OPEN ACCESS

BIO-complexity

Research Article

Overabundant mutations help potentiate evolution: The effect of biologically realistic mutation rates on computer models of evolution

Winston Ewert*

Biologic Institute, Redmond, Washington, USA

Abstract

Various existing computer models of evolution attempt to demonstrate the efficacy of Darwinian evolution by solving simple problems. These typically use per-nucleotide (or nearest analogue) mutation rates orders of magnitude higher than biological rates. This paper compares models using typical rates for genetic algorithms with the same models using a realistic mutation rate. It finds that the models with the realistic mutation rates lose the ability to solve the simple problems. This is shown to be the result of the difficulty of evolving mutations that only provide a benefit in combination with other mutations.

Cite as: Ewert W (2015) Overabundant mutations help potentiate evolution: The effect of biologically realistic mutation rates on computer models of evolution. BIO-Complexity 2015 (1):1–11. doi:10.5048/BIO-C.2015.1.

Editor: Ola Hössjer

Received: March 21, 2015; **Accepted:** October 19, 2015; **Published:** November 16, 2015

Notes: A *Critique* of this paper, when available, will be assigned doi:10.5048/BIO-C.2015.1.c.

*email: evoinfo@winstonewert.com

INTRODUCTION

In order to explain the complexity of biological life we see today, Darwinian evolution must have solved a diverse set of biological engineering problems. For example, evolution must have produced wings that fly, hearts that pump blood, flagella that provide mobility, etc. In order for Darwinian evolution to be a working model of origins, it must actually have the ability to produce these biological structures. The slow speed of evolution renders testing this claim by direct observation infeasible. We simply cannot wait long enough to see if wings, hearts, or flagella can evolve.

In order to help address this gap, various computer models of evolution have been developed [1–3]. Each model proposes a problem, analogous to the various engineering problems that biological evolution had to solve. These problems range from matching a target sequence [3] to producing a functional antenna [4]. These models simulate a process of Darwinian evolution attempting to solve these problems. Naturally, in order for this to be feasible the problem must be much simpler than the problems that biological evolution has had to solve. This way these problems can be solved given limited time and computational resources.

Many computer models have been created that consistently and rapidly evolve solutions to their proposed problems. That is, when run, these computer models produce a good, often optimal, solution to the problem in a relatively small number of generations with high probability. It is argued that evolution's success on these simple problems provides evidence for its ability to solve the more difficult biologically realistic problems in natural settings.

Previous work has critiqued these models as not being suitable analogs to a non-teleological evolutionary process [5–10]. However, even accepting the basic models as given, the parameters employed by these programs diverge from biologically realistic settings. Choosing biologically realistic parameters has been observed to disrupt the evolutionary process typically observed in these models [11–13].

There is a related class of programs, population genetics simulations [14]. Like the models discussed above, they simulate the Darwinian process. However, instead of attempting to model evolution's ability to solve problems, they are focused on the accurate simulation of population dynamics. One example, Mendel's Accountant, has focused on using accurate simulations of evolution to probe the limits of Darwinian adaptation [12,15–18]. Unlike the models under discussion in this paper, population genetics simulations like Mendel's accountant often use biologically realistic parameters.

Mutation rates in these computer simulations are not commensurate with mutation rates in biology. For example, Avida uses a substitution mutation rate of 0.0025 per instruction (an instruction is the basic unit of information in Avida, like the base is the basic unit of information in DNA) [1]. Comparably, Ev's programs experience a single mutation in a small genome [2], giving a substitution rate of approximately 0.0038 per nucleotide. In contrast, viruses have mutation rates ranging from 10^{-4} to 10^{-8} per base pair per generation [19]. A survey by Drake et al. of higher organisms shows mutation rates ranging from 10^{-7} to 10^{-11} [20]. The human mutation rate is estimated to be 1.1×10^{-8} [21]. While biological mutation rates vary by several orders of magnitude, the most rapidly mutating viruses undergo mutations at a rate an order of magnitude less than these computer models.

Lenski et al. argued that "various organisms from nature have genomic mutation rates higher or lower than [Avida's mutation rate]" [1]. However, this computation is made relative to the entire genome rather than individual bases or analogous equivalent (see above). The rate of mutation per Avida program, 0.225, is within the range of rates of mutations per whole genome of various organisms [20]. However, an Avida program is much closer in size to a gene than a genome. The appropriate comparison is to the mutation rate of a single gene, not a whole genome.

Schneider pointed out that HIV only mutates at a rate tenfold lower than his model, Ev. He extrapolates from his model's results to HIV: "Because the mutation rate of HIV is only 10 times slower, it could evolve a 4 bit site in 100 generations [2]". He appears to assume that given a ten times lower mutation rate, evolution will take ten times as long. He does not elaborate on a justification for this inference.

In evaluating whether or not Darwinian processes can account for the biological complexity found in nature, it would be better to avoid attempting to extrapolate from the performance of high-mutation rate evolutionary models to low-mutation rate biological systems; instead, it would be more useful to study low-mutation rate evolutionary models. This paper adapts the three models, AVIDA, Ev, and Acids, to use a biologically realistic mutation rate. The models are compared in three reproductive scenarios: 1) a scenario based on a typical genetic algorithm, with a high mutation rate; 2) a scenario of human evolution using a realistic mutation rate; and 3) a scenario where 20,000 human genes are evolving in parallel. Human evolution is used as an example, as it is a scenario that has been argued to be beyond the ability of Darwinian processes [22].

The results of this research demonstrate that these types of evolutionary simulations generally fail to perform adequately when given realistic mutation rates. Instead they consistently only solve the easiest parts of the problems. Even if we allow 20,000 genes to evolve in parallel, none of them manage to solve these simple problems. The computer models developed thus far do not solve simple problems when using a biologically realistic mutation rate.

The models fail due to the necessity and difficulty of obtaining potentiating mutations. That is, in each of the models, it is impossible to evolve the solution one beneficial mutation at a time. Some of the mutations necessary to solve the problem will be neutral or deleterious when they first arise. These are called potentiating mutations because they are not helpful by themselves, but introduce the potential for other mutations to be beneficial. Note that the two mutations do not have to arrive at the same time, but both must be present before natural selection can favor either one.

Developing adaptations that require such potentiating mutations has been argued to be very improbable [23–26]. In other papers it has been argued it is not as improbable [27], but see a critique of these arguments [28]. The current work adds to the discussion by allowing the requirement for potentiated mutations to follow from the evolutionary model instead of simply postulating the necessity of potentiating adaptations.

This paper uses evolutionary models developed to support Darwinian evolution to demonstrate that when using realistic mutation rates, these models no longer function effectively. This undermines the argument that they support Darwinian evolution and raises a serious challenge to claims of the effectiveness of Darwinian evolution in solving real-world biological challenges.

