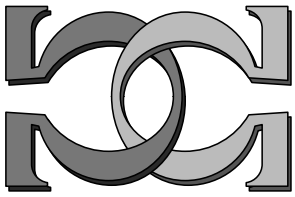
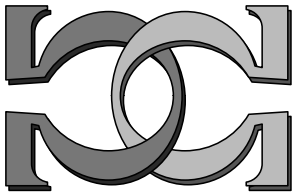
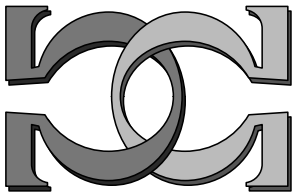
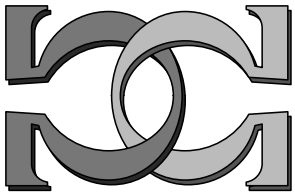


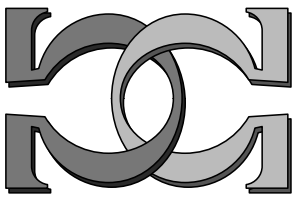


CDMTCS
Research
Report
Series

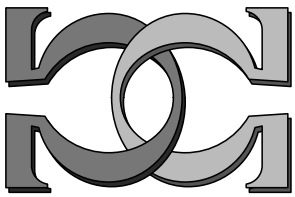




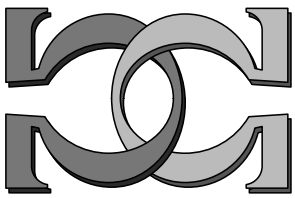
Speculations on Biology, Information and Complexity



G. J. Chaitin
IBM Research, Yorktown Heights, USA



CDMTCS-282
July 2006



Centre for Discrete Mathematics and
Theoretical Computer Science

Speculations on Biology, Information and Complexity

G. J. Chaitin
IBM Research

Abstract

It would be nice to have a mathematical understanding of basic biological concepts and to be able to prove that life must evolve in very general circumstances. At present we are far from being able to do this. But I'll discuss some partial steps in this direction plus what I regard as a possible future line of attack.

Can Darwinian evolution be made into a mathematical theory? Is there a fundamental mathematical theory for biology?

Darwin = math ?!

In 1960 the physicist Eugene Wigner published a paper with a wonderful title, "The unreasonable effectiveness of mathematics in the natural sciences." In this paper he marveled at the miracle that pure mathematics is so often extremely useful in theoretical physics.

To me this does not seem so marvelous, since mathematics and physics co-evolved. That however does not diminish the miracle that at a fundamental level Nature is ruled by simple, beautiful mathematical laws, that is, the miracle that Nature is comprehensible.

I personally am much more disturbed by another phenomenon, pointed out by I.M. Gel'fand and propagated by Vladimir Arnold in a lecture of his that is available on the

web, which is the stunning contrast between the relevance of mathematics to physics, and its amazing lack of relevance to biology!

Indeed, unlike physics, biology is not ruled by simple laws. There is no equation for your spouse, or for a human society or a natural ecology. Biology is the domain of the complex. It takes 3×10^9 bases $= 6 \times 10^9$ bits of information to specify the DNA that determines a human being.

Darwinian evolution has acquired the status of a dogma, but to me as a mathematician seems woefully vague and unsatisfactory. What is evolution? What is evolving? How can we measure that? And can we **prove**, mathematically prove, that with high probability life must arise and evolve?

In my opinion, if Darwin's theory is as simple, fundamental and basic as its adherents believe, then there ought to be an equally fundamental mathematical theory about this, that expresses these ideas with the generality, precision and degree of abstractness that we are accustomed to demand in pure mathematics.

Look around you. We are surrounded by evolving organisms, they're everywhere, and their ubiquity is a challenge to the mathematical way of thinking. Evolution is not just a story for children fascinated by dinosaurs. In my own lifetime I have seen the ease with which microbes evolve immunity to antibiotics. We may well live in a future in which people will again die of simple infections that we were once briefly able to control.

Evolution seems to work remarkably well all around us, but not as a mathematical theory!

