



Ubiquity Symposium

Evolutionary Computation and the Processes of Life

Some Computational Aspects of Essential Properties of Evolution and Life

by Hector Zenil and James A.R. Marshall

Editor's Introduction

While evolution has inspired algorithmic methods of heuristic optimization, little has been done in the way of using concepts of computation to advance our understanding of salient aspects of biological phenomena. The authors argue under reasonable assumptions, interesting conclusions can be drawn that are of relevance to behavioral evolution. The authors will focus on two important features of life—robustness and fitness—which, they will argue, are related to algorithmic probability and to the thermodynamics of computation, disciplines that may be capable of modeling key features of living organisms, and which can be used in formulating new algorithms of evolutionary computation.

Editor



Ubiquity Symposium

Evolutionary Computation and the Processes of Life

Some Computational Aspects of Essential Properties of Evolution and Life

by Hector Zenil and James A.R. Marshall

“It is raining DNA outside. On the bank of the Oxford canal at the bottom of my garden is a large willow tree, and it is pumping downy seeds into the air... It is raining instructions out there; it’s raining programs; it’s raining tree-growing, fluff-spreading, algorithms. That is not a metaphor, it is the plain truth. It couldn’t be any plainer if it were raining floppy disks.”—Richard Dawkins¹

With the work of Darwin it became clear that there were natural processes that shaped features and functions in biological organisms. Evolutionary computation has been inspired by this idea, prompting researchers to posit a correspondence between algorithms and evolution.

In her contribution to the Ubiquity symposium on the question “What is computation?” Melanie Mitchell [1] pondered the possibility that biological computation was a process that actually occurred in nature, and computing would eventually prove as fundamental for biology as physics has been for chemistry.

Insofar as evolutionary computation models important aspects of evolution, it does so because evolution itself behaves very much like a computational process. If biological evolution is taken to be computational in nature, we can expect it to be driven by the same forces and to be constrained by the same laws that govern information processing.

Gregory Chaitin, one of the founders of algorithmic information theory [2] (together with A. Kolmogorov [3], R. Solomonoff [4] and L. Levin [5], among others), recently suggested that:

¹ Dawkins, R. The Blind Watchmaker. W.W. Norton & Company, New York, 1986.

“DNA is essentially a programming language that computes the organism and its functioning; hence the relevance of the theory of computation for biology” [6, 7].

Indeed, a central element in living systems turns out to be digital: DNA sequences, refined by evolution, encode the components and drive the development of living organisms. All examples of life we know have the same (genomic) information-based biology. Information, in living beings, is maintained one-dimensionally through the double-stranded polymer. Each polymer strand in the DNA contains exactly the same information, coded in the form of a sequence of four different pairs of bases. In attempting to deepen our understanding of life, it is therefore natural to turn to computer science, where the concepts of information, data structures, and algorithms are investigated.

Important concepts in the theory of computation can help us understand aspects of behavior and evolution, in particular concepts drawn from algorithmic complexity and computational thermodynamics. Questions such as which genes encode which biological functions, and what shape a protein will fold into, given a DNA sequence, may be effectively formalized in a computational context. We will focus here on two pivotal features of life—robustness and fitness—which, we will argue, can be related to basic properties of computing systems.

Biology and Computational Universality

Evolution manifests some fundamental properties of computation, and not only does it resemble an algorithmic process, it often seems to produce the kind of output a computation would be expected to produce [8]. Witness the fact that the instructions for life are stored in sequences of DNA in identifiable functional units of information even if full of intricate paths and complicated connections with unpredictable outcomes.

Computer simulations performed as part of research into artificial life have reproduced various features of evolutionary behavior [9, 10, 11], all of which have turned out to be deeply connected to the concept of (Turing) universal computation [12]. In Margenstern’s 2012 lecture [13], an encoding of a very small computer program capable of universal computation was presented as an example of a computing system that, written in the letters of the DNA, is smaller in length than any currently known virus yet it is a general-purpose computer that may, in principle, emulate the behavior of any possible virus. This goes to suggest it takes very little to attain Turing universality, which is not in any fashion a special feature of an abstract system. Moreover, it suggests viruses may potentially be capable of any possible unexpected threat to

any possible solution if despite their size they have the computational power of Turing universality, because they can self-assemble in theoretically any possible configuration (and here we are a level below possible physical constraints), unfamiliar to all previous or current antibodies. Not taking into account this phenomenon of pervasive computation in biology; treating it as a mere technicality with little relevance and consequently avoiding it, is a mistake.