ANALYSIS

This paper considers three different evolutionary models. Each one is adapted from a previously published evolutionary model that has been used to defend the thesis that Darwinian evolution is sufficient to produce virtually all observed biological complexity. Each model defines a particular problem to be solved. It also defines the structure and meaning of the genes within the model.

Background

In these models, as in biology, each allele is a sequence of bases: A, C, G, and T. Two of the models were not encoded as bases, but in this paper new encoding has been devised for them using bases. The length of the allele sequence is fixed and determined by the model. Each model defines a procedure to compute a score from the allele. The different models use completely different techniques to produce this score. For example, the Target Acids model measures performance as the number of amino acids that match a particular target sequence. In contrast, the Avida model interprets the allele as a computer program that is run to test which tasks the program successfully completes. These scores are used to emulate natural selection–those alleles that are assigned higher scores are deemed better, and favored by selection relative to those alleles with lower scores.

It should be noted that the scoring used in each of these models is arbitrary, and has little to do with biology. In fact, the models depend more on computer science constructs than on biological ones. For the purposes of this paper, we are accepting the models and their scoring methodology as given, and focusing on the effect of using realistic mutation rates.

In order to evaluate the effect of mutation rates, this paper considers three different reproductive scenarios. Each reproductive scenario defines the starting generation, a process for producing the next generation, and the number of generations that will be produced.

These reproductive scenarios can theoretically be used to test the effects of varying mutation rates, selection strengths, population sizes, numbers of generations, etc. on the outcomes of the different models. We examine only the effects of varying mutations rates here.

Description of the reproductive scenarios

Standard Genetic Algorithm. For the purpose of comparison, the first reproductive scenario is a Monte Carlo genetic algorithm intended to approximate the typical usage of a genetic algorithm. It begins with a population of 2000 members with a fixed number of bases. The number of bases is determined by the model. For each of the nucleotides in each member of the population, a base is randomly selected with uniform probability from the four possible bases.

For each model, the genetic algorithm follows a three-step process. First, for each member of the population, a score is calculated. The score is the measure of fitness for a given sequence of bases. The model, not the reproductive scenario, defines the actual scoring process. The individual models (described later) all potentially produce different scores for the same sequence of bases. The score measures the degree to which the given sequence of bases corresponds to a correct solution to that model's problem. In each case the score is a natural number. Only one model is being used during a run of the simulation.

The second step in the genetic algorithm is roulette wheel selection [29]. Each member of the population is assigned a fitness value. In the case of this scenario, the fitness of a member m_i is $f_i = 1.1^{s_i}$ where s_i is the score of member m_i. This is a multiplicative fitness function, which means that each one-point increase in the score results in a 10% increase in the fitness of the allele, giving it an average of 10% more offspring. Note that this means that a two-point increase will produce a 21% increase in the average number of offspring, not a 20% increase.

A new population is constructed by choosing 2000 members from the original population. The probability of selecting a particular member is proportional to the fitness, and is:

$$p_i = \frac{f_i}{\sum_j f_j},$$

or simply the fitnesses of all members of the population normalized so that they sum to one. After constructing the new population, the old population is discarded and the process continues with the new population.

The third step is mutation with nucleotide substitution. This reproductive scenario uses a nucleotide substitution mutation rate, μ, of one divided by the length of the gene. This is approximately the mutation rate used in Ev [2]. For every nucleotide in every member of the population with probability μ, we replace that nucleotide with one of the three bases not currently at that position. Each alternative base has the same probability, 1/3, of being the replacement. All nucleotide positions have the same probability of being changed.

This process is repeated for 50,000 generations. Overall, the entire process is run for a million Monte Carlo simulations.

It should be noted that this process is intended to be a basic genetic algorithm. It makes no effort to include any of the more advanced features found in some genetic algorithms. It is also does not attempt to be true to the models of population genetics. It should not be taken as an attempt to form a biologically realistic model. It is simply an algorithm loosely based on Darwinian evolution. It is however close enough that some have argued that its success is indicative of the power of Darwinian evolution.

Human evolution with a realistic mutation rate. The second reproductive scenario adapts the genetic algorithm to more closely approximate the evolutionary transition from chimps to humans for a single gene. The population begins with 10,000 members, each having an identical allele. This is a difference from typical genetic algorithms, and the genetic algorithm reproductive scenario considered above. Typically, a genetic algorithm begins with a population with a diverse set of alleles, not just one. For each run of the scenario, this initial allele was constructed by randomly choosing a sequence of bases.

A variety of estimates have been given for the time since the divergence of humans and chimps [30]. By dividing the total time by the generation time, we can estimate the number of generations. Using the data given by Langergraber et al. [30] this gives estimates of approximately 50,000 to 700,000 generations. In this paper, the human evolution reproductive scenario used an estimate of 500,000 generations, which is generous since it allows more time for mutations to accumulate.

Simulating a 10,000 member population over 500,000 generations using typical genetic algorithm implementation techniques would be computationally expensive. Thus, the scenario instead keeps track merely of the number of members sharing a particular allele. This allows more efficient simulation. Let a_i be an allele, and c_i be the number of the members of the population that share this allele.

The simulation proceeds with the same three steps as the genetic algorithm. In the first step, the score of each allele is calculated based on the rules defined by the model.

For the second step, the fitness of a particular allele, a_i, is defined as $f_i = 1.01^{s_i}$ where s_i is the score defined by the model for allele a_i. Note that this is a less powerful selection force than used in the basic genetic algorithm.

Genetic algorithms often use a roulette wheel selection algorithm [29], where each member of the next generation is chosen randomly from the previous generation with a probability proportional to the fitness of that member. Adjusting for the use of counts of members sharing the same allele, the probability of a particular allele, a_i, being selected in this manner is

$$p_i = \frac{c_i f_i}{\sum_j c_j f_j}.$$

The number of members in the next generation, n_i, with a particular allele, a_i, is given by the binomial random variable:

$$n_i \sim B(10000, p_i).$$

While derived from genetic algorithms, readers familiar with population genetics will recognize this as a haploid Wright-Fisher model with selection [31–33].

This count can be approximated by the normal random variable with the same mean and variance [34]:

$$N(10000p_i, 10000p_i(1-p_i)).$$

The simulation uses this approximation to determine how many members of the next generation will have each allele. For each allele, the simulation generates a random variable on this normal distribution and rounds it to the nearest integer to determine how many members of the population will carry that allele in the next generation.

In the case of roulette selection, the total number of the members of the population remains constant. This is not true in this simulation because we have only approximated the original outcome of roulette selection. In particular, we have modeled the numbers of different alleles as independent random variables rather than dependent random variables. This could have been avoided at some computational cost. Since biological populations do not maintain a constant size, there did not appear to be a compelling reason to pay this cost.

