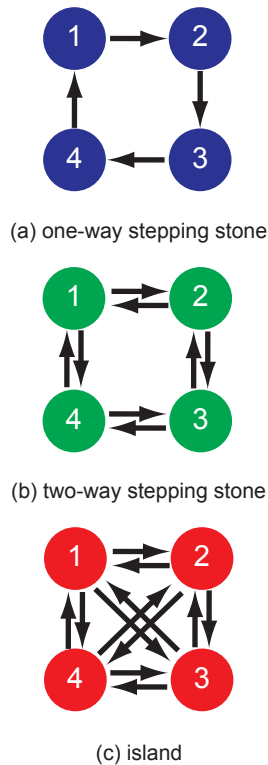


Figure 4. Migration models available in Mendel.

identically. Similarly, for the case of three tribes, the two-way stepping stone and island migration models are equivalent.

Selection

Specifying how selection operates within a population whose members vary in their overall fitness is a critical aspect of any population genetics model. The intensity of selection in Mendel is specified primarily through fertility, that is, the mean number of offspring per female. Normally, the size of the reproducing population is held constant. Selection eliminates surplus offspring beyond the number needed to match the target population size. Selection distinguishes those individuals that will mate and reproduce from those that will not. Generally speaking, the best phenotypes reproduce and the worst usually do not reproduce. However, in nature whether or not a given individual survives to reproduce does not depend exclusively on its genetic makeup. Random circumstances, including random variations in environment, usually play a significant role. Therefore, Mendel offers two options for adding environmental “noise” to genetic fitness prior to applying selection.

The first option is by means of a heritability parameter. Heritability is specified in the standard way—as the ratio of the genetic fitness variance to the total variance of fitness (=sum of the genetic fitness variance and the environmental variance). In

addition to this type of noise (which is present except when heritability equals 1.0), Mendel also allows a user to specify the standard deviation of normally-distributed fitness-independent noise (“non-scaling noise”). The square root of the sum of the variances of these two types of noise yields a total noise standard deviation. This is the scale factor for a normally-distributed random noise term that is added to the genotypic fitness of each individual to obtain its phenotypic fitness, which is then used in the selection process.

Mendel offers two primary selection methods, truncation selection and probability selection. Truncation selection eliminates those individuals in the new generation whose phenotypic fitness falls below an appropriate cutoff value. The cutoff is computed such that the prescribed population size, after selection, is exactly achieved. Mendel currently includes two versions of probability selection. Both versions apply a scaling factor to the phenotypic fitness and use this scaled phenotypic fitness as the criterion (probability) for reproductive success. One version, referred to as “classical” probability selection, limits the amplitude of the scaling factor such that the probability values never exceed one. With certain combinations of mean fitness and number of offspring/female, however, this can reduce the number of reproducing individuals below that required to maintain population size, even when fertility is high enough to maintain it. The other version, referred to here as “unrestricted” probability selection, does not impose this limitation on the scaling factor and therefore allows a sufficient number of offspring to reproduce to maintain population size. Under this method, offspring with scaled fitness exceeding one are automatically selected to reproduce. The second method is a consistent extension of the more traditional “classical” method to situations of low selection intensity (that is, few offspring/female). For moderate and high selection intensity the two methods are identical.

Applications of Mendel's parallel features

Mendel can utilize multiple processors to simulate three possible scenarios: (1) multiple replications of the same scenario, (2) a large homogenous population (to exploit the larger amount of distributed memory), or (3) multiple interacting heterogeneous or homogenous tribes. If one wishes simultaneously to replicate a given scenario many times, the task can be performed in parallel on multiple processors. Each replicate can be dealt with as if it was a fully isolated tribe (zero migration), with each replicate initialized with a different seed for the random number generator. Cases involving large population sizes can frequently exceed the memory capacity of

a single processor. Mendel is able to treat such cases by utilizing the larger amount of distributed memory available across multiple processors. This approach sub-divides the global population into tribes, and each tribe is assigned to a different processor (as below). Both genetic theory and numerical simulation show that as long as the rate of migration is at least 10%, the outcome is essentially identical to that of random mating within the global population.