Our knowledge of life may be advanced from studying notions at the edge of decidability and uncomputability (e.g. as we define it, algorithmic probability is a non-computable measure, but its importance for us lies in the fact that it can be approximated). The concept of Turing universality (and of a universal Turing machine) should simply be treated as a physical system whose richness allows us to study a low level of basic systems behavior without having to worry about particular causes for particular behaviors. Its introduction as a tool shouldn't alienate researchers, leading them to treat it as an abstract concept with no practical relevance to biology. Some artificial self-assembly models [14] inspired by biological systems, for example, are capable of yielding arbitrary shapes [15] because they turn out to have the property of Turing universality. Our current understanding of self-assembly wouldn't be possible without the concept of computation universality. These concepts have been paired since the beginning with the work of John von Neumann and self-replication [17].

Two of the founders of the field of computer science had early interests in biology. Alan Turing was interested in the question of pattern formation in biology that he named *morphogenesis* [16], while John von Neumann was interested in the problem of *self-replication* [17] (later generalized to self-assembly [18]), laying the foundations of what is today the theory of cellular automata and of pattern formation. Von Neumann sought to model one of the most basic life processes—reproduction—by designing lattice-based rules where space is updated altogether in discrete steps for studying self-replication. Self-replication was not only possible but was indeed a direct consequence of Turing's computation universality even if self-replication can be achieved with much less power than that of Turing universality.

Surprisingly, an extremely simple computing system such as Conway's Game of Life [9], a two-dimensional cellular automaton can capture several features that we customarily associate with life (hence its name). At the same time it possesses the most important property of computation, namely Turing universality, thus serving as a strong bridge between these two important scientific disciplines. Today we know that universal systems such as Conway's Game of Life are by no means an exception, often built from a very few, simple components. Many examples have been found (e.g. Langton's ant (a Turing machine) and Wolfram's cellular

automaton rule 110, to mention but 2 examples) or designed. Recently, other respected authors have echoed the position that universality is a ubiquitous natural phenomenon (see [13, 19]). Wolfram’s Principle of Computational Equivalence [11] suggests this ubiquity of computation universality, claiming that *almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication*.

Among the key properties of living organisms is their robustness in the face of genetic change [20]. We propose another important connection to Turing universality offering a low level explanation of biological robustness. We think the properties of algorithmic probability provide a sound computational framework for explaining robustness without leading to any apparent contradiction (such as the one between robustness and evolvability).

Algorithmic Probability and Biological Robustness

There is a measure [4, 5] that describes the probability of a universal Turing machine producing a string s when running a computer program produced at random. In the form of an equation this measure can be written as follows,

$$Pr(s) = \sum_{p:M(p)=s} 2^{-|p|} \quad (1)$$

That is the sum over all the programs p producing the string s running on a (prefix-free²) universal Turing machine M .

Levin shows [5] there is a universal (semi³) measure m that dominates any other probability measure Pr . Hence we can work with m instead, which thanks to the algorithmic Coding theorem [21], can be written as $m(s) \sim 1/2^{K(s)}$, where $K(s)$ is the Kolmogorov-Chaitin complexity of s , that is the length of the shortest computer program producing s . Therefore, m connects the frequency of a pattern to its algorithmic (Kolmogorov-Chaitin) complexity (for a more detailed technical explanation see [22]). Intuitively, m tells that patterned objects are more frequent than random-looking when produced by random computer programs.

In order to interpret this computational notion in the context of biology, let’s think of a *gedankenexperiment*, a container for each of the four nucleotides of DNA: A (for adenine), C

² That is, a machine for which a valid program is never the beginning of any other program, a technicality that allows to define a probability over strings produced by programs for which the sum should be at most one.

³ “Semi” because it is semi-computable, approachable from below.