For the third step, mutation, each base in the population is mutated with a 10^{-8} probability. Thus, on average 10^{-8} of all bases in the population are replaced by one of the three bases not currently at that position. Note that this mutation rate is orders of magnitude lower than that used in the basic genetic algorithm. All bases have an equal probability of being mutated. The probability of each of the other three bases is 1/3. Substitutionary mutations are the only mutations modeled; e.g. there are no insertion or deletion mutations. It is possible, though unlikely, for the same gene to receive multiple mutations.

The entire process was repeated 500,000 times to simulate 500,000 generations. This scenario was run via Monte Carlo simulation one million times for each of the models in order to evaluate its performance.

It should be noted that this model only attempts to make relatively minor adjustments to the genetic algorithm. It has a single common allele at the beginning of the simulation rather than many diverse alleles. It has a mutation rate approximately that of the human lineage. Selection is less powerful. It is not intended to accurately model biology, so much as to look at how existing evolutionary models react to realistic mutation rates.

Best of 20,000 human genes. One response to the problem of too few mutations is to point out that humans have at least 20,000 genes, thus giving much more opportunity to obtain several mutations that can combine to produce an increase in fitness [35]. In order to evaluate this idea, we can estimate the best performance among 20,000 genes (the method to be described below). For this, we take the optimistic assumption that all genes evolve independently. That is we assume that each gene evolves as if none of the other genes were present.

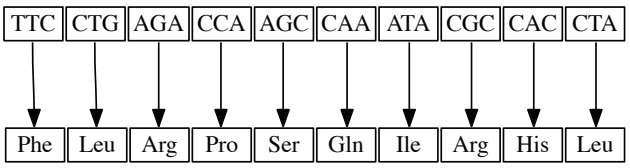

Figure 1: A depiction of the Acids model, showing each codon being translated into an amino acid. **doi:**10.5048/BIO-C.2015.1.f1

We assume that there is no linkage disequilibrium, or selection interference effects. Furthermore, we assume that all 20,000 genes do not have existing functionality that could be disrupted by evolution but rather start from scratch. Each gene starts with its own random sequence of bases. This set of assumptions should be greatly biased towards evolutionary processes adapting.

It would not be feasible to actually model all 20,000 genes evolving in Monte Carlo simulation as was done for the single gene. However, by taking advantage of the independence assumption we can use probability theory to estimate the best of 20,000 genes. For the highest score to be less than some arbitrary value b, all the individual scores of the 20,000 genes must also be less than b. The probability of an individual gene obtaining a score less than any particular value can be obtained via the Monte Carlo simulations done for the human evolution reproductive scenario. Since we have assumed that all the individual genes are independent, we can multiply the probabilities together to obtain the probability of the composite event. Thus, we can evaluate the distribution of the best score by using the formula:

$$\Pr[B < b] = \Pr[X < b]^{20000}$$

where B is the random variable representing the best score over 20,000 genes, and X is the random variable denoting the final score of a single gene.

The adapted models and their performance using the three reproductive scenarios

Target Acids. The Target Acids model is original to this paper, but is inspired by a program written by Richard Dawkins that evolved the phrase, "methinks it is like a weasel" [3]. That program evolved a sequence of English characters and selected for similarity to the target phrase. In this case, instead of evolving a particular English phrase, we evolve towards a target sequence of 100 randomly chosen amino acids. The actual target sequence is chosen randomly at the beginning of the simulation. Each allele consists of 300 bases that use the standard genetic code to encode 100 amino acids. Each codon of three bases is mapped using the standard genetic code. For the purposes of this experiment, we treat the stop codons as just another amino acid, it does not stop the gene. This process is depicted in Figure 1.

The score s_i of an allele a_i is the number of amino acids for which the actual amino acid encoded in the sequence matches the target amino acid. Within this model each codon is independent of the other codons. However, the bases in the codon are dependent on each other and must evolve in some coordinated fashion in order to identify the correct amino acid.

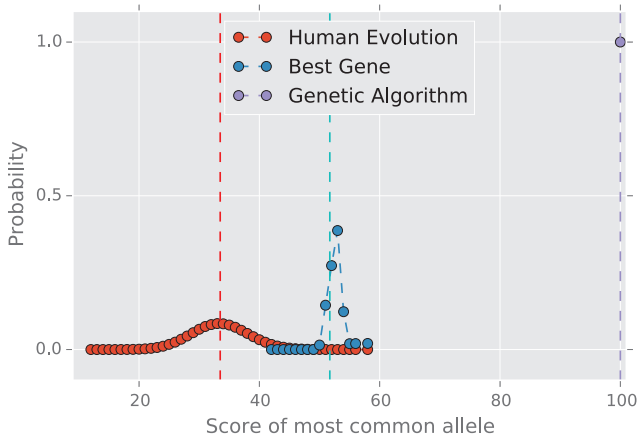

Figure 2: Probabilities of different scores for the various scenarios in the Target Acids model. The dashed vertical lines show the averages for the scenarios. The score is the number of amino acids that match the target sequence in the most common allele for each run, with a perfect score equaling 100. **doi:**10.5048/BIO-C.2015.1.f2

It should be emphasized that mutations are applied to the bases in the allele, and not to the string of amino acids directly. This is a deviation from the original Dawkins model, where mutations would replace letters in the string. In this case, mutations work at a lower level, replacing the bases that are then mapped into the amino acids.

This is not a biologically realistic model. The fitness landscape implied by this model is quite different from what we find in biology [36,37]. This model assumes that there is a single optimal protein, and the fitness of all other proteins derives simply from how similar they are to that protein. Neither of these is expected to be true. However, for the purposes of this paper, we are accepting the model as given and focusing only on the effect of mutation rates.

Under the standard genetic code, a randomly chosen codon has approximately a 4.8% chance of matching the randomly selected target amino acid already from the start. There is an additional 29.1% chance that codon can be made to match the target amino acid with only a single substitution. Thus there is approximately a 33.9% chance that a codon will either start correct or nearly correct given a random initialization. The remaining 66.1% of cases will require at least two mutations in a particular codon in order for it to match the target.

Figure 2 depicts the distribution of scores for the Target Acids model. Each point corresponds to the probability that a gene with that score was the most common allele when the simulation terminated. The genetic algorithm found the optimal solution in every case, finding all 100 correct amino acids. However, the genetic algorithm with parameters similar to human evolution averaged only 33.4% correct. Recall that 33.9% of codons were either correct or one substitution away from being correct in the initial random gene. The evolutionary process does well at evolving correct codons when only one change is necessary, but as more changes are necessary it no longer works as well. The best gene scenario obtains an average of 51.7% of codons correct, leaving the problem just over half solved.