In the next section of this paper I will speculate about possible directions for modeling evolution mathematically. I do not know how to solve this difficult problem; new ideas are needed. But later in the paper I will have the pleasure of describing a minor triumph. The program-size complexity viewpoint that I will now describe to you does have some successes to its credit, even though they only take us an infinitesimal distance in the direction we must travel to fully understand evolution.

A software view of biology: Can we model evolution via evolving software?

I'd like to start by explaining my overall point of view. It is summarized here:

Life = Software ?

program $\longrightarrow$ *COMPUTER* $\longrightarrow$ output
DNA $\longrightarrow$ *DEVELOPMENT/PREGNANCY* $\longrightarrow$ organism

(Size of program in bits) $\approx$ (Amount of DNA in bases) $\times$ 2

So the idea is firstly that I regard life as software, biochemical software. In particular, I focus on the digital information contained in DNA. In my opinion, DNA is essentially a programming language for building an organism and then running that organism.

More precisely, my central metaphor is that DNA is a computer program, and its output is the organism. And how can we measure the complexity of an organism? How can we measure the amount of information that is contained in DNA? Well, each of the successive bases in a DNA strand is just 2 bits of digital software, since there are four possible bases. The alphabet for computer software is 0 and 1. The alphabet of life is A, G, C, and T, standing for adenine, cytosine, guanine, and thymine. A program is just a string of bits, and the human genome is just a string of bases. So in both cases we are looking at digital information.

My basic approach is to measure the complexity of a digital object by the size in bits of the smallest program for calculating it. I think this is more or less analogous to measuring the complexity of a biological organism by 2 times the number of bases in its DNA.

Of course, this is a tremendous oversimplification. But I am only searching for a toy model of biology that is simple enough that I can prove some theorems, not for a detailed theory describing the actual biological organisms that we have here on earth. I am searching for the Platonic essence of biology; I am only interested in the actual creatures we know and love to the extent that they are clues for finding ideal Platonic forms of life.

How to go about doing this, I am not sure. But I have some suggestions.

It might be interesting, I think, to attempt to discover a toy model for evolution consisting of evolving, competing, interacting programs. Each organism would consist of a single program, and we would measure its complexity in bits of software. The only problem is how to make the programs interact! This kind of model has no geometry, it leaves out the physical universe in which the organisms live. In fact, it omits bodies and retains only their DNA. This hopefully helps to make the mathematics more tractable. But at present this model has no interaction between organisms, no notion of time, no dynamics, and no reason for things to evolve. The question is how to add that to the model.

Hopeless, you may say. Perhaps not! Let's consider some other models that people have proposed. In von Neumann's original model creatures are embedded in a cellular

automata world and are largely immobile. Not so good! There is also the problem of dissecting out the individual organisms that are embedded in a toy universe, which must be done before their individual complexities can be measured. My suggestion in one of my early papers that it might be possible to use the concept of mutual information—the extent to which the complexity of two things taken together is smaller than the sum of their individual complexities—in order to accomplish this, is not, in my current opinion, particularly fruitful.

In von Neumann's original model we have the complete physics for a toy cellular automata universe. Walter Fontana's ALChemy = algorithmic chemistry project went to a slightly higher level of abstraction. It used LISP S-expressions to model biochemistry. LISP is a functional programming language in which everything—programs as well as data—is kept in identical symbolic form, namely as what are called LISP S-expressions. Such programs can easily operate on each other and produce other programs, much in the way that molecules can react and produce other molecules.

I have a feeling that both von Neumann's cellular automata world and Fontana's algorithmic chemistry are too low-level to model biological evolution.[1] So instead I am proposing a model in which individual creatures are programs. As I said, the only problem is how to model the ecology in which these creatures compete. In other words, the problem is how to insert a dynamics into this static software world.[2]

Since I have not been able to come up with a suitable dynamics for the software model I am proposing, I must leave this as a challenge for the future and proceed to describe a few biologically relevant things that I **can** do by measuring the size of computer programs. Let me tell you what this viewpoint can buy us that is a tiny bit biologically relevant.

