SCALABLE COMPUTING: PRACTICE AND EXPERIENCE
Volume 8, Number 2, pp. 147–165. http://www.scpe.org
ISSN 1895-1767


# MENDEL'S ACCOUNTANT: A BIOLOGICALLY REALISTIC FORWARD-TIME POPULATION GENETICS PROGRAM*

J. SANFORD†, J. BAUMGARDNER‡, W. BREWER§, P. GIBSON¶, AND W. REMINE‖

**Abstract.** Mendel's Accountant (hereafter referred to as "Mendel") is a user-friendly biologically realistic simulation program for investigating the processes of mutation and selection in sexually reproducing diploid populations. Mendel represents an advance over previous forward-time programs in that it incorporates several new features that enhance biological realism including: (a) variable mutation effect and (b) environmental variance that affects phenotype. In Mendel, as in nature, mutations have a continuous range of effect from lethal to beneficial, and may vary in expression from fully dominant to fully recessive. Mendel allows mutational effects to be combined in either a multiplicative or additive manner to determine overall genotypic fitness and provides the option of either truncation or probability selection. Environmental variance is specified via a heritability parameter and a non-scaling noise standard deviation. Mendel is computationally efficient, so many problems of interest can be run on ordinary personal computers. Parallelized using MPI, Mendel readily handles large population size and population substructure on cluster computers. We report a series of validation experiments which show consistently that Mendel results conform to theoretical predictions. Its graphical user interface is designed to make problem specification intuitive and simple, and it provides a variety of visual representations in the program output. The program is a versatile research tool and is useful also as an interactive teaching resource.

**Key words.** bottleneck, endangered species, environmental variance, genetic load, mutation, population genetics, selection, simulation

**1. Introduction.** Population geneticists have used mathematical modeling for over 75 years to understand better how mutation and selection affect population dynamics. 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 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, during meiosis there are about two crossovers per chromosome pair in most higher organisms [1]. 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.

**2. Background.** 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 [2]. Numerical strategies for population genetics modeling have been under discussion for several decades [3, 4], yet it is only recently that computing resources have become widely available to allow large realistic forward-time simulations.

*This work was supported in part by the FMS Foundation.
†Dept. Hort. Sci., NYSAES, Cornell University, Geneva, NY 14456(jcs21@cornell.edu).
‡Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, retired.
§Computational Engineering Dept., Mississippi State University, Mississippi State, MS 39759.
¶Dept. Plant, Soil, and Agric. Syst., Southern Illinois University, Carbondale, IL 62901
‖Science Dept., Northwestern College, Saint Paul, MN 55128

One type of genetic simulation, known as coalescent simulation, begins with a set of nucleotide sequences sampled in the present, and operates backwards in time to reconstruct a common ancestral sequence [5]. Some coalescent programs can handle recombination and natural selection to a limited extent; however, they become unwieldy if they incorporate natural selection for multiple loci. Coalescent-based methods have little relevance to modeling genetic change forward in time.

Forward-time simulations, although conceptually simpler, are computationally more difficult, and have been used primarily for teaching purposes [6]. Only recently, due to the rapidly decreasing cost of computing resources, has a widespread use of serious forward-time simulation programs become practical. The modeling of random mutations and the operation of natural selection under complex mating/recombination schemes are distinctive advantages of forward-time simulations.

Forward-time simulations provide the opportunity to study evolutionary processes in a genetically explicit and realistic manner. Such programs are increasingly being applied to a large variety of questions in evolutionary biology, conservation biology, and human genetics [7]. However, the availability of a biologically realistic yet efficient and easy to use software package has been lacking. Guillaume and Rougemont [7] point out that most studies in these areas rely on homemade code which is rarely published. This has tended to force researchers to build their models from scratch, models which may or may not have been properly validated.

In addition to the complexities of such things as recombination and recessivity, forward-time simulations of large population sizes and high mutation rates are very demanding on memory and computer time and require software to be specially designed for speed-efficiency.

Below is a summary of the previously available forward-time simulations:

1. PopG and Simul8 [8] are for teaching basic concepts limited to one or two loci.
2. FreGene [9] simulates sequence-like data in large genomic regions under the influence of crossovers, gene conversions, and hot spots. It investigates the sequence patterns produced by these processes, and does not appear to address selection and population fitness.
3. EASYPOP [2] is specifically designed to study neutral evolution without selection.
4. FPG [10] is a simulation that provides many of the same features as Mendel. Both aim at biological realism and can be run on ordinary PCs. However, Mendel allows the user to choose between equal mutation effects (with the magnitude of effect specified separately for beneficial and deleterious mutations) or a natural, continuous distribution of mutation effects. By contrast, FPG's modeling of mutation is restricted to one selection coefficient for all deleterious mutations and applies an equal and opposite effect for all beneficial mutations. In addition, Mendel provides explicitly for user-specified environmental noise, whereas in FPG environmental noise can only be obliquely approximated by reducing the selection coefficient. Moreover, Mendel provides for complete independence of all linkage blocks (useful for comparing with theoretical calculations), whereas FPG does not. Mendel's interactive specification of inputs through its graphical user interface is much more user-friendly than the command-line parameter entry in FPG. Several input parameters are more intuitive to a user in Mendel than they are in FPG. Whereas population size in Mendel is a simple input parameter, the population size in FPG can be changed from its default value of 500 diploid individuals only by modifying and recompiling the program. Some advantages of FPG over Mendel are that the interval of diagnostic outputs is currently fixed in Mendel but can be user specified in FPG, linkage disequilibrium can be analyzed in FPG but not in Mendel, and FPG can output pseudo-DNA sequences for polymorphisms for subsequent molecular-level analysis (such as with the same author's SITES program).
5. SimuPOP [6] is a simulation environment that operates under Python, a widely known object-oriented scripting language.
6. Nemo [7] is a simulation framework, provided through a library of C++ routines.

SimuPOP and Nemo are sophisticated software packages that strive for a high degree of flexibility, but that flexibility imposes a steep learning curve for the user. These general-purpose software packages do not expose their deeper implementation details in a manner that users can easily access or optimize apart from modifying the software. The increased complexity arising from the flexibility of these programs also tends to make these packages computationally less efficient.

Mendel represents an advance in forward-time simulations, compared to those just described, by incorporating several improvements:

1. Mendel adds the ability to model mutations as having a continuous, natural distribution of mutation effects. ("Mutation effects" are often equated with "selection coefficients" in population genetics literature. The effect of a mutation on phenotypic fitness is equal to the selection coefficient of that mutation only with strict probability selection and no environmental variance, but not otherwise. Since Mendel also offers truncation selection

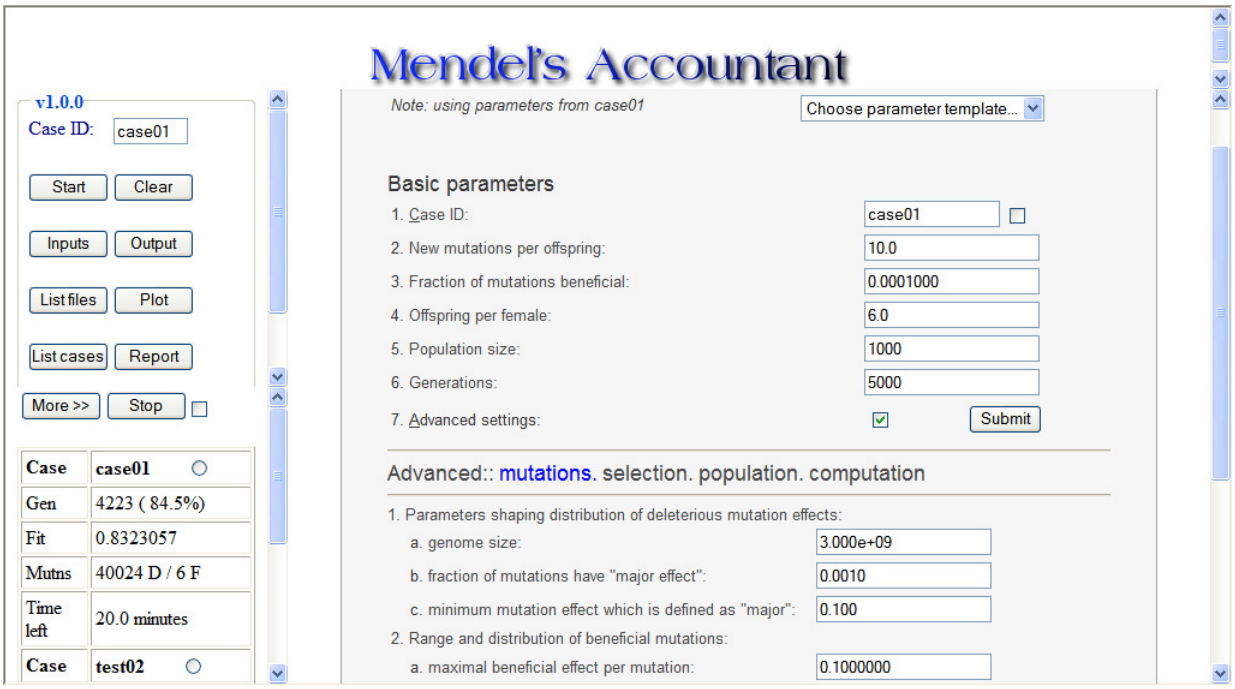


FIG. 2.1. Web user interface of Mendel's Accountant showing a portion of the input window.