Ev

The Ev model comes from Schneider [2]. This model already uses base encoding and thus did not need to be adapted. The mechanism is depicted in Figure 3. Each Ev allele is 261 bases. The first part of the Ev allele encodes a perceptron. The perceptron determines which sequences of six bases are considered binding sites. For example, ACGAGT might be considered a binding site by the perceptron, but TACTAC might not be. Each Ev allele encodes a different perceptron that specifies a different set of 6-base sequences which are considered binding sites. This paper does not discuss the details of the encoding of the perceptron. Understanding the precise details about how the perceptron is encoded and functions is not critical for understanding this paper. For further details, see the original paper [2].

At any point along the genome where a sequence that the perceptron accepts is found, a binding site will be formed. There are sixteen positions where the binding sites should correctly be located. It is possible to form a binding site anywhere along the genome, but only sixteen sites are supposed to be a binding site. The correct binding sites are chosen randomly when the simulation starts, all in the non-perceptron portion of the genome. This differs from the original version of Ev where the binding sites are typically fixed. In either model, the actual binding sites are evolved during the run of the simulation. The score, s_i of an allele, a_i, is the number of positions for which the perceptron made the correct determination. This means that a point is lost both for false positives and false negatives.

There are 256 possible positions for a binding site. The optimal genome thus receives a score of 256. If a binding site is formed at every position, it receives 16 points, corresponding to getting the 16 target binding sites correct. If no binding sites are formed, the score is 240, due to lacking the 16 target binding sites. It is relatively easy to produce a perceptron that will accept few sequences as valid binding sites. The evolutionary challenge is then to evolve the 16 correct binding sites such that they match one of these few sequences. As with codons in the Acids model, the bases for a potential binding site have to evolve in coordination in order to become a valid binding site.

Mutations are applied to the bases in the allele, and not to perceptron or binding sites directly. This is the same as the original model. As with the original model, there are no insertion or deletion mutations.

This is also not a biologically realistic model. Binding sites are not really encoded by perceptrons. Cellular structures do not really have a steady increase in functionality as binding sites

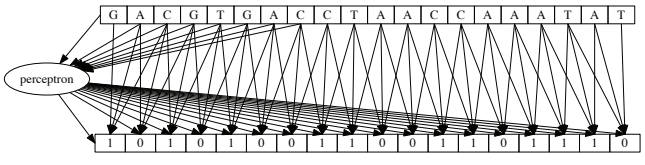

Figure 3: A depiction of the Ev model. The number of bases encoding the perceptron and potential bindings sites is reduced from the actual model for clarity of presentation. **doi:**10.5048/BIO-C.2015.1.f3

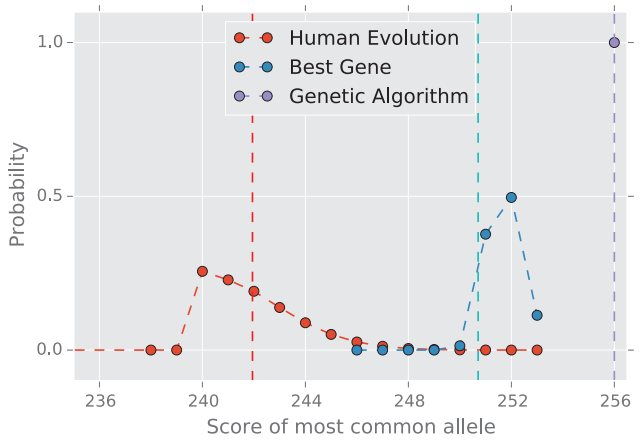

Figure 4: Probabilities of different scores for the various scenarios in Ev. The dashed vertical lines show the averages for the scenarios. The score is the number of positions in the genome for which the presence or absence of a binding site is correctly determined. **doi:**10.5048/BIO-C.2015.1.f4

are added to the correct locations. However, for the purposes of this paper we are accepting the model as given and focusing on the affects of mutation rates.

Figure 4 shows the results of running this model against the three scenarios. The genetic algorithm found the optimal solution in every run. However, a genetic algorithm using parameters closer to human evolution did not find the optimal solution in any runs. Rather, its peak in probability corresponded to 240 points. As noted above, a solution with no binding sites gets 240 points. The human evolution scenario is effective at eliminating the incorrect binding sites, but it does poorly at creating binding sites in the correct locations. Taking the best of 20,000 genes does better, successfully evolving an average of 10.7 of 16 binding sites, but it still fails to reach the optimal solution, instead leaving the problem just over half solved.

Avida

The Avida model is based on Avida as presented by Lenski et al. [1]. This model evolves computer programs that compute various tasks. This involves reading input, computing a function of that input, and outputting the result.

Fully understanding Avida's model would require familiarity with basic computer architecture and instruction set design.

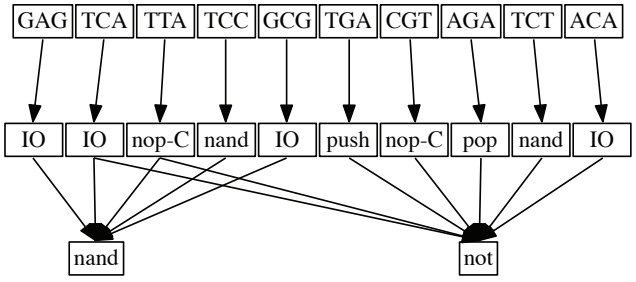

Figure 5: A depiction of the Avida model. **doi:**10.5048/BIO-C.2015.1.f5

For complete details see the original paper. However, for the purposes of this paper only a basic understanding of Avida is required. Within this model, the bases of the allele are mapped via something like the standard genetic code into instructions (analogous to amino acids). The instructions interact in a subtle and complicated way to determine the actual functionality of the program. This roughly resembles how the sequence of amino acids determines the folding and function of the protein. In the case of Avida, there are certain predefined tasks or functions. When a program manages one of these tasks, it receives an increase in its score. This is analogous to how a particular protein might be selected because it better resists disease, or improves the efficiency of energy transport.

The original Avida does not use base encoding, but rather encodes the program as a sequence of 26 possible instructions sometimes represented as English letters. This adaptation of the Avida model represents an Avida program as an allele of 300 bases. Taking inspiration from the standard genetic code, each codon of three bases is mapped to an Avida instruction. However, there are 64 possible codons, which is not evenly divisible by 26. For each run of the simulation, a new mapping between codons and the Avida instruction set is generated.

To generate the mapping, we begin by initializing a list of instructions. Each instruction from the Avida instruction set is added to this list twice, for a total of 52 instructions. Twelve additional instructions are added to the list using uniformly random selection with replacement from the instruction set twelve times. This produces a total of 64 instructions. The list of instructions is then randomly shuffled. AAA is then mapped to the first instruction in the list, AAC to the second, AAG to the third, and so on in lexicographical order.

A number of instructions in Avida relate to control flow or reproduction. This version of the Avida model is concerned only with the computational tasks, and while control flow and reproduction can interact with that, the logic for those instructions was not included. Instead, those instructions are simply skipped when executing the program.