(for cytosine), G (for guanine), and T (for thymine). Connect the container to a “biological” machine, and feed it with a random computer program with instructions constructed from (uniformly distributed) random choices of nucleotides from the containers, forming a syntactically valid sequence s of DNA. The machine will then produce an output equivalent to a protein encoded by the “DNA program” precisely according to the distribution of patterns dictated by $m(s)$. Each random program written in the language of the DNA will produce a protein with probability $m(s)$. Should we expect to see the distribution of patterns predicted by $m(s)$? Biological machinery has been exposed to another very powerful mechanism: natural selection. Nevertheless, biological self-assembly may have started purely computational from mining random computer programs (as suggested in [11]) and $m(s)$ may play an important role in the driving forces constructing these sequences and for understanding the building power of digital biological sequences, even if they are selected upon proving to effectively serve a particular use given that natural selection will favor sequences encoding an evolutionary advantage.

Having learned directly from James D. Watson (the co-discoverer of the structure of DNA), Charles Bennett realized a beautiful analogy which casts RNA polymerase as a specific purpose Turing machine [23], what amounts to a “truly chemical Turing machine,” an enzyme that crawls along a sequence of instructions analogous to a machine “tape” transcribing DNA, stepping left and right just as a Turing machine would (this is not a minor thing; it is the possibility of accessing any part of a Turing machine tape that, alongside other simple factors, contributes to the computational power of a universal Turing machine). The logical state of the RNA polymerase changes according to the chemical information written in the sequence, and Bennett actually used this information to calculate the energy required for this biological process [23].

Not all biological processes may behave like or necessarily correspond to the exact working mechanism of a Turing machine (in fact this would be an unacceptable assumption), and the concept of computational universality is being introduced only so as to facilitate the consideration of all possible cases (since a machine that can produce any possible sequence and is not constrained to be programmed in any particular way can cover all possible—computable—arrangements).

It is perhaps not a coincidence that the sequence that the ribosome sequentially “reads”, acting on the directions it provides, is called the “control tape.” This is also very similar to B cells, vital in the human immune system, producing a huge variety of antibodies by mixing and matching

DNA segments to create antigen-binding sites, picking random combinations of gene segments from many options available from a genome pool [24]. After all, the instructions a ribosome follows to compile a protein inside a cell provided by the mRNA are not very different to the operation of an actual computing machine (see [25]).

Trying to feed a machine with random sequences of nucleotides will certainly prove to be a frustrating experience if undertaken as a source of synthetic organisms. But for those outcomes that are biologically meaningful, algorithmic probability would describe a frequency of distribution of patterns in terms of their algorithmic complexity. Of course, not all programs will produce a biologically viable output. Most will be disappointing, if they produce anything at all, just like Turing machines that may or may not stop, or stop after only one step. These can justifiably be considered to have no relevance to the purpose at hand. Since three DNA bases encode each amino acid, the addition or lack of only one or two bases, for example, would cause a shift in the “reading process,” making it meaningless. This does not interfere with the claim that $m(s)$ can impact the way the encoding and decoding process takes place in biological systems. Even if one thinks of it as marginal, the fact that one can make any prediction about the distribution of protein folding complexity based on its frequency of occurrence is a reasonable low-level explanatory approach to this biological phenomenon, and is consistent with what we have found in natural and artificial systems [26,27].

In biology, robustness manifests as the ability of an organism to preserve its phenotype in spite of random mutations or transitional environmental changes. A system is robust if its behavior remains qualitatively unchanged in the face of sudden random perturbations. The property of being able to produce persistent structures from randomness is essential to our algorithmic model as applied to biology. Start with a random-looking string and run a randomly chosen program on it, there is a good chance that a high probability random-looking string will be turned into a regular, highly structured one (with low Kolmogorov–algorithmic complexity). The programs producing these structured strings may be the kinds of building blocks used by organisms and are thought to be common to living systems as basic constituent units [28]. For example, in an experiment with digital organisms designed to show how complex features originate from random mutation and natural selection, it was found that most evolutionary paths build over prior intermediate states to arrive at the same evolved complex functions [29]. From the stable frequency of the outputs described by $m(s)$, one can see how distinct complex structures can arise simply from running “random” programs, quickly amounting to the same solutions. Yet living systems evolve and are not immune to changes induced by their environments—changes that ultimately favor certain mutations. Chaitin has pointed out [6]