and provides for environmental variance, we distinguish between the terms "mutation effect" and "selection coefficient" and use the term "mutation effect" in the context of Mendel.)

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). This is unlike the commonly used k-allele or stepwise models, which impose highly restrictive limits on mutational variation.
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 2.1), 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 (Figures 2.2-2.4).

Like many current simulations, Mendel also provides a variety of options for mating, bottleneck events, and population 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: (1) multiple interacting heterogeneous tribes (2) multiple replications of a single case, or (3) a very large population comprised of sub-populations but with sufficient migration to maintain a high degree of genetic homogeneity.

**3. Numerical Approach.** A basic overview of the software implementation of Mendel is shown in Algorithm 1, where $NG$ is the number of generations and $NP$ is the population size. 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.

**3.1. 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.

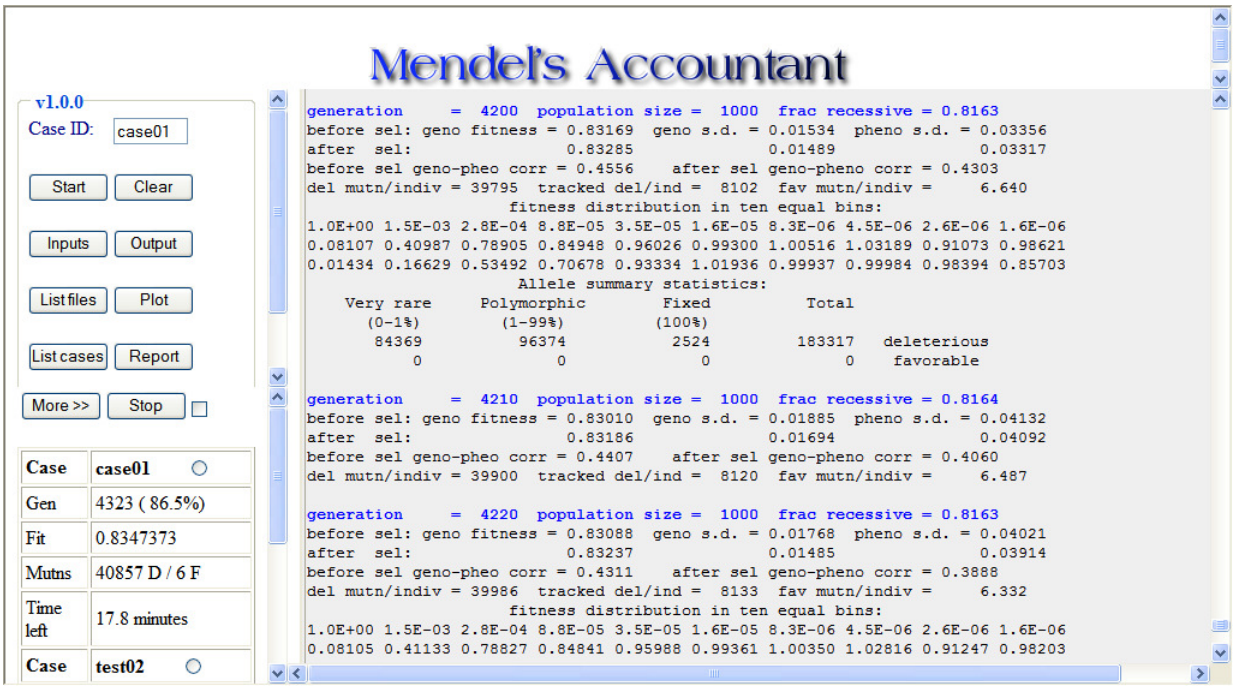


FIG. 2.2. Web user interface of Mendel's Accountant showing a portion of the output window.

**Algorithm 1** PSEUDOCODE OF NUMERICAL APPROACH.

**for** $i = 1$ to $NG$ **do**
  migration {randomly select individuals and send their genetic information to the appropriate neighboring tribes}
  **for** $j = 1$ to $NP/2$ **do**
    mating {randomly mate half of the population with members from other half}
    offspring {offspring receives half its genetic makeup from each of its two parents; add new random mutations to offspring genome}
  **end for**
  selection {impose selection based on phenotypic fitness to reduce the population size}
**end for**

We recognized that to track millions of individual mutations in a sizeable 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 indicates 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 in two haplotype sets, one inherited from each of that individual's parents. Each haplotype is divided into a user-specified number of linkage subunits. In meiosis, one member of each linkage subunit pair is randomly selected, with all its associated mutations, and is inherited by the gamete. If linkage

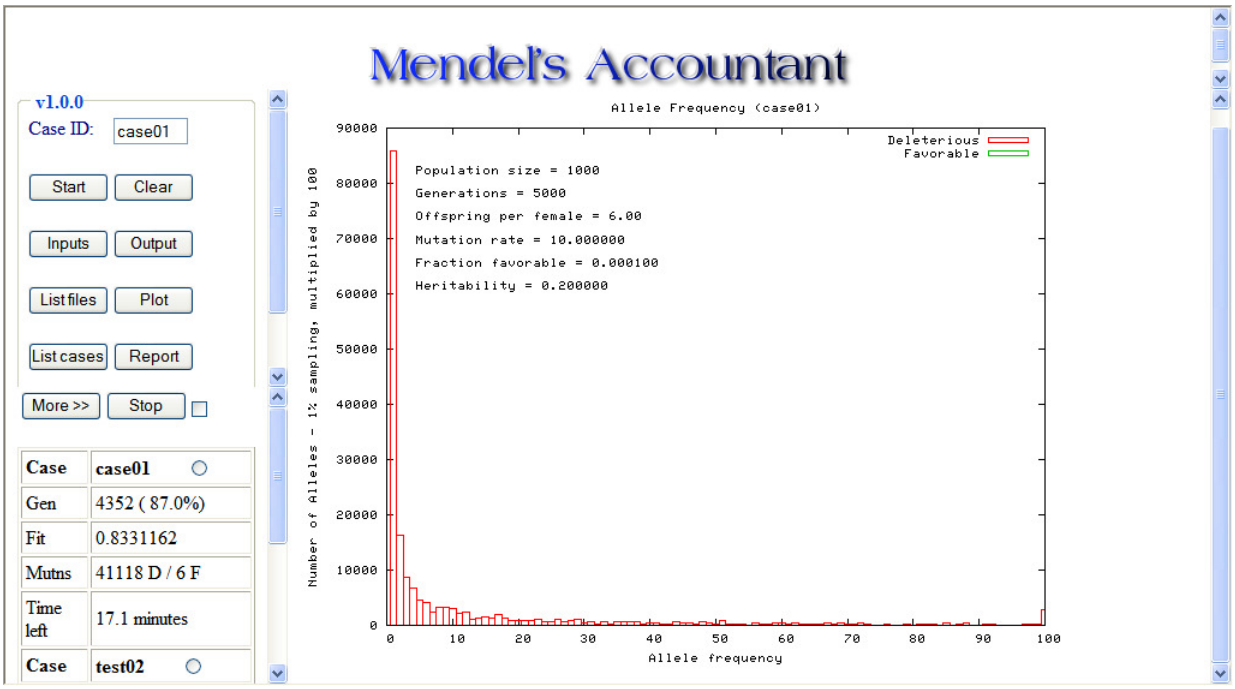


FIG. 2.3. Web user interface of Mendel's Accountant showing plot of allele frequencies.

is specified to be static, all linkage subunits are inherited independently of one another. However, if the user specifies dynamic linkage, many subunits that reside on the same 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 positions are random, they almost always occur at different points along each chromosome from one generation to the next. It is during gamete formation that new mutations are introduced. After mating and reproduction, the memory used to store the mutation information for the reproducing generation is overwritten with the mutation information of 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.