Migration of an individual from one tribe to another is modeled by transferring that individual's genetic information from one Message Passing Interface (MPI) process to another. In general, each tribe is assigned to a separate processor (although with MPI it is possible to assign multiple tribes/processes to each processor). Communication of the genetic information of a migrating individual is performed asynchronously via standard non-blocking MPI `Isend` and `Irecv` calls. For each migrating individual, four types of information are communicated to the destination process: (1) the list of integers encoding the tracked deleterious mutations, (2) the list of integers encoding the tracked favorable mutations, (3) the list of fitnesses for each linkage block, and (4) the list of the total number of mutations in each linkage block. Before communication is performed, the four lists are gathered together from each of the randomly selected migrating individuals and packed into communication buffers. Data in the buffers are then transmitted to the appropriate destination.

Miscellaneous Features

Mendel provides the flexibility to treat bottleneck events beginning with a specified generation, persisting for a specified number of generations, and maintaining the reproducing population size at a specified small value during the bottleneck. Population size is immediately reduced to this small value at the beginning of the bottleneck, and the offspring number/female is maintained at 2 during the bottleneck interval (that is, no selection occurs during the bottleneck). After the bottleneck interval, the offspring number/female is restored to its original value, but selection is maintained at half its normal intensity until the population recovers to its original size.

Mendel also allows restart dumps to be written at a specified generational interval, from which a new run can be initiated, either retaining the original input parameters or specifying new ones. For independent replication of experiments, a user can run multiple instances of the same problem by specifying different random number generator seeds. Mendel can be easily accessed via its web user interface, shown in Figure 1, which enables a novice user simply to select default values but allows any user to gain access to

Mendel's many complex features. After entering the desired biological parameters and starting a run, the user can monitor that run as well as other previously submitted runs, viewing the output plots at the click of a button.

Verification and Validation

Most features on Mendel have been tested for correctness in implementation as well as for agreement with theoretical predictions (Sanford, Baumgardner, Gibson, Brewer, & Remine, 2007a). Simulation results compare very well with the theoretical expectations for situations where mathematical predictions are available. In cases where we could not make specific mathematical predictions, results still matched what general population genetic theory and logic would predict. Altering input parameters consistently resulted in expected effects. Although further validations are under way, current results indicate that Mendel produces reliable results for a wide range of parameter values.

Code Performance and Scaling

Most of the computational work in Mendel is associated with the segregation and recombination of mutations when a new offspring is formed. Mutations are transmitted from parent to offspring in linkage subunit chunks, one chunk from each parent's duplicate set of chromosomes. The amount of work per offspring is nearly proportional to the number of linkage subunits into which the haploid genome is divided. Timing tests on a 2.0 GHz AMD Opteron processor yield a scaling of about 100 nanoseconds per offspring per linkage subunit. For a reproducing population size of 1,000 individuals, three offspring per female, and 1,000 linkage subunits in the haploid genome, this scaling translates to a run time of 0.6 seconds per generation. This scaling assumes the choices of dynamic linkage and probability selection and a mean number of tracked mutations per individual of about 1,000. It also includes the time required for output diagnostics. Static linkage increases the run time slightly, while truncation selection decreases it slightly. The time requirement increases only modestly as the number of mutations increases beyond this reference value. Approximately an eighth of the total time is required by the selection process. Forming the offspring takes most of the rest of the time, with a few percent for the output diagnostics. For larger populations and/or large numbers of generations, Mendel can be run in a mode in which no tracking of individual mutations is performed but their fitness effects nonetheless still contribute fully to the linkage subunit composite value. In this mode Mendel runs about twice as fast as it does when a usual number of mutations are tracked. In this mode all mutations

are taken to be co-dominant, with a heterozygous expression of 50% of the homozygous value. This is an adequate approximation in many cases of interest. For most scenarios involving multiple tribes, parallel performance is close to single processor performance in terms of clock time per offspring per linkage subunit because in most cases only a few individuals are exchanged between processors and the amount of data communicated per individual is small.