In order to compute the score s_i of an allele a_i the model translates the sequence of bases into Avida instructions using the genetic code mapping discussed above. The instructions are executed according to the rules of the Avida virtual machine. For details on the semantics of the instructions, see the Avida paper [1]. The IO instruction outputs data from the virtual machine. If these outputs match the desired output for the built-in tasks, bonus points are awarded to the allele. For the definitions of the tasks, see the Avida paper. The number of points is the minimum number of nand operations required to perform the task. This is one point for "nand" or "not", two points for "and" or "or not," three points for "or" and "and not," four points for "nor" or "xor", and five points for "equ." The bonus for a task can only be earned once, and thus the total possible score is 25.

Mutations are applied to the bases in the allele, and not to the string of instructions. This is a deviation from the original model where mutations would replace instructions in the string. In this case, mutations work at a lower level, replacing

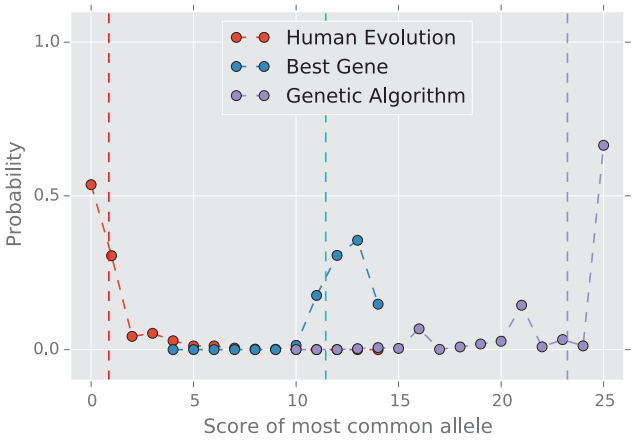

Figure 6: Probabilities of different scores for the various scenarios in Avida. The dashed vertical lines show the averages for the scenarios.
doi:10.5048/BIO-C.2015.1.f6

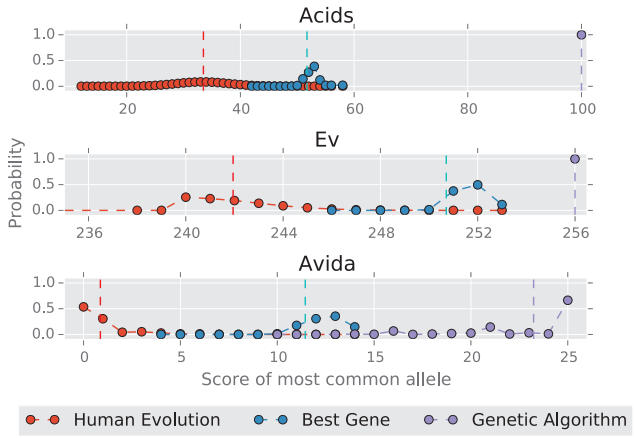

Figure 7: The Probabilities of different scores of the Target Acids, Ev, and Avida models under the various scenarios. The dashed lines are the averages for each combination of reproductive scenario and model.
doi:10.5048/BIO-C.2015.1.f7

the bases that are then mapped into the instructions. There are no insertion or deletion mutations, which is different from the original model.

This is not a biologically realistic model. Avida rewards partial versions of complete functionality in order to allow the evolution of complex functionality [8]. Each individual instruction performs a small task, making them better analogues to proteins than to amino acids [11]. However, for the purposes of this paper we are accepting the model as given and studying the effect of mutation rates.

The Avida model has two layers of coordination required in order to evolve the tasks. The bases in a particular codon have to mutate in a coordinated fashion in order to produce particular instructions. The instructions have to evolve in a coordinated fashion in order to perform various tasks.

Figure 6 shows the distribution of performance for Avida in the different scenarios. In this case, the standard genetic algorithm didn't always find an optimal solution, but does so a majority of the time. In contrast, the human evolution scenario fails to evolve anything in a majority of cases. Unlike the previous two models, there is no "easy" problem to solve like codons that almost match the target or incorrect binding sites that can be removed. The only way to gain points is to evolve code to perform new tasks, which evolution does poorly. The best of 20,000 genes obtained an average of 11.4 points. For comparison, obtaining all but the three hardest tasks earns the allele 12 points. As in the previous cases, the problem is just over half solved by the best of 20,000 genes.

Comparison of the three models

Figure 7 plots the performance of the three models for the three scenarios on the same graph. This is the same information as depicted in Figure 2, Figure 4, and Figure 6 brought together for ease of comparison. The horizontal axis represents the score obtained by the most common allele at the end of the simulation. The vertical axis represents the probability of obtaining that score. The dashed vertical lines are the average score for each model and scenario.

In each of the models, human evolution peaked at the transition between the part of the problem that evolution solves easily, and the point where evolution proceeds slowly. The Acids model readily found the correct codons when the codon was nearly correct to begin with. Ev readily disabled the incorrect binding sites. However, evolving codons that do not start out similar to target codons or new binding sites is a challenge. Avida lacks any easy part of its task, leading it to have a peak probability at zero, corresponding to accomplishing no tasks.

The human evolution scenario did not solve the simple problems considered here. In none of the million runs for each model did the optimal solution evolve. The average performance of each model is at best only a couple of points better than could be obtained by solving the easiest part of each problem. The human evolution scenario is insufficient to account for the evolution of the solutions to these problems

In contrast, the standard genetic algorithm, using the same gene and scoring logic, was able to solve the problem most of the time. The Ev and Target Acids models successfully evolved the optimal solution every time. Avida was able to evolve the correct solution in a majority of cases. Thus the optimal solutions are reachable, even if the human evolution scenario cannot reach them.

If we consider the highest performing of 20,000 genes evolving in parallel we get the best gene scenario. However, even the best of 20,000 genes fails to evolve the optimal solution. While performing better than the human evolution scenario, they still fall clearly short.

For any given allele, there is some number of available beneficial substitutions. These are the single point substitutions that would be beneficial according to the model's rules. By looking at the evolutionary history of the most common allele, we can calculate how many available beneficial substitutions were present at each point in the evolutionary history. This is done by

trying each possible substitution and counting which ones the model considers an improvement.

Figure 8 depicts the average number of available beneficial substitutions along with the average number of substitutions that were already in the genome of the ancestor of the most common allele in the final generation. Each model accumulates mutations over time, but at a decreasing rate. The number of available mutations decreases, explaining the decreasing rate of mutation accumulation. As mutations are fixed, new potential mutations do not appear to take their place. Instead, for each mutation fixed, the number of potential mutations decreases by more than one in many cases.

In the Acids model, there is initially an average of just over thirty available beneficial substitutions. This corresponds to approximately thirty of one hundred amino acids that we would expect to be one substitution away from being correct. Over the course of the simulation, almost all of these mutations are fixed in the population, but we do not observe new beneficial single point mutations becoming available.

