Weasel, a flexible program for investigating deterministic computer 'demonstrations' of evolution

Les Ey and Don Batten

In his book, The Blind Watchmaker, Richard Dawkins described a computer program and the results that he claimed demonstrated that evolution by random changes, combined with selection, was virtually inevitable.

The program described herein mimics Dawkins' program, but also provides the user with the opportunity to explore different values for the parameters such as the mutation rate, number of offspring, the selection coefficient, and the 'genome' size. Varying the values for these parameters shows that Dawkins chose his values carefully to get the result he wanted. Furthermore, the user can see that, with realistic values for the parameters, the number of generations needed to achieve convergence increases to such an extent that it shows that evolution of organisms with long generation times and small numbers of offspring is not possible even with a uniformitarian time-frame. And this is with a deterministic exercise, which cannot be a simulation of real-world evolution anyway. The program also allows the user to set up a target amino acid sequence with the mutations occurring in the DNA base pair order. Since there is redundancy in the triplet codons, the dynamics of the convergence are different to a simple alphabetical letter sequence. The program also allows for the user to include deletions and additions, as well as substitutions, as well as variable length in the 'evolving' sequence.

Cosmologist Sir Fred Hoyle (1915–2001) said the probability of the formation of just one of the many proteins on which life depends is comparable to that of the solar system packed full of blind people randomly shuffling Rubik's cubes all arriving at the solution at the same time.[1] In others words, it is impossible. In response to this huge problem for their naturalistic scenario, many evolutionists try to avoid the issue by breaking the evolution of proteins down into small and gradual steps. Richard Dawkins, a prominent atheist, is one such apologist.

Many introductory courses in biology at universities have *The Blind Watchmaker*, by Dawkins,[2] as required reading. The title, a play on William Paleys' watchmaker analogy, wherein Paley (1743–1805) argued that the complexity of living things demanded an intelligent creator, reveals Dawkins' aim—to rid his readers of any sense of a need for a Creator. The blind watchmaker is purely natural—mutation and natural selection. Dawkins' book is an undisguised polemic for atheism.

In this book, Dawkins presents a description of a computer program that generated the sequence of letters, 'METHINKS IT IS LIKE A WEASEL'[3] from a starting sequence of random letters. The process involves randomly changing letters in each 'generation' and selecting the 'offspring' closest to the target sequence. The mutation and selection process is repeated until the sequence is arrived at. This supposedly showed that evolution by cumulative selection of favourable random changes was inevitable, easy and fast.

At the time (1986) it was fairly showy to have a computer program to demonstrate something and many readers were duped into thinking that the program had proved something, not realizing that a program will do whatever its programmer designs it to do. Because of the deceptive nature of Dawkins' demonstration, several creationist authors saw the need to counter Dawkins' dupe.[4–6] These authors have pointed out reasons why Dawkins' program does not 'prove evolution'. It should be fairly obvious that any program that sets a target sequence of letters and then achieves it, by whatever means, has not demonstrated that the information in the sequence has arisen by some natural process not involving intelligence. The programmer specified the information; it did not arise from a 'simulation' of evolution.

Dawkins' program has apparently been lost. Evolutionist David Wise wrote a program that gave similar results to Dawkins' program.[7] Creationist Royal Truman created an Excel spreadsheet program that generated similar results to Dawkins' program.[8]

In this paper we describe a stand-alone program, *Weasel*, that closely mimics the one Dawkins describes, as well as providing a range of options for the user to explore—such as user-defined mutation rate, offspring number and selection coefficient. The program also provides for a peptide sequence target, with mutations occurring in the base sequence of a randomly generated DNA segment.

How Dawkins' program worked

To begin with, a target string of letters was chosen. Dawkins chose, 'METHINKS IT IS LIKE A WEASEL'.

Next, the computer generated a sequence of random uppercase letters to represent the original 'organism'. So, there were only 26 letters, plus a space, to choose from to generate the starting organism. This sequence always contained exactly the same number of letters as the target phrase—28 letters and spaces. The parent sequence would be copied, probably about 100 times (how many is not stated, but it must be a large number to get the results obtained), to represent reproduction. With each copy there would be a chance of a random error, a mutation, in the copying. Now for what was supposedly analogous to selection, each copy would now be tested to determine which copy was most like the target string 'METHINKS IT IS LIKE A WEASEL'. A copy would be chosen even if only one letter matched the target in the correct place, so long as it happened to be the best match.