**3.2. 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 how mutations are assumed to alter fitness. In particular, selecting a distribution of mutational effect that matches biological reality is 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 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 [11] 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 [12] that is a generalization of the more usual exponential function. Our function, expressed by eq. (3.1), 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.

$$d(x) = d_{sf} \exp(-ax^{\gamma}), 0 \leq x \leq 1. \tag{3.1}$$

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

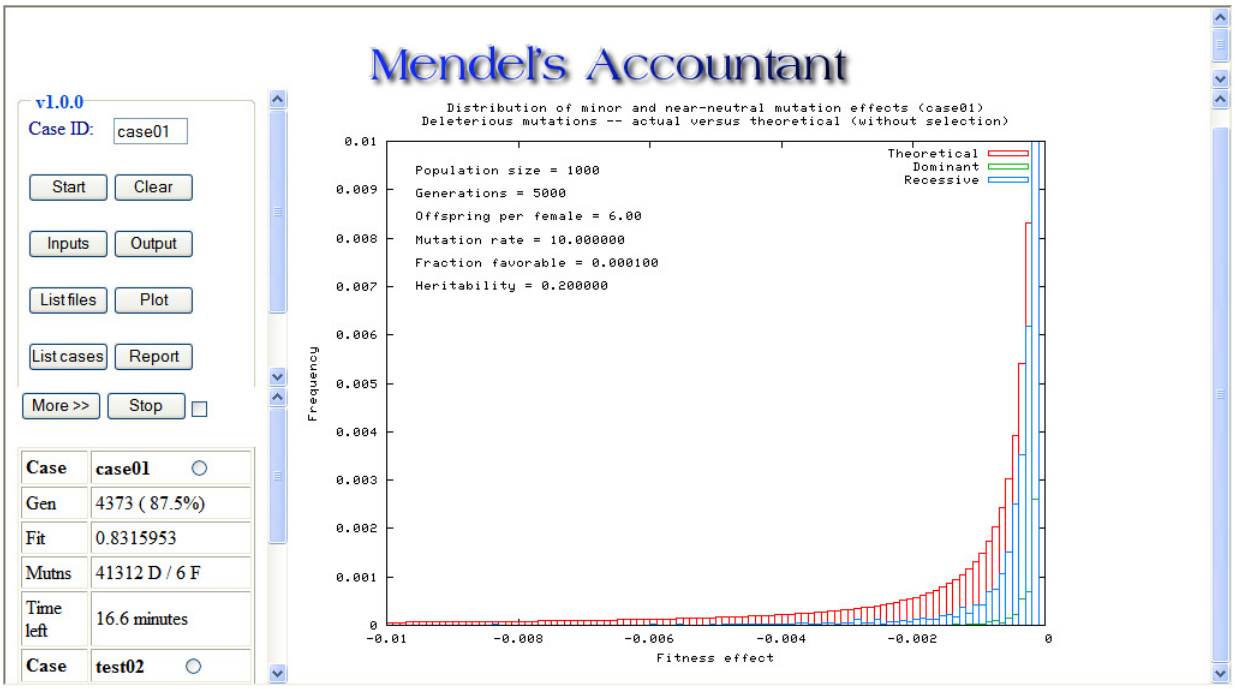


FIG. 2.4. Web user interface of Mendel's Accountant showing distribution of mutations with respect to fitness effect. Red bars represent mutation distribution in the absence of selection. Blue and green bars represent actual accumulated recessive and dominant mutations, respectively, in the presence of selection. The two bars representing mutation classes with effects nearest zero extend beyond the vertical scale of the plot.

are meaningful 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 (e.g., 0.01 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 $\gamma$, both positive real numbers, determine the shape of the fitness effect distribution. We apply the same values of $a$ and $\gamma$ 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 $d(x)$ 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 is essentially 0. For organisms with smaller genomes such as yeast, which has a value for $G$ on the order of $10^7$, the minimum absolute effect is larger. This is consistent with the expectation that each nucleotide in a smaller genome on average plays a greater relative role in the organism's fitness.

The second parameter $\gamma$ can be viewed as controlling the fraction of mutations that have a large absolute fitness effect. Instead of specifying $\gamma$ directly, we select two quantities that are more intuitive and together define $\gamma$. The first is $\theta$, 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 code the value of $\gamma$ is computed that satisfies these requirements. We reiterate that Mendel uses the same value for $\gamma$, and thus the same values for $\theta$ and $q$, for both favorable and deleterious mutations. Figure 3.1 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 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.

**3.3. 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

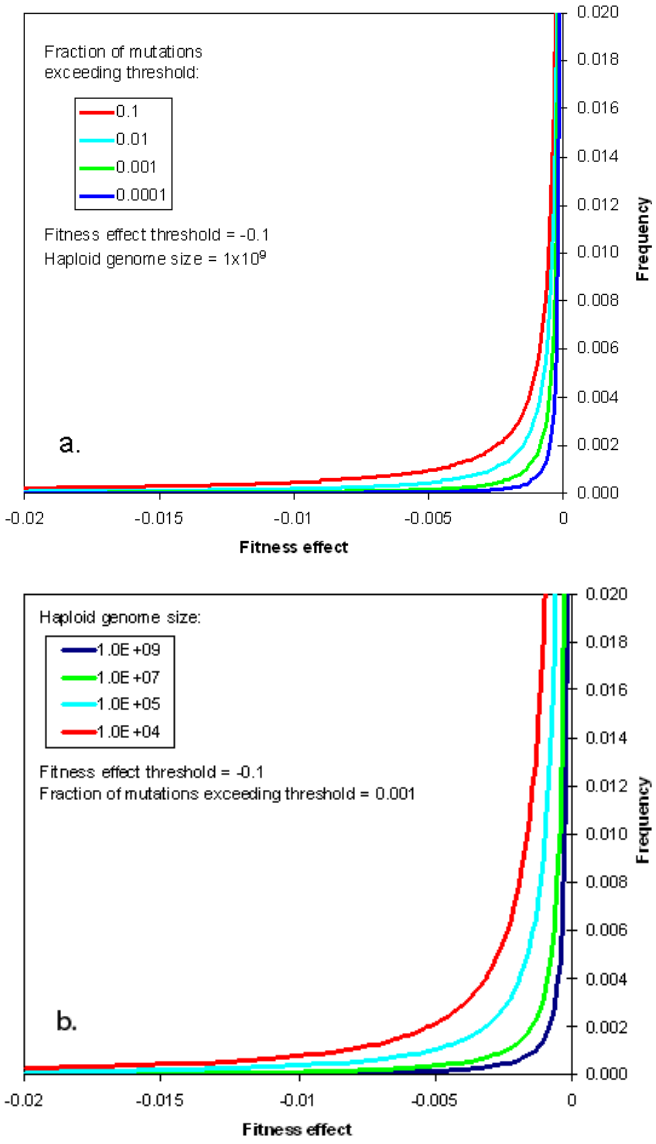


FIG. 3.1. (a.) Response of the fitness-effect distribution function to changes in the fraction of "high impact" mutations (0.0001 to 0.1). (b.) Response of the fitness-effect distribution function to changes in the specified haploid genome size (number of nucleotides =$1 \times 10^4$ to $1 \times 10^9$). 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 value.

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 modulus $\mu$ given by $2^{31} - 1 = 2,147,483,647$ (which is the largest value a four-byte integer can assume) divided by $\lambda$, the number of linkage subunits. For example, if $\lambda$ is 2000, then we choose $\mu = 1,073,741$. If we let the symbol $\sigma$ be either 1 or $-1$ to denote whether the mutation is dominant or recessive and let $m$ be the integer 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 real value of the random number between zero and one that specifies the mutation's fitness effect. We apply the modulo function with modulus $\mu$ to the absolute value of $m$ to recover $x$. We divide $m$ by $\mu$ and use the `int` function to recover $l$.