Pure mathematics has infinite complexity and is therefore like biology

Okay, program-size complexity can't help us very much with biological complexity and evolution, at least not yet. It's not much help in biology. But this viewpoint has been developed into a mathematical theory of complexity that I find beautiful and compelling—since I'm one of the people who created it—and that has important applications in another major field, namely metamathematics. I call my theory *algorithmic*

[1] A model with perhaps the opposite problem of being at too **high** a level, is Douglas Lenat's AM = Automated Mathematician project, which dealt with the evolution of new mathematical concepts.

[2] Thomas Ray's *Tierra* project did in fact create an ecology with software parasites and hyperparasites. The software creatures he considered were sequences of machine language instructions coexisting in the memory of a single computer and competing for that machine's memory and execution time. Again, I feel this model was too low-level. I feel that too much micro-structure was included.

information theory, and in it you measure the complexity of something X via the size in bits of the smallest program for calculating X, while completely ignoring the amount of effort which may be necessary to discover this program or to actually run it (time and storage space). In fact, we pay a severe price for ignoring the time a program takes to run and concentrating only on its size. We get a beautiful theory, but we can almost never be sure that we have found the smallest program for calculating something. We can almost never determine the complexity of anything, if we chose to measure that in terms of the size of the smallest program for calculating it!

This amazing fact, a modern example of the incompleteness phenomenon first discovered by Kurt Gödel in 1931, severely limits the practical utility of the concept of program-size complexity. However, from a philosophical point of view, this paradoxical limitation on what we can know is precisely the most interesting thing about algorithmic information theory, because that has profound epistemological implications.

The jewel in the crown of algorithmic information theory is the halting probability Ω, which provides a concentrated version of Alan Turing's 1936 halting problem. In 1936 Turing asked if there was a way to determine whether or not individual self-contained computer programs will eventually stop. And his answer, surprisingly enough, is that this cannot be done. Perhaps it can be done in individual cases, but Turing showed that there could be no general-purpose algorithm for doing this, one that would work for all possible programs.

The halting probability Ω is defined to be the probability that a program that is chosen at random, that is, one that is generated by coin tossing, will eventually halt. If no program ever halted, the value of Ω would be zero. If all programs were to halt, the value of Ω would be one. And since in actual fact some programs halt and some fail to halt, the value of Ω is greater than zero and less than one. Moreover, Ω has the remarkable property that its numerical value is maximally unknowable. More precisely, let's imagine writing the value of Ω out in binary, in base-two notation. That would consist of a binary point followed by an infinite stream of bits. It turns out that these bits are irreducible, both computationally and logically:

- You need an N-bit program in order to be able to calculate the first N bits of the numerical value of Ω.
- You need N bits of axioms in order to be able to prove what are the first N bits of Ω.
- In fact, you need N bits of axioms in order to be able to determine the positions and values of any N bits of Ω, not just the first N bits.

Thus the bits of Ω are, in a sense, mathematical facts that are *true for no reason,* more precisely, for no reason simpler than themselves. Essentially the only way to determine

the values of some of these bits is to directly add that information as a new axiom.

And the only way to calculate individual bits of Ω is to separately add each bit you want to your program. The more bits you want, the larger your program must become, so the program doesn't really help you very much. You see, you can only calculate bits of Ω if you already **know** what these bits are, which is not terribly useful. Whereas with $\pi = 3.1415926\ldots$ we can get **all** the bits or all the digits from a single finite program, that's all you have to know. The algorithm for π compresses an infinite amount of information into a finite package. But with Ω there can be no compression, none at all, because there is absolutely no structure.

Furthermore, since the bits of Ω in their totality are infinitely complex, we see that pure mathematics contains infinite complexity. Each of the bits of Ω is, so to speak, a complete surprise, an individual atom of mathematical creativity. Pure mathematics is therefore, fundamentally, much more similar to biology, the domain of the complex, than it is to physics, where there is still hope of someday finding a theory of everything, a complete set of equations for the universe that might even fit on a T-shirt.

In my opinion, establishing this surprising fact has been the most important achievement of algorithmic information theory, even though it is actually a rather weak link between pure mathematics and biology. But I think it's an actual link, perhaps the first.