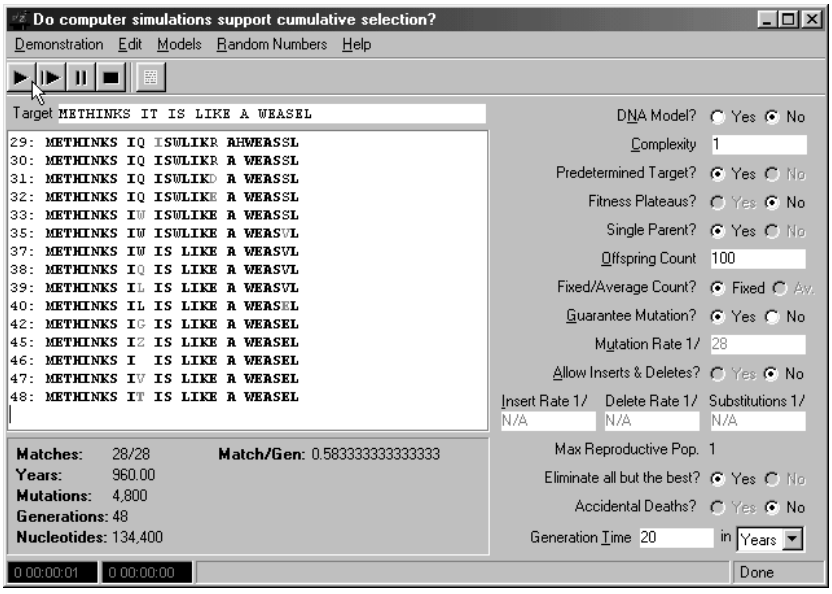

Figure 1. *A screen shot at the end of run of the Dawkins model, showing the user interface, the output window and status bars.*

The chosen copy would then be copied several times, again with introduced errors in the copying. In turn this 'progeny' was also tested to find the best match. This process would be repeated until a copy was found that matched the target exactly.

Weasel

Written in Borland Delphi by LE, Weasel was updated in 2015 to a JavaScript program, which can be downloaded from downloads.creation.com/zips/fp_extras_weasel.zip

Standard models available in Weasel

Under the Models menu item within Weasel, four models are available: Dawkins (default), error catastrophe, realistic mutation rates and DNA model.

Dawkins model (default)

In the Dawkins model (Fig. 1), the target sequence and parameters are set as per Dawkins' original exercise. Running the model will show convergence on the target usually in 30 to 60 generations (iterations). Since this is a probabilistic exercise involving a random starting sequence and random mutations, the result will vary with each run. The only addition to the original program concept here is the 'generation time'. Here the years for a generation can be entered and the program then calculates the time taken for the convergence on the target (obviously if your imaginary organism has a generation time of hours, then read the output bar at the bottom left as hours, not years).

Error Catastrophe model

Error catastrophe occurs when genetic information is destroyed by mutations at such a rate that all progeny are less fit than the parent/s so that selection cannot maintain the integrity of the genome and, in a Dawkinsian-type model, a target sequence cannot be achieved.

In the Error Catastrophe model, the offspring number is simply reduced from 100 to 10; all other parameters remain as in the Dawkins model. Because the number of offspring is low, the chances of a desirable mutation occurring in at least one offspring are reduced. Furthermore, as the model moves towards convergence, the probability of a mutation undoing what has been achieved rises to the point where it equals the probability of adding a desirable new mutation. So the model fails to converge.

The user can also induce error catastrophe by increasing the mutation rate after selecting the <no> option for <Guarantee Mutation?> One mutation in six letters per generation is about the error catastrophe point with 100 offspring. With 10 offspring the error catastrophe mutation rate drops to about 1 in 18. Increasing the length of the target letter sequence shows that the mutation rate has to be decreased in proportion to avoid error catastrophe.

To avoid error catastrophe, the mutation rate (per letter or base per generation) has to be inversely proportional to the size of the genome. That is, the larger the genome, the lower the mutation rate. Once this is factored into the theory, 'evolution' slows down to such a slow pace that it could never account for the amount of biological information in existence (the basic point of 'Haldane's Dilemma', which Walter ReMine spells out[9]).

With an amino acid sequence ('DNA model' under the <Models> menu item), with a small offspring number of say 10, the substitution mutation rate cannot be much more than one in the length of the target sequence. E.g., if the

target is 33 amino acids (99 base pairs), a mutation rate of 1 in 50 produces error catastrophe. So the Dawkins model will converge with a mutation rate of 1 in 28 with a target of 28 letters, but not on a genome just a little bit bigger and certainly not with a human-sized genome of $3x10^9$ nucleotides.