The mutation indices just described are stored in ascending numerical order for each haplotype in each individual. This allows us to be able to readily test whether a given mutation is homozygous, that is, whether or not the mutation

occurs on both copies of the individual's chromosome set. When a new mutation is introduced, the existing mutation 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 haplotypes 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 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, the appropriate additional effect is applied as a multiplier of the particular mutation effect. Since a mutation that is exactly co-dominant has a heterozygous effect of 0.5 of the homozygous effect, the added effect from homozygosity will be $> 0.5$ for a recessive mutant, and $\leq 0.5$ for a dominant mutant.

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 an array to contain the cumulative heterozygous fitness effects from all the mutations associated with each linkage subunit for each of the two haplotypes 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 the added homozygous effect 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.

**3.4. 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 3.2 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 case of the two-way stepping stone model, if the user specifies one individual, 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 identically. Similarly, for the case of three tribes, the two-way stepping stone and island migration models are equivalent.

**3.5. 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 con-

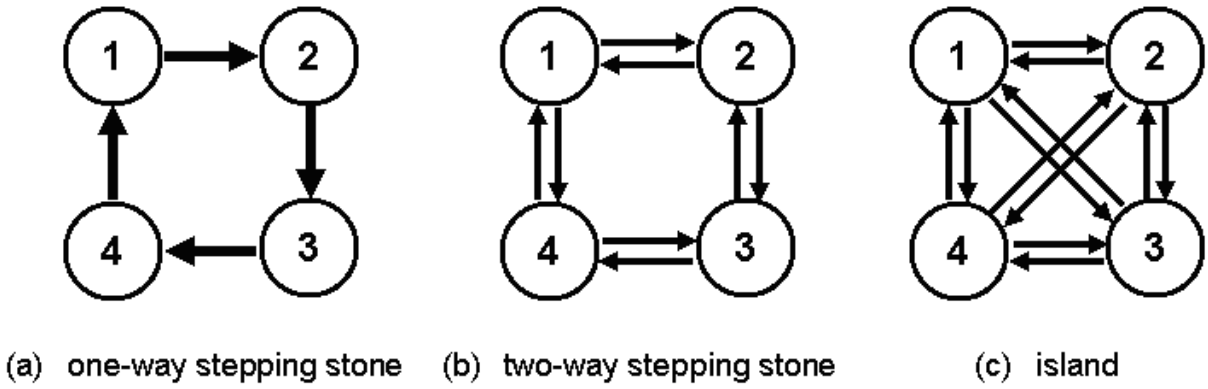


FIG. 3.2. Migration models available in Mendel.

stant. 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), 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 (i. e., few offspring/female). For moderate and high selection intensity the two methods are identical.

**3.6. Parallel Implementation.** 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.

**3.6.1. Multiple replications of the same scenario.** 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.

**3.6.2. Large homogeneous populations.** Cases involving large population sizes can frequently exceed the memory capacity of a single processor. Mendel is able 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.

**3.6.3. Multiple interacting tribes.** Migration of a 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. Algorithm 2 represents the subroutine that is called every M generations, where M is specified by the user, NP is the total number of tribes, NRT is the number of receiving tribes, and NI is the number of individuals sent to each receiving tribe.

**Algorithm 2** PSEUDOCODE FOR TRIBAL MIGRATION.

```
   select {randomly select all individuals to migrate from the local tribe and find the required buffer size based on
   maximum mutation count}
 2: for m = 1 to NRT do
      compute destination process {destination process (island model) = mod(myid + m, NP)}
 4:   for i = 1 to NI do
        pack buffers
 6:     call MPI Isend
        call MPI Irecv
 8:     call MPI Waitany
        call MPI Waitall
10:     unpack buffers
      end for
12: end for
```

**3.7. 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 (i. e., 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 2.1, 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.

**4. Validation.** Extensive validation is under way for each input parameter and for many of their combinations. Evidence that the program correctly responds to the most important input parameters is presented below.

**4.1. Validating mutation creation and the resulting fitness values.** Mutation numbers per individual (beneficial and detrimental) are generated as a random Poisson function. The number of mutations generated by the simulation matches the number of mutations specified in the input. The comparison was evaluated for a wide range of detrimental mutation rates ($u = 0.001$ to 1000) in simulations of 1000 generations each. Variances of mutation number among individuals very closely correspond to the mean, as is expected with a Poisson distribution. In early generations, discrepancies from the input values were well within the limits of random error. As accumulation progressed, these discrepancies were consistently less than 1% of the total expected numbers.

When mutational effects were specified as equal, the fitness means and variances corresponded exactly to the number of mutations (assuming all loci co-dominant, i. e., heterozygous expression = 0.5). With unequal mutation effects, the distribution of effects created by Mendel for these test runs corresponded very well with those calculated from the input

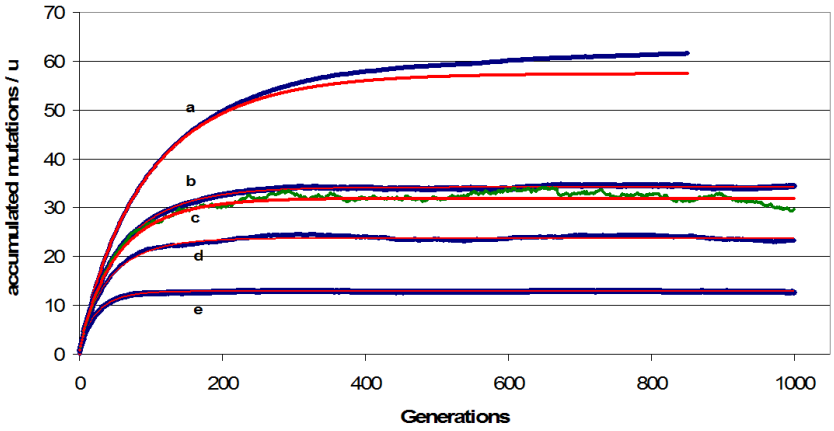


FIG. 4.1. *Comparison showing excellent agreement between simulations (dark blue or green) and theoretical calculations (red) for numbers of accumulated mutations with truncation selection. The number of new mutations per generation (u) and number of offspring/female (o) vary. Number of offspring/female may also be expressed as selection intensity (i). Input parameters for the five cases were: (a) u=100, o=20, i = 0.9; (b) u = 25, o=6, i = 0.67; (c) u =1, o=2.2, i = 0.1; (d) u =10, o = 4, i = 0.5; and (e) u = 25, o=20, i =0.9. Note that the graph presents the actual number of accumulated mutations divided by u in order to accommodate a wide range of input values for mutation rate and selection intensity. Other input values were constant, including equal effects of mutations, 2000 fixed linkage blocks, population size = 1000, and heterozygous effect = 0.5.*

parameters. Evaluations included the default input set (genome size: $3 \times 10^9$, high-impact effect threshold: 0.1, high-impact mutation frequency: 0.001) and input sets which varied each of the three parameters that define the mutation-effect distribution. Simulation outputs from the three options for combining effects (additive, multiplicative, and combinations of the two) agreed precisely for the joint effect of equal-effect mutations. The agreement was also excellent for unequal mutation effects. Individual fitness values output by the simulation matched the proportions of recessive and dominant mutations and their respective fitness expression in the heterozygote state. The latter was evaluated by comparing heterozygote effect = 0 with heterozygote effects between 0 and 1, while keeping all other input parameters the same.

**4.2. Validating selection.** Truncation selection is the most powerful form of selection, and is used extensively by plant and animal breeders. In this type of selection, reproduction by an individual depends exclusively on whether its phenotypic fitness is greater than the truncation value. However, truncation selection probably never occurs in nature. Rather, some form of probability selection occurs, in which individuals with higher phenotypic fitness have a higher probability of reproduction. The exact relationship between phenotypic fitness and reproductive success in nature is not known, but certainly it varies, depending on the organism and its environmental context. Mendel currently offers two versions of probability selection as described in section 3.5, but except for circumstances of low fertility (i. e., few offspring/female), the two versions are identical.