In the Ev model, initially there is a large number of available beneficial mutations. These correspond to many possible ways to disable incorrect binding sites. After a few mutations are accumulated, all of the incorrect binding sites are disabled, and the model runs out of available beneficial substitutions.

In the Avida model, there is an average of three available beneficial substitutions. In most cases there are no available beneficial substitutions, but when a beneficial substitution is possible, there are usually several substitutions with the same effect.

The models are limited in their performance because they have a depleting supply of potential single-substitution beneficial mutations to work with. Further improvements are possible, but require making more than one substitution. It simply requires multiple changes for a codon to match a target amino acid, or for Ev to activate a binding site, or for Avida to perform a new task.

While possible for such multiple-substitution mutations to occur, it is unlikely. Genetic drift is modeled implicitly by the stochastic selection used in the genetic algorithm.. Some neutral mutations will become fixed in a population simply by random chance. In fact, under certain assumptions, this will occur at approximately the mutation rate [38]. The mutation

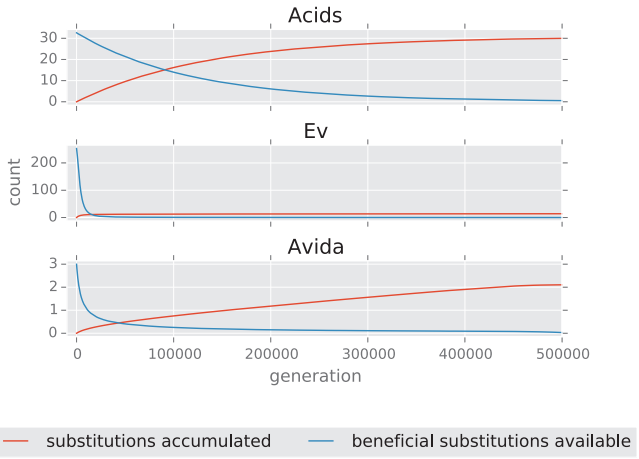

Figure 8: The substitutions accumulated and available beneficial substitutions over the evolutionary history for the three models in the human evolution scenario. **doi:**10.5048/BIO-C.2015.1.f8

rate per genome is the product of the size of gene and the mutation rate per nucleotide. Multiplying this by the number of generations should give the possible number of substitutions over all generations. The result of this calculation is depicted in Table 1. We can compare this to the average observed number of such substitutions in the same table. In each case the average observed number is somewhat lower than the expected value. This is due primarily to the presence of many non-neutral (beneficial or deleterious) mutations in each of the models. However, it is insufficient for a neutral mutation to be fixed, it must be potentiating, somehow enabling a later mutation to be beneficial. This is shown in the final row of the table, in each case approximately an order of magnitude less than the number of neutral mutations. Potentiating mutations are determined by looking at the evolutionary history of the most common allele in each run and counting the cases where a mutation was initially either deleterious or neutral when it first arose, but when reverted the final allele was deleterious. Thus these mutations were originally at best neutral, but eventually became beneficial.

The models do get potentiating mutations. However, the rate of potentiating mutations is less than one per gene, which means that any individual gene cannot expect potentiating

Table 1: Detail about neutral mutations in the various models.

Name	Acids	Ev	Avida
Gene length	300	261	300
Possible number of fixed neutral mutations	1.5	1.31	1.5
Observed average number of fixed non-beneficial* mutations	1.23	0.99	1.36
Observed average number of potentiating non-beneficial* mutations	0.200	0.066	0.134

*Non-beneficial mutations includes both neutral and deleterious mutations.

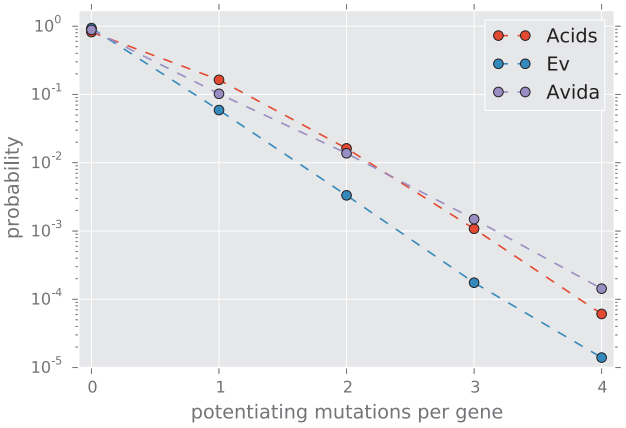

Figure 9: The probability of obtaining varying numbers of potentiating mutations per gene in the human evolution reproductive scenario across the three models. **doi:**10.5048/BIO-C.2015.1.f9

mutations. Figure 9 depicts the probability of obtaining multiple potentiating mutations per gene derived from counting the number of occurrences in the million runs of the human evolution scenario. The probability of obtaining multiple potentiating mutations decreases rapidly with the number of potentiating mutations.

How many potentiating mutations would be necessary to solve these simple problems? We can get a rough idea by looking at how much effect each potentiating mutation has as depicted in Figure 10. It depicts the increase in score for each of the three models given different numbers of potentiating mutation. Score increases with the number of potentiating mutations. However, the increase is slow, in most cases less than a point. For each of the models, three potentiating mutations produces about a two-point increase in score. This is far less than the amount necessary to solve the simple problems. Solving these simple problems does not require only a few potentiating mutations, but a steady diet of them.

DISCUSSION

The results demonstrate that we cannot simply dismiss the difference in mutation rates between biology and computer models as insignificant. The change in parameters converts a model that routinely finds optimal solutions to one which only solves the easiest part of the problem. If these parameters had been used in an attempt to demonstrate the power of evolution, the demonstration would have been less than convincing. If Dawkins had demonstrated the evolution of "thghmnksgiocdp lrfktaxxeahee", Schneider the evolution of two of sixteen binding sites, or Avida the evolution of zero tasks, the demonstrations would have been underwhelming. As such, the existing models fail to demonstrate a process that could account for complex adaptations. With realistic parameters, they fall far short of their goal.

Can an appeal to the best of 20,000 genes resolve the problem? The best of 20,000 genes is noticeably more successful than just focusing on a single gene. However, in each case the problem is roughly half solved. These are not highly complex or difficult problems. These are simple problems idealized to make evolution easier. The expectation is that any real biological problem will be harder, not easier to solve. Even with the entire resources available to human evolution focused on solving these problems, it only solves half of the problem. Even the best of 20,000 genes does not produce a workable model of biological adaptation.

What does this mean for actual biological human evolution? If it is not possible to construct a working Darwinian model of human evolution, then human evolution did not occur via a Darwinian process. However, the failure of existing models does not demonstrate that no such model can exist. It remains to be seen whether a working model can be constructed.