Example Case

Mendel provides a powerful capability for simulating mutation and natural selection in a detailed and precise way and for exploring how these processes actually work. A companion paper (Sanford, Baumgardner, Gibson, Brewer, & Remine, 2008) treats this topic more extensively, so here we will provide only a single example case that highlights the main results. For this case we choose parameters that are suitable for a human population. Some of the most important parameters have to do with the mutation rate and the character of the mutation effect distribution. We choose 100 new mutations per offspring for the mutation rate (Araten, et al., 2005). To specify the mutation effect distribution, we choose the fraction of major deleterious mutations to be 0.0001, with a major deleterious mutation defined as causing a decrease in fitness to be at least 10%. The resulting fitness distribution is that of the dark blue curve in Fig. 3(a). The resulting average deleterious fitness effect for this distribution is -0.00009 .

In regard to favorable mutations, we choose that rate to be 0.0001 times the deleterious rate, or 0.01 favorable mutation per offspring. This, of course, is at least 100 times greater than what observation can support, but we want to include enough favorable mutations that their effects can be analyzed. We select a constant reproducing population size of 3,000 individuals. This is large enough that the results do not change in any major way compared with population sizes that are larger. With regard to selection intensity, we pick a reproduction rate of six offspring per female. Since only two offspring are required to maintain constant population size, this means that in every generation two-thirds of the offspring are selected away and do not reproduce. This represents relatively severe selection intensity. In regard to the level of environmental noise, we choose a heritability of 0.5. This represents a relatively mild role of environmental variation on phenotypic fitness, but this important factor is nonetheless included.

Figures 5–9 provide a visual description of the results of this case run for 500 generations. We can note from Figure 5 that the rate of accumulation of deleterious mutations per individual in the population is amazingly constant and very close to the product of

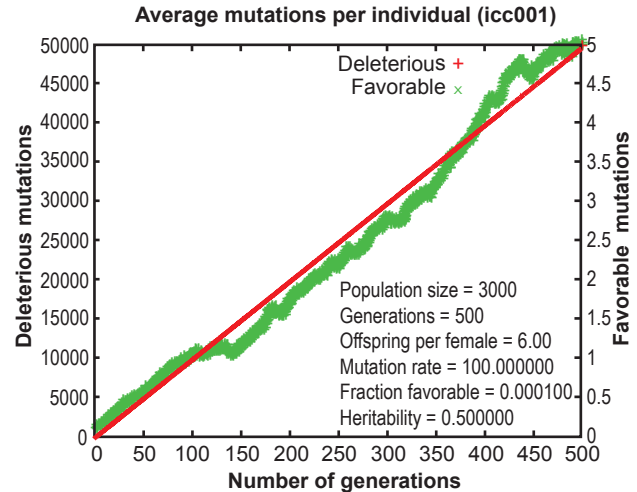


Figure 5. Number of deleterious and favorable mutations as a function of generation number. Note that both numbers are very nearly equal to the products of the mutation rate and the number of generations. Note also that the rate of increase in deleterious mutations is very nearly constant. These features are a consequence of the fact that the vast majority of mutations have effects too small to be detected individually by natural selection. Hence the vast majority of mutations, both deleterious and favorable, cannot be altered in their frequency by the natural selection process. Because of the small numbers of favorable mutations, statistical fluctuations in their accumulation rate are evident.

the number of new mutations per offspring (100) and the number of elapsed generations. This implies that the average number of mutations in the individuals that do not survive and reproduce must be very close to the average number of mutations in the individuals that do. After 500 generations each individual carries on average some 49,556 deleterious mutations, just 444 mutations short of the number of new mutations per offspring (100) times the number of generations (500). The number of favorable mutations also increases in at a relatively constant rate, but because the numbers of mutations are small the statistical variations are more prominent. Nevertheless, the average number of favorable mutations per individual is 4.935, very close to the value of 5.000 obtained from 0.01 mutations per generation for 500 generations. This behavior is a direct consequence of the fact that the vast majority of deleterious mutations and almost all the favorable mutations are unselectable, that is, their effects are individually too small for selection to detect. Selection, of course, acts only on cumulative phenotypic fitness, which includes the effects of variable environmental conditions on different members of the population.