**4.2.1. Validating selection with equal mutational effects.** Selection effects were extensively validated using: 1) equal mutational effects (all mutations have an effect of identical magnitude, specified by the user); 2) complete co-dominance (heterozygous expression $=$ 0.5); 3) truncation selection; and 4) no environmental noise. The resulting simulated means and variances for mutation numbers and fitness corresponded almost perfectly with theoretical values. As illustrated in Figure 4.1, this was true: i) through hundreds or thousands of generations; ii) for mutation rates from 0.01 to 1000; and iii) for selection intensities from 10% to 90% (2.22 to 20 offspring/female). As expected, lower mutation rates showed greater proportional fluctuations over generations than did higher mutation rates (case c in Figure 4.1). In a few cases, the accumulation of mutations exceeded somewhat the predictions from theoretical calculations (case a in Figure 4.1). These cases involved either substantial numbers of accumulated mutations per linkage block ($> 3$) or very large numbers of accumulated mutations. The theoretical calculation assumed infinite population size (no sampling error, no inbreeding), no linkage, and no fixation of alleles. Thus, greater mutation accumulation in the simulation compared with predicted numbers might plausibly result from several factors: 1) co-segregation of alleles within linkage blocks; 2) effects of Muller's ratchet [13]; 3) accelerated allele fixation due to small population size (especially $<= 1000$); 4) inbreeding associated with recurrent selection in small populations. Further study of the effect of these factors is under way.

Theoretical calculations for each generation were based on the standardized selection differential ($k$) corresponding to a given selection intensity, and on before-truncation genetic variance predicted from the previous generation. The

expected average number of accumulated mutations per individual in generation $g$ after selection, $M_g$ is expressed by

$$M_g = M_{g-1} + u - k\sigma_g \tag{4.1}$$

where $u$ is the number of new mutations added and $\sigma_g$ is the post-selection genetic standard deviation of generation $g$. This post-selection genetic standard deviation in turn is given by

$$\sigma_g^2 = \sigma_{g-1}^2/2 + M_{g-1}/2 + u. \tag{4.2}$$

The divisor, 2, of $\sigma_{g-1}^2$ results from a gamete receiving an average of half as many mutations as its parent (thus dividing $\sigma_{g-1}^2$ by 4), then being combined with another unrelated gamete (thus doubling $\frac{1}{4}\sigma_{g-1}^2$). This divisor of 2 can also properly be thought of as resulting from the reduction in variance from averaging of the mutation numbers in the two parents. For convenience, we assume the mutation numbers of the parents to be normally distributed, since the skewing produced by selection results in only small departures from normality. $M_{g-1}$ is also divided by 2 to reflect the binomial variance of $\frac{1}{4}N_{g-1}$ for gamete mutation number, given a specified parental mutation number, which is then doubled because of the summing of two gametes in the zygote. This binomial variance of $\frac{1}{4}N_{g-1}$ equals $p(1-p)N_{g-1}$, where $N_{g-1}$ is the number of mutations in the parent and $p = 0.5$ is the probability of transmission of a specific mutant allele from parent to offspring. The two variances on the right in equation (4.2) are essentially uncorrelated due to random mating and random recombination of gametic mutation numbers from mating pairs. The addition of $u$ reflects the Poisson variance of the new mutations entering the population each generation.

Initial evaluation of effects of the level of dominance on selection gave the predicted results, confirming our expectation from both theory and the programming approach that there should be excellent congruence between simulated and theoretical results with regard to level of dominance.

In probability selection, the likelihood of reproductive success of an individual is proportional to its fitness, but the correlation is imperfect, so reproduction is dependent in part on chance. Therefore, the mean and variance of a the reproducing individuals are more variable than they are in truncation selection. Also, the standardized selection differential is lower than in truncation selection. In truncation selection, only the relative fitness of individuals influences reproductive success. However, in probability selection, the actual fitness value itself, resulting from the absolute magnitude of each mutational effect, influences the probability of successful reproduction of a specific individual. These factors complicate the prediction equations and so they are not presented here. The simulation results for each type of probability selection corresponded very well with the theoretical expectations (Figure 4.2). Here, unrestricted and strict probability selection are presented as one, since they are identical except when there are few offspring/female (i. e., mild selection intensity).

**4.2.2. Validating selection with unequal mutational effects.** Mendel's default mode creates a natural and continuous distribution of unequal mutation effects. This results in an essentially exponential distribution where most mutational effects are extremely small. Such small mutation effects will be almost uniformly distributed in all individuals. Thus individual fitness, both before and after selection, will vary largely based upon the magnitude of the effects of a few large-effect and medium-large effect mutations. For example, with a population of 1000, a mutation rate of 10, and 0.1% high-impact mutations ($|d| > 0.1$), roughly 10 individuals will have a deleterious mutation ranging in absolute effect from 0.1 to 1. With truncation selection, these 10 individuals will almost always be selected away (unless fewer than 1% of the individuals are being eliminated). In contrast, how often individuals with these high-impact mutations are retained with probability selection depends on numerous variables including the specific type of probability selection and the variability in fitness among individuals. In addition, mutations with somewhat smaller effects than the high-impact category will very rarely be retained with truncation selection but will often be retained under probability selection, at least for a number of generations.

With unequal mutation effects, it is difficult to produce precise theoretical predictions of means and variances because significant mutations are continuously and randomly occurring that are not consistently being eliminated. In the absence of precise predictions, the validity of Mendel was supported by the fact that different numbers of mutations/generation (u) resulted in the expected pattern of fitness decline, as did comparison of different selection intensities (truncation selection with $u = 20$ is shown in Figure 4.3). The pattern of elimination of mutations followed very closely the selection intensity (Figure 4.4), and the magnitude of mutation effect for which selection was no longer effective corresponded very closely for each selection intensity with that predicted by Kimura ($s = 1/(2N)$), where $s$ is the selection coefficient, and $N$ is the effective population size [14]. The selection coefficient associated with a specific magnitude of mutation effect $d$ was calculated as $s = kd/\sigma_p$ (data not shown), where $k$ is the standardized selection differential for a given selection intensity, and $\sigma_p$ is the phenotypic standard deviation of fitness across individuals [15]. In addition, we verified that the effect of selection from one generation to the next was approximately what we expected for both truncation selection and

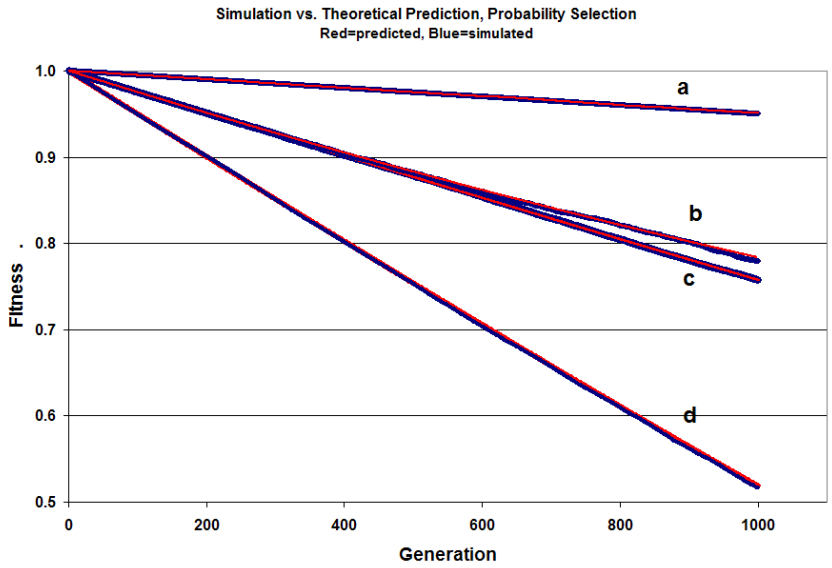


FIG. 4.2. *Simulation results (dark blue) vs. theoretical calculations (red) for fitness over time with probability selection for differing numbers of new mutations(u) added to each zygote each generation. Mutations were of equal effect, of magnitude d. Input combinations shown are: (a) u=10, d=0.0001; (b) u=5, d=0.0001; (c) u=5, d=0.0005, and (d) u=1, d=0.001. Other key input parameters were: fixed linkage blocks=2000, population size = 1000, offspring/female = 6, heterozygous effect = 0.5.*