something that seems compatible with both the notion of building blocks of life and the notion of biological modularity from the point of view of computer programming languages [30], viz. that mutations are likely to happen at the level of the program. This means the system somehow remains consistent at the level of the specification even if modules change over time. The mutation is therefore not exactly random at the point level, which in terms of our algorithmic model, remains stable, since most programs produce the same structured output as opposed to a uniform distribution, where any change would yield a noticeable outcome and greater fragility (the assumption behind this argument is it is the structured output that matters for robustness, and not the random-looking output, which appears more fragile in the distribution of patterns according to $m(s)$).

For example, one answer to the question of convergent evolution (same solutions to different environments) suggested by this approach is that there are far fewer possible programs producing different outputs and program variations (subroutine changes) than there are possible simple combinatorial solutions, hence reducing the solution space from a possibly uniform distribution to one according to $m(s)$ (a power law distribution) that favors, in frequency and complexity, simple structured outputs.

Bio-computational Thermodynamics and Fitness

As we mentioned before, Bennett had already made a beautiful connection between universality, computing machines, randomness and biological thermodynamics. If biological systems are regarded as computational systems one may inquire into biological systems as one would into computational physics. Organisms use information about their environment to determine the value of different environmental parameters [31]. For example, according to Galef, learning is a set of complex ontogenetic processes that allows animals to acquire, store, and subsequently use information about the environment [32].

To grasp the role of information in biological systems think of a computer as an idealized information-processing system. Today it is fairly easy to understand, from a practical point of view, how energy may be converted into information [23, 33, 34]. Computers may be thought of as engines transforming energy into information (the information may already be there, but without energy one is unable to extract it). One need only connect one's computer to the electrical grid in order to have it perform a task and produce or reveal information.

This works in reverse as well, that is, information can be converted into energy, as has been studied in the thermodynamics of computation. A good way to understand the process is by using the well-known *gedankenexperiment* known as the Maxwell demon paradox [23, 35]. The basic idea is that one can use knowledge about the microstate of a system to make it hotter. Using bits to produce energy is well explained by Feynman and Sethna [34, 36]. The connections to information theory and algorithmic complexity, however, are less well understood, but we have a good sense of how we might use algorithmic information theory to connect these concepts at a deep level.

It is clear animals convert information into energy—for example, by using information to locate food, to navigate within a habitat, or by learning how to hunt. Similarly, it is clear energy can be used to extract even greater quantities of energy from the environment—through investment in the construction and operation of a brain. From this it follows the extent to which an organism has adapted to or is able to produce offspring in a particular environment depends on a positive exchange ratio between information and energy.

Information processing in organisms should be understood as the process whereby the organism compares its current knowledge about the world with the observable state of the world at a given time. In other words, an animal weighs every possible future outcome against its current representation. While the larger the number and the more accurate the processed observables from the environment the better the decisions and predictions, organisms cannot spend more than the total return received from the use of that information [37].

In the context of living organisms, a recent paper has shown, under certain assumptions, information processing by individuals can only be a fitness enhancing property [38]. But from computational thermodynamics it follows that any system with access to a finite amount of resources (e.g. memory) has to incur a cost, assuming nothing other than information processing.

Exchanging information for energy and energy for information some energy inevitably escapes into the environment. Dissipation is a general phenomenon in the real world and it tells us that something is lost in the exchange process, something which itself interacts with the environment, affecting other organisms. In accordance with the second law of thermodynamics, one can see this as information about the system's irreversibility. In biology this kind of exchange happens all the time. Neurons, for example, dissipate about $10^{11} kT$ per discharge [23]. Computers (mainly because of their volatile memory devices—the RAM memory)

also dissipate energy by at least $1kT$ joules per bit [17, 23, 34] (where k is the Boltzmann constant and T the ambient temperature)—which is the reason computers heat up and require an internal fan. Today, living cells are commonly represented in systems biology (even in textbooks, e.g. [39]) as information processing machines, with *E. Coli*, given as an example, transcribing 5×10^6 genes during a half hour, which is about 10 Gb/h of information. Of course this dissipation is negligible compared to the dissipation of human activity in the transformation and use of energy, but its source is of the same thermodynamic nature; it just belongs to a very different level.