Adjusted mutation rate model

In effect, the mutation rate cannot be much greater than one per genome per generation. This then severely limits the rate of progress from a chimp-like species to human, if this were possible, even with perfect selection and all the other assumptions.

Real-world mutation rates are many orders of magnitude less than used in Dawkins' model, or others supposed simulations of evolution for that matter. Spetner summarizes the knowledge on actual rates of mutation as follows:

> 'In bacteria the mutation rate per nucleotide is between 0.1 and 10 per billion transcriptions [refs]. But in all other forms of life the rate is smaller. For organisms other than bacteria, the mutation rate is between 0.01 and 1 per billion [ref.].'[10]

We expect that the reason for this difference between bacteria and other organisms relates to genome size: bacteria have the smaller genomes and can therefore sustain higher mutation rates without error catastrophe.

Biological replication is extremely accurate. This level of accuracy is due to the processes of proof reading and error correction. This is vital since mutations disorder existing functional DNA sequences, and are therefore overwhelmingly harmful (and even rare beneficial mutations are the result of information loss).

The Adjusted Mutation Rate model shows what happens when a more realistic mutation rate is applied to Dawkins' model. A mutation rate of 1 in 100,000,000 (10 per billion letters) means that the model takes a long time to run. It could take a few weeks on a typical slower PC. Of course the Adjusted Mutation Rate model is still somewhat unrealistic, being the upper limit estimated for bacteria, but it helps to illustrate the point that real life is nothing like the Dawkins model.

To cope with realistically low mutation rates, a suitable pseudo random number generator had to be found to replace the one provided in Delphi, which started to repeat the pattern before the end of a typical run. The Mersenne Twister pseudo-random number algorithm[11] generates a pattern that repeats every 2^{19937} numbers and distributes the numbers more evenly than Delphi's internal generator. This makes it possible for mutation rates down to 1 in 10^{10} to be resolved.

DNA Model

Any standard biochemistry text would describe how proteins are made from the information contained in the base sequences on DNA. We have provided a brief tutorial provided with the program (under <Help>). An important difference between the DNA model and Dawkins' Model, or any alphabet model, is that the DNA of an organism is not compared directly with the target as it is in alphabetical model. Another important factor is redundancy, some of the amino acids can be coded by different codons. With some codons, only the first two base pairs are needed to determine which amino acid is produced. This gives the genetic code some resistance to change. In some cases you would require more than one mutation to convert the code of one expressed amino acid into the code for another.

In running the DNA model, even though there are only four possible 'letters' compared to 27 in the Dawkins model, a target requiring 30 base pairs takes close to twice the number of generations to be reached compared to Dawkins' target of 28 letters.

The user can enter their own amino acid sequence—the program has an editor to assist in this—and adjust the various parameters to see what happens.

One of the big differences in DNA mutations is that stop codons can be generated. These effectively truncate the sequence if occurring within the sequence rather than at the end.

Irreducible Complexity

Behe uses a mousetrap to illustrate the concept of irreducible complexity.[12] He points out that the individual parts of a mousetrap cannot function independently of each other. If you remove a single part or change its dimensions to a significant degree, the mousetrap will fail to function at all. This (among other issues) makes a mousetrap resistant to any step-wise explanation of its origin; that, of course, is without the aid of an intelligent designer. Dawkins' weasel analogy, and all other evolutionary story telling, fails to address this issue. It does not demonstrate how a suite of interdependent proteins can evolve in parallel to a point where functionality appears.

In Part II of his book, Behe discusses several irreducibly complex systems, such as blood clotting, where there is no conceivable gradual build up of functionality. For example, the proteins involved in blood clotting are required to act in unison. It's not just a case of a slight lack of functionality if an essential protein is missing because the whole system is finely balanced. On the one hand, if one component is missing an animal could bleed to death; on the other hand, all of the animal's blood could become one massive blood clot. These kinds of systems are all-or-nothing systems. Dawkins' weasel model assumes functionality for any and every step in the run of the model with the only requirement for selection being greater likeness to the pre-specified goal.

Michael Behe addresses Dawkins' response to Paley's argument for the irreducible complexity of a watch and the need for an intelligent designer:

> 'Neither Darwin nor Dawkins, neither science

> nor philosophy, has explained how an irreducibly complex system such as a watch might be produced without a designer.'[13]

Dawkins' concept of a slow, gradual build up of functionality is not valid for a system of proteins that have no function at all until all the proteins are present in the correct amounts and at the same time. Indeed almost every biochemical pathway is irreducibly complex. There is hardly a trait in a living organism that is independent of other traits for its function.