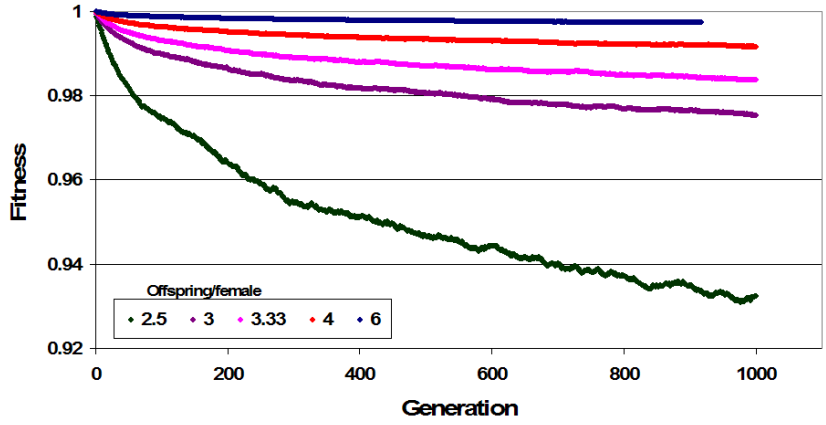


FIG. 4.3. *Fitness decline becomes less severe as numbers of offspring increase. Offspring/female of 2.5, 3, 3.33, 4, and 6 correspond to selective elimination fractions of 0.2, 0.3, 0.4, 0.5, and 0.67, respectively. The simulations used unequal mutational effect (default distribution, see text) combined with truncation selection. Other key input parameters were: u=20, population size=1000, fixed linkage blocks=2000, heterozygous effect = 0.5.*

probability selection. This conclusion was based on several input values of $u$ and offspring/female where simulated mean fitness in specific generations with and without selection in the final generation were compared with the expected effect of selection, given the array of individual fitness values without selection. All reported runs used the default mutation distribution (genome size of $3 \times 10^9$, high-impact mutation effect $|d| >= 0.1$, and high-impact mutation fraction of 0.001), which yields a mean degradation/mutation of 0.000506 and a median degradation/mutation = $2.7 \times 10^{-8}$.

**4.3. Validating the effect of noise on selection.** Mendel allows the user to specify two types of environmental effects that cause phenotypic fitness to be more variable than genotypic fitness. The first type of environmental variability is expressed via a heritability parameter, defined as the ratio of genetic variance to total variance. In the absence of environmental noise, heritability is 1.0. In nature, the heritability of fitness may be 10% or less [16]. For each generation, Mendel calculates the genetic variance and adds a random noise factor scaled to yield the heritability value the user has specified. In addition to this, the program can add fixed magnitude noise (called non-scaling noise). Non-scaling

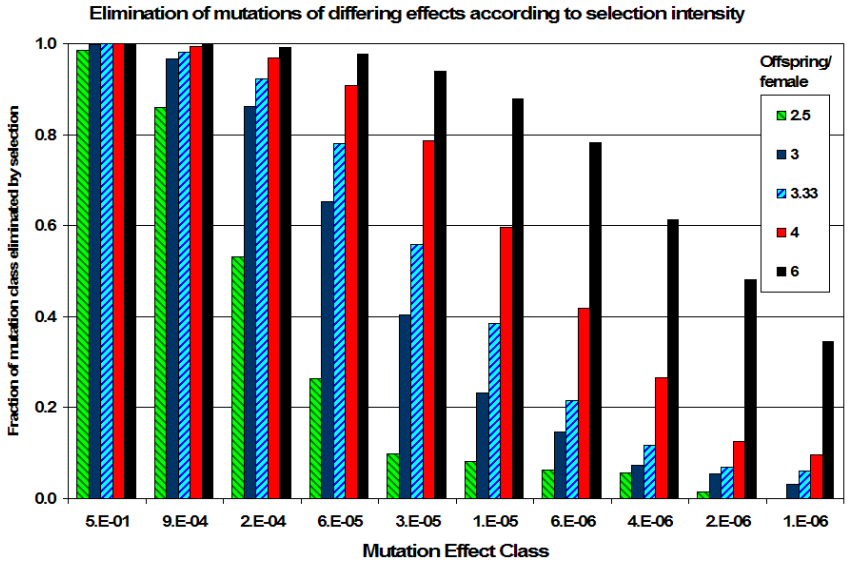


FIG. 4.4. *Diminishing effectiveness of selection with diminishing magnitude of mutation effect (x-axis). Adjacent bars represent the fraction of mutations of each class eliminated by different selection intensities (offspring/female). Mutations were grouped into 10 classes of magnitude of effect (x-axis, large effects on left), with the median effect shown as the x-axis label for each class. Numbers of offspring/female of 2.5, 3, 4 and 6 correspond to selective elimination fractions of 0.2, 0.3, 0.4, 0.5, and 0.67 of the population, respectively. Note the absence of the bar (zero elimination, complete retention) for mutations of small effect ($10^{-6}$) combined with weak selection (2.5 offspring/female). Other key input parameters were: u=20, population size=1000, fixed linkage blocks=2000, heterozygous effect = 0.5)*

noise is added after the noise specified by the heritability, and makes actual "realized" heritability less than the input heritability. Mutation accumulation under truncation selection corresponded well with expected values when noise was added to mutations of equal effects. As expected, selection efficiency was greatly reduced with heritability values less than 0.5. Noticeable reductions in selection efficiency were also seen even with heritability values of 1 when non-scaling noise was added. A range of input values of heritability and/or non-scaling noise reduced selection efficiency in the manner expected.

**4.4. Validating the effect of linkage on selection.** The default number of linkage blocks per haploid genome is purposely set lower than the actual number in humans (approximately 100,000 [17]) because the computer memory and run time requirements increase greatly with increases in the number of linkage blocks. The default value for fixed linkage is 1000 independent blocks. For dynamic linkage the default is 23 equal-length chromosomes, with two random crossovers per chromosome per generation, and 1000 total blocks (actually 989 blocks are used as the default, since 989 is the nearest exact multiple of 23). These default parameters are adequate up to the point where the number of mutations exceeds three per haploid block, in which case the effectiveness of selection is reduced, presumably because of Muller's ratchet. Consistent with this explanation is the fact that with large numbers of accumulated mutations, reductions in block number increased mutation accumulation rates with both fixed and dynamic linkage. Also as expected, in many parameter combinations with a specific number of linkage blocks, mutation accumulation was slightly to moderately greater with dynamic linkage than with fixed linkage. With dynamic linkage in nature and in Mendel, segments about 1/3 of a chromosome in length are transmitted intact to the progeny, allowing less opportunity for selection to act freely on each of the blocks of the transmitted unit. With dynamic linkage, neighboring blocks are co-transmitted for several generations until recombination occurs between them.

**4.5. Summary of validation efforts.** Simulation results compared very well with the theoretical expectations whenever we were able to make mathematical predictions. 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.

**5. 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 haplotype 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 1000 individuals, three offspring per female, and 1000 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 1000. 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 per individual is small.

**6. Application Examples.** Understanding the accumulation of mutations is of great importance to society [18]. In man, mutation accumulation is at the heart of many important health problems. Cancer is largely the result of mutation accumulation within our somatic cells, and accumulated mutations in our germline cells are clearly implicated in our predisposition to various cancers [19]. The aging process itself is clearly associated with accumulation of mutations in our somatic cells. This appears to be especially true of the aging effects of mitochondrial mutations, particularly in the heart and brain [20]. The high rate of birth defects (3-4% in the US) is largely due to accumulated mutations. There is considerable concern about the growing "genetic load" within modern populations [18, 21, 22, 23]. Mendel can help us understand more about human mutation accumulation, which might help us to understand the importance of possible mutation mitigation measures.

In addition to mutations within the human genome, we are also affected by the accumulation of mutations in the pathogens that affect us. Mutations within a pathogen's genome can change the pathogen's antigenic character, resulting in resistance to immune responses and antibiotics. In some cases certain strains of pathogens may undergo genetic degeneration and error catastrophe, changing the dynamic balance between strains. This aspect of epidemiology might be better understood using Mendel in the near future, once haploid clonal reproduction is added as an option.