Hence, if an organism aims to gather information about the world, updating its previous state by replacing old information with new (learning), a cost is unavoidable and takes the form of dissipation. The connection to fitness is then quite natural. Living systems can be regarded as systems with access to finite resources and subject to the same costs as any other physical system ultimately constrained to this rather computational limit [37].

It is obvious natural selection has equipped organisms to cope with such trade-offs, to decide whether to undertake certain actions in return for spending resources. This happens, for example, with some animals, which spend most of their energy in the very first instants of a hunt. Because they are not able to keep up the speed of pursuit for long periods of time, they have to ponder and hit upon the optimal starting point for a chase. So information (e.g. cues indicating the location of the prey) can be a fitness enhancing property if the energy return is greater than the cost of processing it and taking action.

Szilárd established a connection between energy and information at a very low level [40]. The connection is information can be exchanged for energy because it can help to extract energy. Gaining information lowers the entropy (uncertainty) of a system. The use of a fundamental unit of information to produce energy is well explained in Feynman [34] and Sethna [36]. Szilárd showed if one had one bit of information about a system, one could use that information to extract an amount of energy given by $W = kT \log 2$. Szilárd's result in fact implies the Second Law of Thermodynamics and can be written as follows [41]:

$$W_{in} \geq kT \log(2) \geq W_{out} \quad (2)$$

In other words, one can extract at most $kT \log 2$ of work from a bit, and it costs at least $kT \log 2$ to erase (reset) a bit, with M the number of bits to erase (or reset). Landauer studied this thermodynamical argument [34] and proposed a principle: If a physical system performs a logically irreversible classical computation, then it must increase the entropy of the

environment with an absolute minimum of heat release of W per lost bit. Landauer's principle has recently been tested in the laboratory [43].

As pointed out by de Vladar and Barton, natural selection can be seen as extracting information from the environment and coding it into a DNA sequence [43]. But current efforts in the direction of quantifying information content typically do not venture beyond Shannon's communication theory. Nevertheless, computational thermodynamics can help us understand how selection entails the accumulation and exchange of information and energy with the environment in the replacement of populations, as it has already helped to connect computation and physics by Szilárd, Landauer, Bennett and Feynman among others.

From the thermodynamics of computation it can be formally concluded that learning is an energy-saving strategy while "forgetting" takes work and consumes information- processing energy (as proven by Landauer and Bennett, this cost is more fundamental, although perhaps more marginal than others). Organisms, therefore, will make an effort to learn fast and keep a structured representation of the world. It also tells us they wouldn't survive in a random environment, without storing information resources or coping with the computational power of the environment [37]. This means some predictability is necessary for their survival, and the existence of living systems constitutes a thermodynamical proof of the algorithmic structure of nature [37], and therefore connecting the two aspects that we are covering in this manuscript.

Concluding Remarks

As a complement to evolutionary computation, and indeed in recompense for what biology has bequeathed to computation in the form of nature-inspired models, we ought to acknowledge that such an exchange from the theory of computation to evolutionary biology is possible. Not only because of the existence of strong similarities, but also because computation may explain fundamental aspects of life in the form of abstract models, as in and of itself it may explain pivotal aspects of biological processes and biological functions. And the purpose of abstraction, as Dijkstra would have said, is not to be vague, but to create a semantic level in which one can be absolutely precise, and certainly make process in the formalization of some of these aspects.

We have yet to see how these concepts may also shape new evolutionary computational algorithms, refining techniques and perhaps improving our grasp and application of them. In this regard it is important to point out that work using algorithmic probability for optimal search algorithms in artificial intelligence has been developed [45], but has not, to the authors' knowledge, been used to complement current approaches to evolutionary computation. So it

would be interesting to incorporate this knowledge and trespass the barriers separating the disciplines of computation and biology at a level not attempted before (moving from computation to biology that is, since there has been movement in the other direction, which has greatly benefitted evolutionary algorithms).