Computing Ω in the limit from below as a model for evolution

I should also point out that Ω provides an extremely abstract—much too abstract to be satisfying—model for evolution. Because even though Ω contains infinite complexity, it can be obtained in the limit of infinite time via a computational process. Since this extremely lengthy computational process generates something of infinite complexity, it may be regarded as an evolutionary process.

How can we do this? Well, it's actually quite simple. Even though, as I have said, Ω is maximally unknowable, there is a simple but very time-consuming way to obtain increasingly accurate lower bounds on Ω. To do this simply pick a cut-off t, and consider the finite set of all programs p up to t bits in size which halt within time t. Each such program p contributes $1/2^{|p|}$, 1 over 2 raised to p's size in bits, to Ω. In other words,

$$\Omega \quad = \quad \lim_{t \longrightarrow \infty} \quad \left(\sum_{|p| \le t \text{ \& halts within time } t} 2^{-|p|} \right).$$

This may be cute, and I feel compelled to tell you about it, but I certainly do not regard this as a satisfactory model for biological evolution, since there is no apparent connection with Darwin's theory.

References

The classical work on a theoretical mathematical underpinning for biology is von Neumann's posthumous book [2].[3] Interestingly enough, Francis Crick—who probably contributed more than any other individual to creating modern molecular biology—for many years shared an office with Sydney Brenner, who was aware of von Neumann's thoughts on theoretical biology and self-reproduction. This interesting fact is revealed in the splendid biography of Crick [3].

For a book-length presentation of my own work on information and complexity, see [4], where there is a substantial amount of material on molecular biology. This book is summarized in my recent article [5], which however does not discuss biology. A longer overview of [4] is my Alan Turing lecture [6], which does touch on biological questions.

For my complete train of thought on biology extending over nearly four decades, see also [7,8,9,10,11].

For information on *Tierra,* see Tom Ray's home page at `http://www.his.atr.jp/~ray/`. For information on ALChemy, see `http://www.santafe.edu/~walter/AlChemy/papers.html`. For information on Douglas Lenat's Automated Mathematician, see [12] and the Wikipedia entry `http://en.wikipedia.org/wiki/Automated_Mathematician`.

For Vladimir Arnold's provocative lecture, the one in which Wigner and Gel'fand are mentioned, see `http://pauli.uni-muenster.de/~munsteg/arnold.html`. Wigner's entire paper is itself on the web at `http://www.dartmouth.edu/~matc/MathDrama/reading/Wigner.html`.

[1] J. Kemeny, "Man viewed as a machine," *Scientific American,* April 1955, pp. 58–67.

[2] J. von Neumann, *Theory of Self-Reproducing Automata,* University of Illinois Press, Urbana, 1967.

[3] M. Ridley, *Francis Crick,* Eminent Lives, New York, 2006.

[4] G. Chaitin, *Meta Math!,* Pantheon Books, New York, 2005.

[5] G. Chaitin, "The limits of reason," *Scientific American,* March 2006, pp. 74–81.

[6] G. Chaitin, "Epistemology as information theory: from Leibniz to Ω," European Computing and Philosophy Conference, Västeraas, Sweden, June 2005.

[7] G. Chaitin, "To a mathematical definition of 'life'," *ACM SICACT News,* January 1970, pp. 12–18.

[8] G. Chaitin, "Toward a mathematical definition of 'life'," R. Levine, M. Tribus, *The Maximum Entropy Formalism,* MIT Press, 1979, pp. 477–498.

[3] An earlier account of von Neumann's thinking on this subject was published in [1], which I read as a child.

[9] G. Chaitin, "Algorithmic information and evolution," O. Solbrig, G. Nicolis, *Perspectives on Biological Complexity,* IUBS Press, 1991, pp. 51–60.

[10] G. Chaitin, "Complexity and biology," *New Scientist,* 5 October 1991, p. 52.

[11] G. Chaitin, "Meta-mathematics and the foundations of mathematics," *Bulletin of the European Association for Theoretical Computer Science,* June 2002, pp. 167–179.

[12] D. Lenat, "Automated theory formation in mathematics," pp. 833–842 in volume 2 of R. Reddy, *Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, MA, August 1977,* William Kaufmann, 1977.