It is also now realized that minimizing mutation accumulation is a critical factor for preserving endangered species and avoiding mutational meltdown. Likewise, in agriculture all our efforts to collect and preserve germplasm for future plant and animal breeding might be nullified unless drift and mutation accumulation are not kept in check. The breeding value of otherwise desirable genes and linkage blocks could be largely negated by these effects. Mendel is clearly a tool that can provide greater understanding in all these areas. Below are three specific applications for which Mendel can inform us about real world genetic situations.

**6.1. Decreased Exposure to Mutagens.** We can generate a realistic simulation of what would happen to the human population if mutation rates could be decreased ten-fold. To do this we can start from a previous Mendel case which has reached a near-equilibrium fitness after 2000 generations. Suppose this population has been experiencing a mutation rate of 10 new mutations per individual per generation. We can restart this old case where it left off—but now with only one new mutation per individual per generation. What we see in Figure 6.1a is an immediate and dramatic reduction in the rate of accumulation of new mutations. In addition, fitness begins to increase markedly (Figure 6.1b). This indicates that the deleterious mutations that had previously accumulated are now being removed from the population faster than new mutations are being added—so "genetic load" is actually being reduced. This indicates that if the human mutation rate could be reduced significantly, in time it would have a major impact on human fitness and health.

**6.2. A Population Bottleneck.** Many previously endangered species have recovered from genetic bottlenecks (which result from a temporary reduction in population size). The American Bison is an example of this. Many currently endangered species, such as the panda, are still in the bottleneck phase, and hopefully will recover and expand. Using Mendel, we can generate a realistic simulation of a genetic bottleneck. We can again re-start from an equilibrated population. The population size is reduced from 1000 to 100, for 500 generations, and then the original population size is restored.

What we see when this prolonged bottleneck begins is that the rate of mutation accumulation does not change significantly (Figure 6.2a). However, Figure 6.2b makes it clear that if the bottleneck had continued, extinction would have

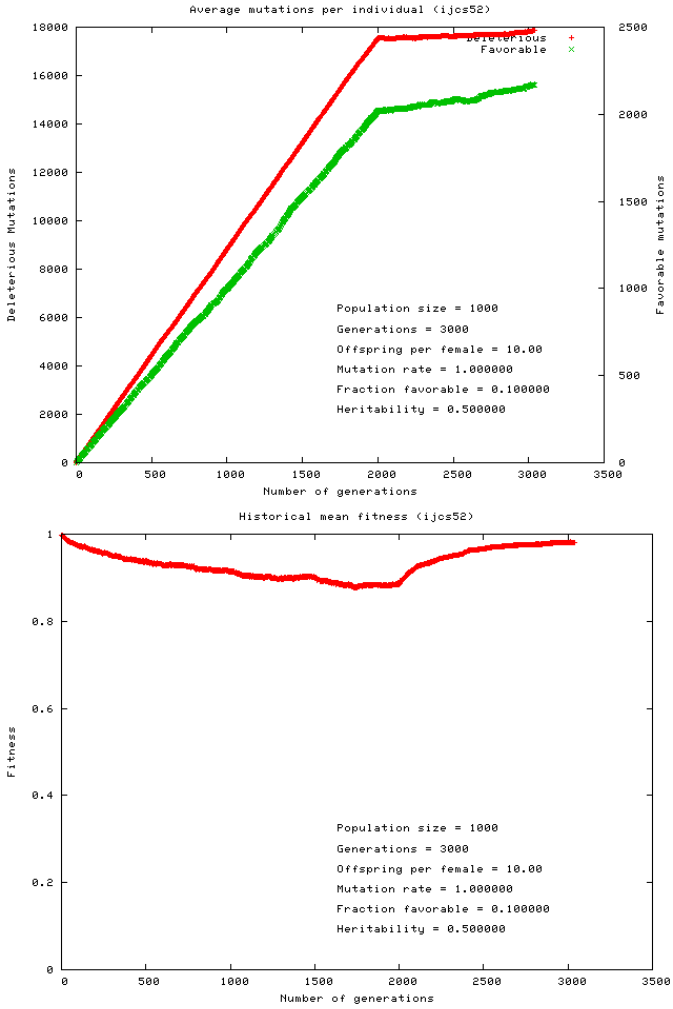


FIG. 6.1. Effect of decreased exposure to mutagens following generation 2000. Top: (a) deleterious mutation count per individual. Bottom: (b) historical mean fitness. This case employed probability selection and the multiplicative model for combining gene effects, with 80% of the mutations recessive. Parameters prescribing the mutation effect distribution were: genome size = 3 billion, high impact mutation fraction = 0.001, high impact threshold = 0.1, maximum favorable fitness effect = 0.01.

occurred. This is because higher-impact mutations that would otherwise have been selected away were then accumulating due to stronger genetic drift. When the bottleneck ended, the population had a very strong re-bound in fitness. Restored to its larger population size, selection was able to override drift, and the higher-impact deleterious mutations that had been accumulating began to be eliminated. However, the recovery in fitness was only partial because of the deleterious mutations that reached fixation during the bottleneck.

This experiment reveals three interesting things. First, bottlenecks cause rapid genetic degeneration and will lead to extinction if not halted. Secondly, when a bottleneck ends, there is a strong rebound in fitness as effective selection is restored. Thirdly, long bottlenecks cause irreversible genomic damage. Fortunately, other experiments (not shown) clearly indicate that bottlenecks lasting only a few generations do not cause permanent genomic damage. This indicates that it is imperative that species bottlenecks be ended as soon as possible. Mendel can help predict the minimum population size required to allow maximum population recovery.

**6.3. Population Substructure.** Totally isolated small populations, such as populations on small ocean islands, are potentially subject an irreversible bottleneck phenomenon. However, if there is a modest amount of migration, the problem of local inbreeding and drift can be largely relieved. Mendel allows us to model large "global populations" which are subdivided into many smaller sub-populations. To allow for bigger runs, Mendel has been parallelized so that each sub-population can be run on its own computer processor, allowing $N$ sub-populations to be run on $N$ parallel processors

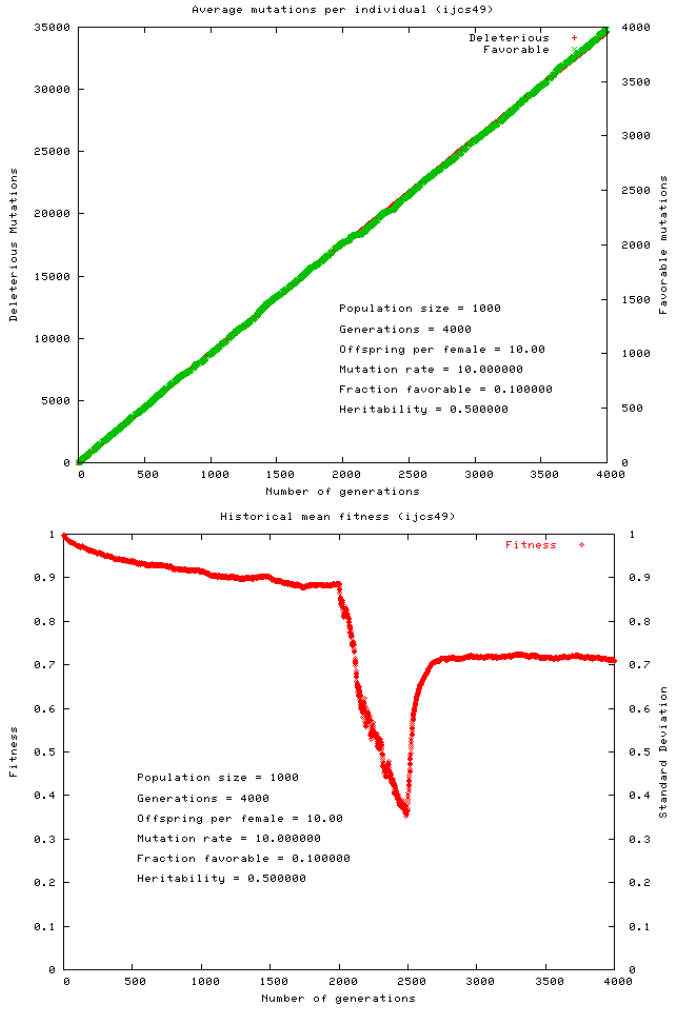


FIG. 6.2. Example of a population bottleneck starting at generation 2000 and lasting 500 generations. Top: (a) deleterious mutation count per individual. Bottom: (b) historical mean fitness. This case employed probability selection and the multiplicative model for combining gene effects, with 80% of the mutations recessive. Parameters prescribing the mutation effect distribution were: genome size = 3 billion, high impact mutation fraction = 0.001, high impact threshold = 0.1, maximum favorable fitness effect = 0.01.