Acknowledgements

We wish to thank the editors of this *Ubiquity* Symposium for their invitation, as well as for their comments, which helped improve this presentation. H. Zenil also wishes to thank the Foundational Questions Institute (FQXi) for collaboration support under mini-grant number FQXi-MGA-1212 “Connections of Computation and Biology”, and the Silicon Valley Community Foundation (for mini-grant number 2012-96393 (4661)).

References

- [1] Mitchell, M. Ubiquity Symposium: What is Computation. Biological Computation *Ubiquity* February 2011.
- [2] Chaitin, G.J. On the Length of Programs for Computing Finite Binary Sequences: Statistical considerations, *Journal of the ACM*, 16, 1 (1969), 145–159.
- [3] Kolmogorov, A.N. Three Approaches to the Quantitative Definition Of Information. *Problems of Information and Transmission* 1, 1 (1965), 1–7.
- [4] Solomonoff, R. A Preliminary Report on a General Theory of Inductive Inference. (Revision of Report V-131), Contract AF 49(639)-376, *Report ZTB –138*, Zator Co., Cambridge, Mass., Nov, 1960.
- [5] Levin, L. Laws of Information Conservation (non-growth) and Aspects of the Foundation of Probability Theory. *Problems in Form. Transmission* 10 (1974), 206–210.
- [6] Chaitin, G.J. Metaphysics, Metamathematics and Metabiology. In H. Zenil (ed.), *Randomness Through Computation*. World Scientific, 2011, 93–103.
- [7] Chaitin, G.J. Life as evolving software. In H. Zenil (ed.), *A Computable Universe*. World Scientific, 2012.

- [8] Li, H., Helling, R., Tang, C., and Wingreen, N. Emergence of Preferred Structures in a Simple Model of Protein Folding. *Science*. 273, 5275 (1996), 666–669.
- [9] Gardner, M. Mathematical Games—The fantastic combinations of John Conway’s new solitaire game “life.” *Scientific American* 223 (1970), 120–123.
- [10] Langton, C.G. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena* 22, 1-3 (1986), 120–149.
- [11] Wolfram, S. *A New Kind of Science*. Wolfram Media, 2002.
- [12] Turing, A.M. On Computable Numbers, with an Application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society*. 2, 42 (1936–37), 230–65.
- [13] Margenstern, M. Universality everywhere and beyond, an epic of computer science. Invited Lecture. *Turing in Context II, Historical and Contemporary Research in Logic, Computing Machinery and AI*, 10-12 October 2012, Royal Flemish Academy of Belgium for Science and the Arts, Brussels, Belgium.
- [14] Rothemund, P.W.K., Papadakis, N., and Winfree, E. Algorithmic Self-Assembly of DNA Sierpinski Triangles. *PLoS Biol* 2, 12 (2004). <http://www.plosbiology.org/article/info:doi/10.1371/journal.pbio.0020424>
- [15] Winfree, E. Simulations of Computing by Self-assembly. *Technical Report CS-TR: 1998.22*. Caltech, 1998.
- [16] Turing, A.M. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London*. B 14. 1952.
- [17] von Neumann, J. In A. Burks (ed). *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.
- [18] Winfree, E. *Algorithmic Self-Assembly of DNA Thesis*. In partial fulfillment of the requirements for the degree of Doctor of Philosophy, Caltech, 1998.