The <**Complexity**> option in the program allows the user to specify how many of the target letters or amino acids have to be present together for an increase in 'fitness'. This enables some recognition of the fact that not every point mutation can be adaptive in the change from one sequence to another. It does not address irreducible complexity at the system level. With <Complexity> set at three, for example, a mutant with one of the target letters added could not be selected against one without the letter. Nor would another mutant with two letters. Only if three new target letters were present together would the mutant be selected. With a setting of three, the number of generations for convergence for Dawkins' model blows out to about 30,000, or about 600,000 years for human generation times—and this is with perfect selection, high mutation rate and 100 offspring!

The <**Allow inserts and deletes**> option allows the user to specify that insertions and deletions are allowed, and the rates of occurrence per mutation event. For example, a deletion rate of 1 in 3 means that one in three mutations will be deletions. If four is then entered for insertions, then the rate for substitutions has to be 1:2.4, because a mutation can only be a substitution, deletion or an insertion. This option only applies to the adjusted mutation rate and DNA models (where <Guarantee mutation?> is set to 'No'), not the basic Dawkins model. The user has to be a little judicious in selection of values for each these rates. For example, if a high insertion rate is used with a low deletion rate, at a high mutation rate, the sequence can diverge further and further away from the target as the lengths of the genomes of the offspring get longer and longer. This is another cause of error catastrophe.

Other buttons on the user interface merely underline other limitations of this computer modelling of 'evolution':

- **Fitness plateaus?** For forelimbs to change into wings, for example, there would have to be a decrease in the functionality as legs prior to there being any increase in functionality as wings. Consequently, evolution from a tetrapod to a bird would require that the transitional animal would have to move from a fitness peak as a functional tetrapod into a fitness valley (less fit to survive) at some stage during the transition. This is a huge problem for evolutionary scenarios (ReMine discusses this[14]).
- **Single parent?** The Dawkins' model, and ours, assumes asexual reproduction. This makes the offspring number in Dawkins' model even more unrealistic, because the offspring number per generation for one bacterium is one. Sexual reproduction introduces other problems for evolutionary scenarios. These include genetic drift, wherein because only half the genetic information in a parent is passed onto each offspring, there is a significant likelihood of a given information-adding mutation being lost from a population. The other problem relates to recessive traits, where two copies of a gene need to be present for it to be expressed. Even if a male and female having one copy of the recessive gene happen to find each other to mate, the chances of an offspring receiving two copies is only one in four. This greatly slows down the rate of substitution of a new trait into a population—this is part of 'Haldane's Dilemma' mentioned earlier.
- **Fixed/average count?** Is the offspring count exactly that specified for every generation, or does it vary with a mean of the set value?
- **Guarantee mutation?** In Dawkins' model, there was apparently one mutation per offspring, and only one, in every offspring. The only random factor was the choice of which letter position would be changed and what it would be changed to. This is not the real world. A given offspring might actually receive two mutations, or none at all. When the mutation rate is not fixed, the number of generations needed for convergence increases. For example, with guaranteed mutation, 15 runs of the Dawkins model took an average of 46 generations to converge, whereas without guaranteed mutation, it took an average of 82 runs.
- **Eliminate all but the best?** This states that the selection coefficient is 1.0 in each and every generation. In other words, perfect selection operates. The sequence closest to the target is 100% fit to survive and all the others have 0% fitness—none of them survive. This is not the real world. A more typical real-world generous selection coefficient would be 0.01. If this could be factored in, the number of generations would multiply enormously.
- **Accidental death?** No allowance is made for the accidental death of the surviving organism in each generation.

There is of course also no allowance made for lethal mutations. The exercise assumes that in every generation, one offspring will have 100% fitness.

There are other limitations to this approach; limitations because it is a computer programming exercise. One area not covered is that mutations and selection occur in populations, not just in one individual's offspring in each generation. This aspect introduces the whole area of population genetics. With a large population, desirable mutations are more likely—this can be seen by increasing the number of <**Offspring count**> in the different models. However, the larger the population, the longer it takes for the new gene variant to take over the population, where all the individuals without the mutation have to die off (this is with realistic selection coefficients), and the more likely that it will be lost through genetic drift, etc.