What would it take to produce a working model? In particular, how could a model avoid the challenges posed by potentiating mutation in these models? A model could attempt to increase the number of potentiating mutations that occur. It is easy to increase the number of potentiating mutations by increasing the mutation rate, but that renders the model biologically unrealistic. Any organism mutated at the rate of a typical genetic algorithm would be dead. However, these simulations did not include diploid genes, gene flow, environmental changes, insertion/deletion mutations, gene duplications, or any number of other possible factors. It is possible that one or more of these factors would increase the number of potentiating mutations. It remains to be seen whether a model can be built demonstrating the easy evolution of potentiating mutations as argued by Lynch and Abegg [27] or that potentiating mutations remain difficult as argued by many [23–26,28].

Another approach is to develop a model that does not require potentiating mutations. The simulations presented in this paper do show many cases of beneficial single point mutations successfully evolving. However, fully solving any of the

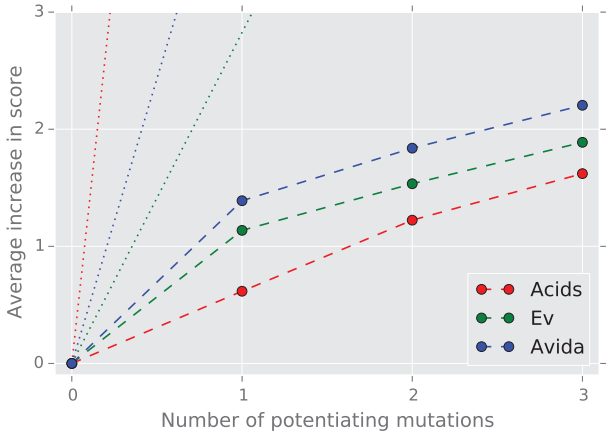

Figure 10: The average score for each model given the varying number of potentiated mutations in the human evolution reproductive scenario. The dashed lines show the averages from the experiments. The dotted lines show the necessary trajectories to solve the simple problem in four potentiating mutations. **doi:**10.5048/BIO-C.2015.1.f10

problems could not be accomplished by single substitution mutations alone. If all of the evolution necessary to produce humans required only the easy single-substitution changes that are seen to evolve in these simulations, then the problem is resolved. However, it takes very little complexity in the model to require potentiating mutations. Even translating codons into amino acids via the standard genetic code is sufficient. The beneficial mutations that are observed only solve part of deliberately simple problems. Developing a model that solves a non-trivial problem that does not require potentiating mutations would be difficult. Furthermore, this would imply that the hardest problem in human evolution is much easier than these simple problems. Many adaptations in nature and the lab have been demonstrated to require potentiating mutations [39–45]. Given the hundreds[1] of ways in which humans differ from the great apes, arguing that none of them required potentiating mutations seems implausible.

Given that potentiating mutations are rare and necessary for solving non-trivial problems, a possible solution is to develop a model where the large number of possible adaptations renders the rarity of each individual potentiating mutation moot. This idea is invoked by some Darwinists to account for biological evolution [35,43,44]. The models presented here do have multiple optimal solutions and adaptations but not sufficient to overcome the rarity of each individual adaptation. There are not nearly enough alternatives for the Darwinian account to work. Furthermore, while we do see the occasional potentiated adaptation in the lab or nature, there is no evidence for the multitude of alternatives invoked by Darwinists.

We have argued based on computer models and biological data that potentiating mutations are necessary for adaptation, individual potentiating mutations are very improbable, and there are only a handful available at any point in time. If these three facts are true, there is no way that Darwinism can account for human evolution. For Darwinism to be true, one or more will have to be overturned.

If the thesis of this paper is correct, further attempts to make biologically realistic models will repeat the theme here of being unable to solve simple problems. Furthermore, biological research will continue to document that even simple changes often require potentiating mutations. Only a small number of potentiated adaptations will be observed in very large populations. If the Darwinist claims are correct, other models will show that evolution can quite successfully solve simple or even complex problems. Biological research will show either that almost all adaptations do not require potentiating mutations or that a large number of potentiated mutational adaptations will be demonstrated, even in small populations.

Thus far, the available evidence strongly supports the claims in this paper. Biological evolution cannot resolve the challenge of potentiating mutations. Darwinian adaptation is not capable of explaining human evolution, or of comparable biological scenarios.

[1] http://carta.anthropogeny.org/moca/

1. Lenski RE, Ofria C, Pennock RT, Adami C (2003) The evolutionary origin of complex features. Nature 423:139–44. **doi**:10.1038/nature01568
2. Schneider TD (2000) Evolution of biological information. Nucleic Acids Res 28:2794–2799. **doi**:10.1093/nar/28.14.2794
3. Dawkins R (1986) The Blind Watchmaker: Why The Evidence Of Evolution Reveals A Universe Without Design. Norton (New York).
4. Hornby GS, Lohn JD, Linden DS (2011) Computer-automated evolution of an X-band antenna for NASA's Space Technology 5 mission. Evol Comput 19:1–23. **doi**:10.1162/EVCO_a_00005
5. Ewert W, Montañez G, Dembski WA, Marks II RJ (2010) Efficient per query information extraction from a Hamming oracle. 42nd Southeastern Symposium on System Theory (SSST 2010) IEEE:.290–297. **doi**:10.1109/SSST.2010.5442816
6. Ewert W, Dembski WA, Marks II RJ (2012) Climbing the Steiner tree–Sources of active information in a genetic algorithm for solving the Euclidean Steiner tree problem. BIO-Complexity 2012(1):1–14. **doi**:10.5048/BIO-C.2012.1
7. Montañez G, Ewert W, Dembski WA, Marks II RJ (2010) A vivisection of the **Ev** computer organism: Identifying sources of active information. BIO-Complexity 2010:1–6. **doi**:10.5048/BIO-C.2010.3
8. Ewert W, Dembski WA, Marks II RJ (2009) Evolutionary synthesis of nand logic: Dissecting a digital organism. 2009 IEEE International Conference on Systems, Man and Cybernetics. IEEE 2009:3047–3053. **doi**:10.1109/ICSMC.2009.5345941
9. Meyer SC (2004) The origin of biological information and the higher taxonomic categories. Proc Biol Soc Wash 117:213–239.
10. Meyer SC (2003) DNA and the origin of life: information, specification, and explanation. In: Darwinism, Design and Public Education pp 223–285.
11. Nelson CW, Sanford JC (2013) Computational evolution experiments reveal a net loss of genetic information despite selection. In: Marks II RJ, Behe MJ, Dembski WA, Gordon BL, Sanford JC, eds. Biological Information: New Pespectives. World Scientific pp 338–368. **doi**:10.1142/9789814508728_0014
12. Sanford JC, Baumgardner JR, Brewer WH (2013) Selection threshold severely constrains capture of beneficial mutations. In: Marks II RJ, Behe MJ, Dembski WA, Gordon BL, Sanford JC, eds. Biological Information: New Pespectives. World Scientific pp 264–297. **doi**:10.1142/9789814508728_0011
13. Nelson CW, Sanford JC (2011) The effects of low-impact mutations in digital organisms. Theor Biol Med Model 8: 9. **doi**:10.1186/1742-4682-8-9
14. Yuan X, Miller DJ, Zhang J, Herrington D, Wang Y (2012) An overview of population genetic data simulation. J Comput Biol 19:42–54. **doi**:10.1089/cmb.2010.0188
15. Sanford JC, Baumgardner JR, Brewer WH, Gibson P, Remine W (2007) Mendel's Accountant: a biologically realistic forward-time population genetics program. Scalable Comput Pract Exp 8:147–165.
16. Gibson P, Baumgardner JR, Brewer WH, Sanford JC (2013) Can purifying natural selection preserve biological information? In: Marks II RJ, Behe MJ, Dembski WA, Gordon BL, Sanford JC, eds. Biological Information: New Pespectives. World Scientific pp 232–263. **doi**:10.1142/9789814508728_0010
17. Brewer WH, Smith FD, Sanford JC (2013) Information loss: Potential for accelerating natural genetic attenuation of RNA viruses. In: Marks II RJ, Behe MJ, Dembski WA, Gordon BL, Sanford JC, eds. Biological Information: New Pespectives. World Scientific pp 369–384. **doi**:10.1142/9789814508728_0015