(where $N$ is the number of available processors). Mendel can be used to determine empirically how much migration/cross-breeding is needed to prevent the island inbreeding effect. When a population of 1000 is divided into 10 sub-populations of 100 individuals each, and where there is zero migration, we see rapid degeneration for each sub-population, and the first sub-population goes extinct in just 591 generations (Figure 6.3a). However, if there is just one inter-tribal migration per tribe every ten generations, the island inbreeding effect is largely relieved, resulting in population stabilization (Figure 6.3b).

**7. Conclusions.** Mendel's Accountant is a biologically realistic numerical simulation that models forward-time genetic change within a population, as affected by mutation and selection. It is highly flexible, computationally efficient, allows large scale simulations, and is user-friendly. Mendel is freely available to users and can be downloaded from `http://mendelsaccountant.info` or from `http://sourceforge.net/projects/mendelsaccount`

REFERENCES

[1] E. SANTIAGO AND A. CABELLERO (2000) Application of reproductive technologies to the conservation of genetic resources, Conservation Biology, 14, pp. 1831–1836.
[2] F. BALLOUX (2001) Computer Note—EASYPOP (Version 1.7): A Computer Program for Population Genetics Simulations, J. Genetics, 92(3).
[3] A. FRASER AND D. BURNELL (1970) Computer Models in Genetics, McGraw-Hill, New York, NY.

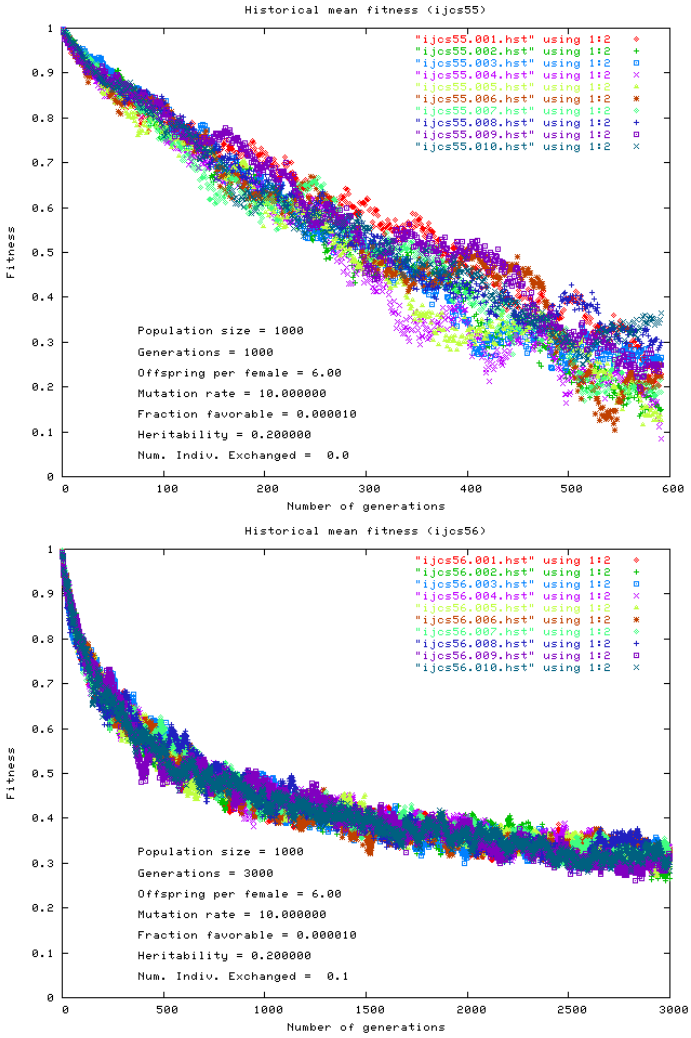


FIG. 6.3. Population substructure example showing mean tribal fitness as a function of generation for ten inbreeding tribes, each represented by a different color on the graph. Top: (a) without migration. Bottom: (b) migration rate = 1 individual per tribe every 10 generations. Note that the total number of generations in (a) is 600 and in (b) it is 3000. These cases employed probability selection and the multiplicative model for combining gene effects, with 80% of the mutations recessive. Parameters prescribing the mutation effect distribution were: genome size = 3 billion, high impact mutation fraction = 0.001, high impact threshold = 0.1, maximum favorable fitness effect = 0.01.

[4] J. L. CROSBY (1973) Computer Simulation in Genetics, John Wiley, New York, NY.
[5] J. KINGMAN (1982) The coalescent, Stochastic Proc. Appl., 13, pp. 235–248.
[6] B. PENG AND M. KIMMEL (2005) simuPOP: a forward-time population genetics simulation environment, Bioinformatics, 21(18), pp. 3686–3687.
[7] F. GUILLAUME AND J. ROUGEMONT (2006) Nemo: an evolutionary and population genetics programming framework, Bioinformatics, 22(20), pp. 2556–2557.
[8] J. FELSENSTEIN (2005) http://evolution.gs.washington.edu/popgen/popg.html (accessed 12 January 2007).
[9] C. HOGGART, T.G. CLARK, R. LAMPARIELLO, M. DE IORIO, J. WHITTAKER, B. BALDING (2005) FREGENE: software for simulating large genomic regions. Technical Report, Department of Epidemiology and Public Health, Imperial College, http://www.ebi.ac.uk/projects/BARGEN/download/FREGEN/fregeneweb.html (accessed 12 January 2007).
[10] J. HEY (2004) A computer program for forward population genetic simulation, http://lifesci.rutgers.edu/~heylab/HeylabSoftware.htm#FPG (accessed 12 January 2007).
[11] M. KIMURA (1979) Model of effectively neutral mutations in which selective constraint is incorporated, PNAS, 76, pp. 3440–3444.
[12] NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/eda/section3/eda3668.htm (accessed 15 February 2007).
[13] H. J. MULLER (1964) The relation of recombination to mutational advance, Mutation Research, 1, pp. 2–9.
[14] M. KIMURA AND J. CROW (1978) Effect of overall phenotypic selection on genetic change at individual loci, PNAS, 75(12), pp. 6168–6171.
[15] R. MILKMAN (1979) Selection differentials and selection coefficients, Genetics, 88, pp. 391–403.

[16] M. KIMURA (1983) Neutral Theory of Molecular Evolution, Cambridge University Press, New York, NY, pp. 30–31.
[17] S. A. TISHKOFF AND B. C. VERRELLI (2003) Patterns of human genetic diversity: implications for human evolutionary history and disease, Ann. Rev. Genomics and Human Genetics, 4, pp. 293–340.
[18] J. F. CROW (1997) The high spontaneous mutation rate: a health risk? PNAS, 94, pp. 8380–8386.
[19] D. J. ARATEN, D. W. GOLDE, R. H. ZHANG, H. T. THALER, L. GARGIULO, R. NOTARO, AND L. ZUZZATTO (2005) A quantitative measurement of the human somatic mutation rate, Cancer Research, 65, pp. 8111–8117.
[20] G. C. KUJOTH, P. C. BRADSHAW, S. HAROON, AND T.A. PROLLA (2007) The role of mitochondrial DNA mutations in mammalian aging, PLoS Genetics, 3, pp. 0161–0173.
[21] A. S. KONDRASHOV (1995) Contamination of the genome by very slightly deleterious mutations: why have we not died 100 times over?, J. Theor. Biol., 175, pp. 583–594.
[22] L. LOEWE (2006) Quantifying the genomic decay paradox due to Muller's ratchet in human mitochondrial DNA, Genetical Res., 87, pp. 133–159.
[23] J. SANFORD, J. BAUMGARDNER, W. BREWER, P. GIBSON, AND W. REMINE (2007) Using computer simulation to understand mutation accumulation dynamics and genetic load, in Y. Shi et al. (eds.), ICCS 2007, Part II, LNCS 4488, Springer-Verlag, Berlin, Heidelberg, pp. 386–392.

*Edited by:* Dazhang Gu
*Received:* March 5, 2007
*Accepted:* April 15, 2007