Other problems for simplistic models of gene evolution are also ignored: issues such as pleiotropy (one gene

affecting several different traits) and polygeny (two or more genes working together to affect a trait). Another problem is multiple-coding genes, where the same sequence of DNA can be read using different frames, or the complementary strand read, or read backwards, or the messenger RNA edited (alternative splicing), to produce different proteins.[15] Evolving a gene for one protein with one function is difficult enough; evolving one that can produce several different functional proteins would have to be completely out of the question. Following are some other problems that are ignored: the complexities of gene control (producing a new protein without control over the amount would not be very helpful), mutation hotspots (many base sequences are quite resistant to mutations; others are quite prone), the intron / exon structure of many eukaryotic genes (where introns are removed from the messenger RNA before protein synthesis—signals have to be coded into the DNA to control this editing) and the necessity of new control systems to destroy the new proteins when their job is finished.[6]

Conclusion

Dawkins' weasel program does not generate any new information—the information was completely specified in the target phrase. The target phrase is effectively a mould that is used to shape the virtual species. Perfect selection that is goal based hammers this 'species' until it is forged into the likeness of the predetermined target. There is no mould that natural selection can use. The program uses many such unrealistic assumptions that all contribute to making evolution look easy, even inevitable. When the parameters of Dawkins' weasel analogy are modified, it can be seen how carefully Dawkins chose the values for the parameters. Far from demonstrating how inevitable evolution is, the program presented here can be used to show that in realistically sized genomes error catastrophe is a major hindrance to the speed at which evolution could occur, even when ignoring all the other unrealistic assumptions. With realistic mutation rates, the program shows how slow evolution would be, even given the remaining unrealistic constraints, such as perfect selection.

Added to that, the issue of irreducible complexity makes it clear that the vast amount of biological information that we see in organisms today could not have arisen from random processes, even with natural selection to supposedly aid the process.

Spetner points out[16] that no one has found a single point mutation that adds biological information (specified complexity). This is not to say that such a mutation cannot or does not happen, just that such mutations cannot be the mechanism for generating the vast amount of biological information that we see.

References

1. Hoyle, F., The Big Bang in astronomy, *New Scientist* **92(**1280):527, 1981; Hoyle's obituary, Damme, G. and Sarfati, J., Big bang critic dies, *TJ* **15**(3):6–7, 2001.
2. Dawkins, R., *The Blind Watchmaker*, Norton, New York, 1986.
3. From Shakespeare's *Hamlet*.
4. Gitt, W. and Wieland, C., Weasel words, *Creation* **20**(4):20–21, 1998.
5. Truman, R., Dawkins' weasel revisited, *CEN Tech. J.* **12**(3):358–361, 1998.
6. Bergman, J., Why Dawkins' weasel demonstrates mutations cannot produce a new functional gene, *TJ* **15**(2):69–76.
7. The DOS program was apparently made available through the atheist funded and operated anti-creationist organisation, National Center for Science Education in the USA.
8. Truman, Ref. 5.
9. ReMine, W., *The Biotic Message*, Saint Paul Press, Saint Paul, chapters 7–9, 1996; review by Batten, D., *CEN Tech. J.* **11**(3):292–298, 1997.
10. Spetner, L.M., *Not by Chance,* The Judaica Press, New York, p. 92, 1997; review by Wieland, C., *Creation* **20**(1):50–51, 1997.
11. Copyright (C) 2000, Andrew N. Driazgov <andrey@asp.tstu.ru>.
12. Behe, M., *Darwin's Black Box*, The Free Press, New York, pp. 42, 43, 1996; review by Ury, H., *CEN Tech. J.* **11**(3):283–291, 1997.
13. Behe, Ref. 12, p. 213.
14. ReMine, Ref. 9, Chapter 5.
15. The human genome project has identified only about 35,000 genes, but humans can produce upwards of 100,000 proteins. Just how all the extra proteins are coded for is unknown. Batten, D., Catchpoole, D. and Wieland, C., Message mania, *Creation* **23**(3):16–19, 2001.
16. Spetner, Ref. 10, p.

Les Ey has worked as a relational database and applications programmer (Delphi) since 1995. He is involved in software systems from design, programming, testing, upgrading and customer support. Les has a certificate in Business Computing and has worked on an accounting system, an automated data replication system for a database called AREV and CGI software for a Web site with online ordering. His main interests are apologetics, computing and science.

Don Batten has a Ph.D. in plant physiology from the University of Sydney and has worked as a research scientist for 20 years. He co-wrote and edited *The Answers book* and co-authored *One Blood*. Don has written many articles for *Creation* magazine, the *AiG* Web site and *TJ*. He is on the editorial panel of *TJ* and consults on the production of *Creation*, and frequently speaks on Christian worldview, Creation evidences and the relevance to Creation.