18. Baumgardner JR, Brewer WH, Sanford JC (2013) Can synergistic epistasis halt mutation accumulation? Results from numerical simulation. In: Marks II RJ, Behe MJ, Dembski WA, Gordon BL, Sanford JC, eds. Biological Information: New Pespectives. World Scientific pp 312–337. **doi**:10.1142/9789814508728_0013
19. Sanjuán R, Nebot MR, Chirico N, Mansky LM, Belshaw R (2010) Viral mutation rates. J Virol 84: 9733–48. **doi**:10.1128/JVI.00694-10
20. Drake JW, Charlesworth B, Charlesworth D, Culberson JC (1998) Rates of spontaneous mutation. Genetics 148: 1667–86.
21. Roach JC, Glusman G, Smit AFA, Huff CD, Hubley R, Shannon PT, et al (2010) Analysis of genetic inheritance in a family quartet by whole-genome sequencing. Science 328:636–639. **doi**:10.1126/science.1186802
22. Gauger AK, Axe DD, Luskin C (2012) Science & Human Origins. Discovery Institute Press (Seattle, Washington).
23. Behe MJ, Snoke DW (2004) Simulating evolution by gene duplication of protein features that require multiple amino acid residues. Protein Sci 13:2651–2664. **doi**:10.1110/ps.04802904
24. Behe MJ (2007) The Edge of Evolution. Free Press (New York, New York).
25. Durrett R, Schmidt D (2008) Waiting for two mutations: with applications to regulatory sequence evolution and the limits of Darwinian evolution. Genetics 180: 1501–1509. **doi**:10.1534/genetics.107.082610
26. Sanford JC, Brewer WH, Smith F, Baumgardner JR (2015) The waiting time problem in a model hominin population. Theor Biol Med Model 12:1–28. **doi**:10.1186/s12976-015-0016-z
27. Lynch M, Abegg A (2010) The rate of establishment of complex adaptations. Mol Biol Evol 27:1404–1414. **doi**:10.1093/molbev/msq020
28. Axe DD (2010) The limits of complex adaptation: An analysis based on a simple model of structured bacterial populations. BIO-Complexity 2010(4): 1–10. **doi**:10.5048/BIO-C.2010.4
29. Thomas B (1996) Evolutionary Algorithms in Theory and Practice. Oxford University Press (Oxford).
30. Langergraber KE, Prüfer K, Rowney C, Boesch C, Crockford C, Fawcett K, et al (2012) Generation times in wild chimpanzees and gorillas suggest earlier divergence times in great ape and human evolution. Proc Natl Acad Sci USA 109: 15716–15721. **doi**:10.1073/pnas.1211740109
31. Durrett R (2002) Probability Models for DNA Sequence Evolution. Springer Science & Business Media.
32. Ewens WJ (2004) Mathematical Population Genetics 1: Theoretical Introduction. Springer Science & Business Media.
33. Crow JF, Kimura M (2009) An Introduction to Population Genetics Theory. Blackburn Press.
34. Papoulis A, Pillai SU (2002) Probability, Random Variables and Stochastic Processes. 4th ed. McGraw-Hill (Europe).
35. Durrett R, Schmidt D (2008) Reply to Michael Behe. Genetics 181: 821–822. **doi**:10.1534/genetics.109.100800
36. Axe DD (2004) Estimating the prevalence of protein sequences adopting functional enzyme folds. J Mol Biol 341: 1295–315. **doi**:10.1016/j.jmb.2004.06.058
37. Axe DD (2010) The case against a Darwinian origin of protein folds. BIO-Complexity 2010(1): 1–12. **doi**:10.5048/BIO-C.2010.1
38. Kimura M (1968) Evolutionary rate at the molecular level. Nature 217:624–626.
39. Meyer JR, Dobias DT, Weitz JS, Barrick JE, Quick RT, Lenski RE (2012) Repeatability and contingency in the evolution of a key innovation in phage lambda. Science 335:428–32. **doi**:10.1126/science.1214449
40. Summers RL Dave A, Dolstra TJ, Bellanca S, Marchetti RV, Nash MN et al (2014) Diverse mutational pathways converge on saturable chloroquine transport via the malaria parasite's chloroquine resistance transporter. Proc Natl Acad Sci USA 111: E1759–67. **doi**:10.1073/pnas.1322965111
41. Blount ZD, Borland CZ, Lenski RE (2008) Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. Proc Natl Acad Sci USA 105:7899–7906. **doi**:10.1073/pnas.0803151105
42. Blount ZD, Barrick JE, Davidson CJ, Lenski RE (2012) Genomic analysis of a key innovation in an experimental *Escherichia coli* population. Nature 489: 513–518. **doi**:10.1038/nature11514
43. Harms MJ, Thornton JW (2014) Historical contingency and its biophysical basis in glucocorticoid receptor evolution. Nature 512:203–207. **doi**:10.1038/nature13410
44. Bridgham JT, Ortlund EA, Thornton JW (2009) An epistatic ratchet constrains the direction of glucocorticoid receptor evolution. Nature 461:515–519. **doi**:10.1038/nature08249
45. Bloom JD, Gong LI, Baltimore D (2010) Permissive secondary mutations enable the evolution of influenza oseltamivir resistance. Science 328:1272–1275. **doi**:10.1126/science.1187816