Figure 6 shows the fitness and standard deviation of fitness within the population as a function of the elapsed number of generations. Note that the average population fitness drops by 65% over 500 generations. The inexorable decrease in fitness is a consequence of

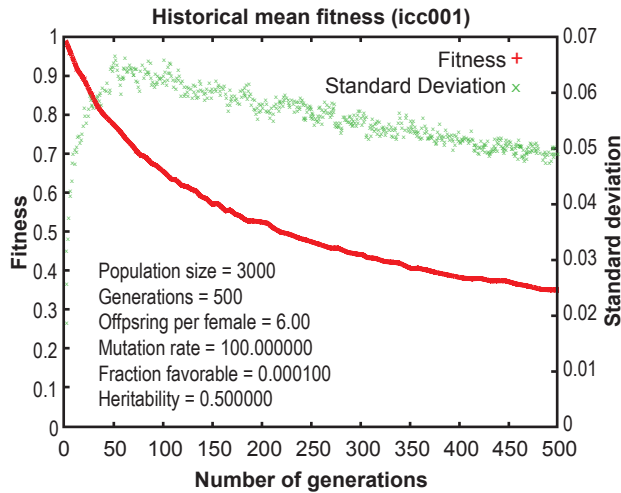


Figure 6. Mean population fitness and standard deviation in fitness as functions of the generation number. The mean deleterious fitness effect is -0.000091 . With a mutation rate of 100, without selection, there would be a decline in mean fitness of 0.91% per generation or 91% in 100 generations. By removing individuals with the most serious deleterious mutations, selection is able to reduce the fitness decline to only 65% in 500 generations. For this case the selection intensity is relatively high; four out of each six offspring do not survive to reproduce as a consequence of the selection process.

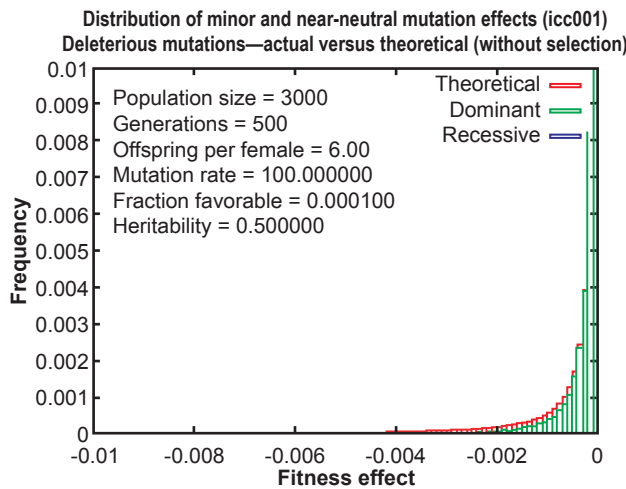


Figure 7. Distribution of mutation effect for deleterious mutations after 500 generations (green) compared with the distribution of effect in the absence of any selection (red). Note that for effects between 0.0 and -0.001 , the change in frequency due to selection is small to negligible. Note that only the bottommost 1% of the vertical scale is shown, which implies that the majority of all the mutations are in the rightmost bin. This plot shows clearly that the vast majority of mutations are not being influenced by the selection process, even though the selection intensity is relatively high. Only 1% of the horizontal scale, the rightmost portion, is shown; the maximum effect is -1.0 .

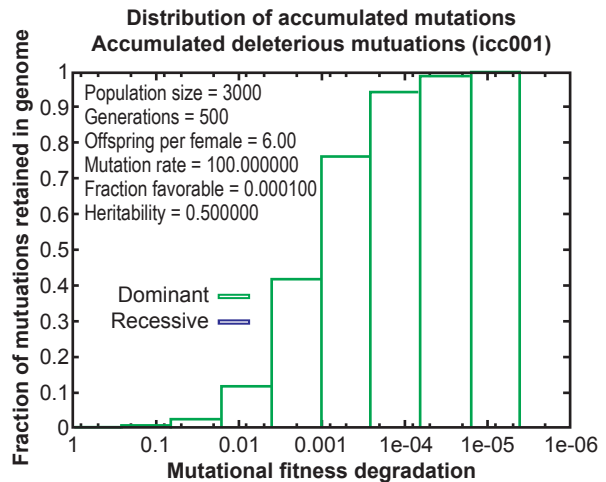


Figure 8. Fraction of deleterious mutations remaining after 500 generations as a function of mutation effect. The horizontal scale is logarithmic, but the portion of the distribution for effects smaller in magnitude than 10^{-6} is not shown. Without selection, each bin would contain the same equal number of deleterious mutations. Note that, as is evident also in Figure 7, mutations with effects smaller in magnitude than about 0.001 are hardly affected by selection.

the relentless accumulation of unselectable deleterious mutations generation after generation. The numerical simulation clearly confirms the reality of the process of genetic entropy (Sanford, 2005).

Figures 7 and 8 show the distribution of deleterious mutation effects for the mutations present in the population after 500 generations. Figure 7 displays the data on linear, but truncated, scales, both for fitness effect (horizontally) and frequency (vertically). The red bars denote the mutation effect distribution for the mutations entering the population before selection can act to modify the distribution. The green bars show the actual distribution of mutation effects after selection has operated for 500 generations. Note that the distribution is modified very little for effects smaller in magnitude than 0.001. Also note that the only 1% of the vertical and horizontal scales is displayed. Most mutations are in the rightmost bar. Most mutations with effects larger in magnitude than 0.01, which lie beyond the left edge of the plot, have been selected away.

Figure 8 attempts to highlight the mutation effect distribution for the high impact portion of the distribution by using a logarithmic scale for the mutation effect. That scale extends from a minimum absolute value of 10^{-6} on the right to a maximum absolute value of 1.0 on the left. The plot is constructed such that each bar in the histogram represents the fraction of mutations actually retained under the influence of selection relative to the mutations that would exist in the absence of selection. Thus deviations in height from a value of one reflect either

statistical noise or the action of selection. In this plot we can observe more clearly than in Figure 7 that selection is eliminating in an effective way the most deleterious mutations. Like Figure 7, it shows that the distribution of mutations with absolute effects smaller than about 0.001 is hardly influenced by selection. The numerical simulation shows that these low impact mutations are effectively unselectable. Though individually small in their effect, together they cause significant reduction in overall population fitness as shown in Figure 6.

Figure 9 displays the frequency of occurrence of individual mutations in the population, that is, the fraction of the population which carries the same mutation. This plot agrees with the theoretical predictions of the rate of drift of mutations in a population of a fixed size. We see clearly that on a time scale of 500 generations, drift is slow. When Mendel is run for tens of thousands of generations, fixation of mutations (that is, every individual in the population carries the mutation) is indeed observed to occur at the expected rate. Fixation of deleterious mutations, of course, represents irreparable damage to the genome. The slow rate of fixation of favorable mutations, as discovered by Haldane, means that positive evolution, if it were not ruled out because of other difficulties, is simply too slow to be of significance in higher organisms, an inference sometimes referred to as “Haldane’s dilemma.”

Discussion

The example case of the preceding section provides a glimpse of the more general conclusions one obtains as Mendel is applied to evaluate the

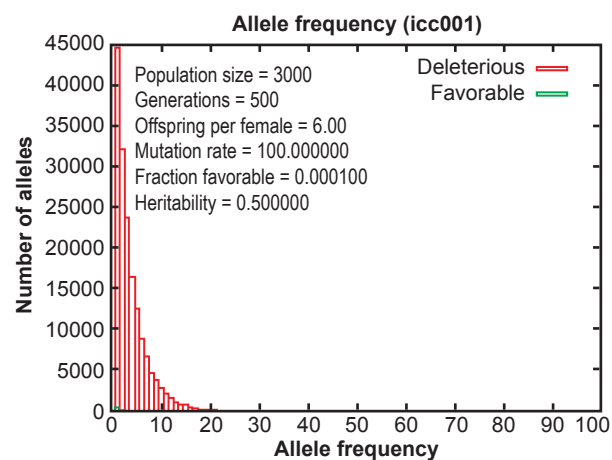


Figure 9. Frequency of occurrence of individual mutations in the population after 500 generations. The highest frequency is for mutations that occur only in one percent of the population. No individual mutation occurs in more than 22% of the population. This distribution shows the process of genetic drift in operation and that, even in a population as small as 3,000 reproducing individuals, drift is slow relative to a timescale of 500 generations.

efficacy of mutation and natural selection to produce genetic improvement, an efficacy that evolutionists so earnestly believe and affirm. In summary, if one uses parameters corresponding even in a crude sense to observed biological reality, then Mendel always shows genetic deterioration, not genetic improvement (Sanford et al., 2008).

There are several rather obvious reasons why this must be true. First is the well-documented and well-known fact that deleterious mutations are overwhelmingly more numerous than favorable ones. Next is the much less well-known, but nevertheless well-documented fact that the vast majority of mutations, both deleterious and favorable, are invisible to natural selection because of their small impact on overall fitness (Kimura, 1979). Their specific individual effects are simply too small for selection to detect and act upon, especially in the presence of normal environmental variation that produces significant non-genetic variations in individual fitness. Because of their small impact on fitness, most mutations accumulate unaffected by selection and, because the vast majority are deleterious, steady genetic deterioration inevitably results (Kondrashov, 1995, Sanford, 2005, Sanford, Baumgardner, Brewer, Gibson, & ReMine, 2007b).

Still another reason is that the nucleotides encoding genetic information are linked together in large blocks, blocks smaller than chromosomes but large nonetheless, which are passed intact from parent to offspring. Mutations, both deleterious and favorable, which fall within these blocks are inherited together and their effects all combine together. Hence, any favorable mutations that may occur are typically linked with dramatically more deleterious ones. The collective negative effects of the deleterious mutations almost always overwhelm the collective effects of any favorable mutations. Selection responds to the fitness effect of the entire block, not to that of individual mutations within the block. Individual mutations therefore become even less visible to selection than they would be otherwise because of the averaging effect of linkage (Sanford, 2005).

Population geneticists, although long aware of these realities, have allowed their unswerving commitment to evolution to cloud their reasoning and to prompt them to invoke a theoretical result that has no connection with biological reality. The theoretical result is the discovery by James Crow (Crow, 1997) that when, contrary to reality, mutations are assumed all to have *equal* effect on fitness, they then all become equally visible to the selection process. If one, as a further step away from reality, also omits all random aspects of the selection process and applies what is called truncation selection, then such selection does eventually halt the inexorable

accumulation of deleterious mutations. Mendel indeed confirms that this scenario works just as Crow found by paper and pencil analysis. But this model scenario has no connection with the real biological world. There is simply no debate, even from Crow, that deleterious mutations vary in their individual effects from lethal to nearly neutral. Neither is there any debate, even from Crow, that in the real world random processes do indeed interfere with selection's ability to sort and rank offspring in strict accordance with their genetic fitness. Mendel shows with no room for controversy that when mutations vary in their impact on fitness, the deleterious mutations that are invisible to selection accumulate without limit and fitness declines accordingly. This reality removes the final thread of hope for evolutionists that mutation and natural selection as general rule lead to genetic improvement and not genetic deterioration. The processes of mutation and natural selection simply do not deliver what evolutionists have been led, mainly by a handful of population geneticists, to believe and claim. Careful numerical modeling now confirms this conclusion with certainty.

Conclusions

Mendel's Accountant is a state-of-the-art forward-time population genetics model that tracks millions of individual mutations with their unique effects on fitness and unique location within the genome through large numbers of generations. It treats the process of natural selection in a precise way. It allows a user to choose values for a large number of parameters such as those specifying the mutation effect distribution, reproduction rate, population size, and variations in environmental conditions. Mendel shows consistently that when biologically realistic parameters are selected, genetic deterioration is an inevitable outcome of the processes of mutation and natural selection.

Mendel is a versatile and capable research tool that can be applied to problems in human genetics, plant and animal breeding, and management of endangered species. With its user-friendly graphical user interface and its ability to run on laptop computers it can also be fruitfully employed in teaching genetics and genetic principles, even at a high school level. Mendel is freely available to users and can be downloaded from <http://mendelsaccountant.info> or from <http://sourceforge.net/projects/mendelsaccount>.

References

- Araten, D.J., Golde, D.W., Zhang, R.H., Thaler, H.T., Gargiulo, L., Notaro, R., & Zuzzatto, L. (2005). A quantitative measurement of the human somatic mutation rate. *Cancer Research*, *65*, 8111–8117.
- Balloux, F. (2001). Computer Note—EASYPPOP (Version 1.7): A computer program for population genetics simulations. *Journal of Genetics*, *92*(3).
- Crosby, J.L. (1973). *Computer simulation in genetics*. New York, New York: John Wiley.
- Crow, J.F. (1997). The high spontaneous mutation rate: A health risk? *Proceedings of the National Academy of Sciences*, *94*, 8380–8386.
- Fraser, A., & Burnell, D. (1970). *Computer models in genetics*. New York, New York: McGraw-Hill.
- Goffeau, A., Barrell, B.G., Bussey, H., Davis, R.W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J.D., Jacq, C., Johnston, M., Louis, E.J., Mewes, H.W., Murakami, Y., Philippsen, P., Tettelin, H., & Oliver, S.G. (1996). Life with 6000 genes. *Science*, *274*, 546–552.
- Kimura, M. (1979). Model of effectively neutral mutations in which selective constraint is incorporated. *Proceedings of the National Academy of Sciences*, *76*, 3440–3444.
- Kondrashov, A.S. (1995). Contamination of the genome by very slightly deleterious mutations: why have we not died 100 times over? *Journal of Theoretical Biology*, *175*, 583–594.
- NIST/SEMATECH-Handbook of Statistical Methods. Retrieved June 25, 2007 from <http://www.itl.nist.gov/div89/handbook/eda/section3/eda3668.htm>.
- Sanford, J. (2005). *Genetic entropy and the mystery of the genome*. Lima, New York: Elim Publications.
- Sanford, J., Baumgardner, J., Gibson, P., Brewer, W., & ReMine, W. (2007a). Mendel's Accountant: A biologically realistic forward-time population genetics program. *Scalable Computing Practice and Experience*, *8*(2), 147–165.
- Sanford, J., Baumgardner, J., Brewer, W., Gibson, P., & ReMine, W. (2007b). Using computer simulation to understand mutation accumulation dynamics and genetic load. In Y. Shi, G.D. van Albada, J. Dongarra, & P.M.A. Sloot (Eds.), *International Conference on Computer Science 2007, Part II, Lecture Notes in Computational Science 4488* (pp. 386–392). Berlin and Heidelberg: Springer-Verlag.
- Sanford, J., Baumgardner, J., Gibson, P., Brewer, W., & ReMine, W. (2008). Using numerical simulation to test the validity of neo-Darwinian theory. In A.A. Snelling (Ed.), *Proceedings of the sixth international conference on creationism* (pp. 165–175). Pittsburgh, Pennsylvania: Creation Science Fellowship and Dallas, Texas: Institute for Creation Research.
- Santiago, E., & Cabellero, A. (2000). Application of reproductive technologies to the conservation of genetic resources. *Conservation Biology*, *14*, 1831–1836.