- [19] Davis, M. Universality is Ubiquitous. Invited Lecture. *History and Philosophy of Computing (HAPOC11)*, Ghent, November 8, 2011.
- [20] Wagner, A. *Robustness and Evolvability in Living Systems*. Princeton University Press, 2005.
- [21] Cover, T.M. and Thomas, J.A. *Elements of Information Theory*. John Wiley and Sons, 2006.
- [22] Zenil, H. and Delahaye, J-P. On the Algorithmic Nature of the World. In G. Dodig-Crnkovic and M. Burgin (eds), *Information and Computation*. World Scientific Publishing Company, 2010.
- [23] Bennett, C.H. The Thermodynamics of Computation—A Review. *International Journal of Theoretical Physics* 31, 12 (1982), 905–940.
- [24] Allman, D., Srivastava, B., and Lindsley, R.C. Alternative Routes to Maturity: Branch points and pathways for generating follicular and marginal zone B cells. *Immunol. Rev.*, 197 (2004), 147–60.
- [25] Steitz, T. A. A Structural Understanding of the Dynamic Ribosome Machine. *Nature Reviews Molecular Cell Biology* 9 (2008), 242–253.
- [26] Dingle, K., Zenil, H., Marshall, J.A.R., and Louis, A.A. Simplicity Bias in Genotype-Phenotype Maps. In preparation.
- [27] Zenil, H. Turing Patterns with Turing Machines: Emergence and Structure-Formation from Universal Computation. *Natural Computing*, 10.1007/s11047-013-9363-z.
- [28] McGinnis, W., Levine, M.S., Hafen, E., Kuroiwa, A., and Gehring, W.J. A conserved DNA Sequence in Homoeotic Genes of the *Drosophila* Antennapedia and Bithorax Complexes. *Nature* 308, 5958 (1984), 428–33.
- [29] Lenski, R., Ofria, C., Pennock, R.T., and Adami, C. The Evolutionary Origin of Complex Features. *Nature* 423, 139–144, 2003.
- [30] Wagner, G.P. Homologues, Natural Kinds and the Evolution of Modularity. *Integrative and Comparative Biology* 36 (1996), 36–43.

- [31] Valone, T.J. Are Animals Capable of Bayesian Updating? An Empirical Review. *Oikos*, 2006.
- [32] Galef, B.G. Jr. Social Learning in Animals: Empirical studies and theoretical models. *Bioscience*. 2005.
- [33] Landauer, R. Irreversibility and Heat Generation in the Computing Process. *IBM Journal of Research and Development* 5 (1961), 183–191.
- [34] Feynman, R.P. *Feynman Lectures on Computation*. Westview Press, 2000.
- [35] Bennett, C.H., Gács, P., Li, M., Vitányi, P.M. B., and Zurek, W.H. Thermodynamics of Computation and Information Distance. In *STOC '93 Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*. 1993.
- [36] Sethna, J. *Statistical Mechanics: Entropy, Order Parameters and Complexity*. Oxford University Press, 2006.
- [37] Zenil, H., Gershenson, C., Marshall, J.A.R. and Rosenblueth, D., Life as Thermodynamic Evidence of Algorithmic Structure in Natural Environments. *Entropy* 14, 11 (2012), 2173–2191.
- [38] McNamara, J.M. and Dall, S.R.X. Information is a Fitness Enhancing Resource. *Oikos*. 119 (2010), 231–236.
- [39] Sneppen, K., and Zocchi, G. *Physics in Molecular Biology*. Cambridge University Press, 2005.
- [40] Szilárd, L. *Über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen* (On the reduction of entropy in a thermodynamic system by the interference of intelligent beings). *Z. Physik* 53 (1929), 840–856.
- [41] van den Broeck, C. Thermodynamics of information: Bits for less or more for bits? *Nature Physics* 6, 12 (2011), 937–938.

- [42] Toyabe, S., Sagawa, T., Ueda, M., Muneyuki E., and Sano, M. Experimental Demonstration of Information-to-Energy Conversion and Validation of the Generalized Jarzynski Equality. *Nature Physics* 6 (2010).
- [43] de Vladar, H.P. and Barton, N.H. The Contribution Of Statistical Physics to Evolutionary Biology. *Trends in Ecology & Evolution*. 26, 8 (2011), 424–432.
- [44] Zenil, H. Information Theory and Computational Thermodynamics: Lessons for Biology from Physics. *Information* 3, 4 (2012), 739-750.
- [45] Hutter, M. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*, (1st ed. 2005 edition), Springer, 2010.

About the Authors

Hector Zenil is a Research Associate in the Department of Computer Science, University of Sheffield and Head of the Algorithmic Nature Group. He is a member of the Turing Centenary Advisory Committee and the editor of *Randomness Through Computation and A Computable Universe*. (<http://www.mathrix.org/zenil/>)

James A. R. Marshall is a Reader in Computational Systems Biology in the Department of Computer Science, University of Sheffield. He is also a member of the interdisciplinary Kroto Research Institute, where he leads the behavioral and evolutionary theory research theme. (<http://staffwww.dcs.shef.ac.uk/people/J.Marshall/lab/AboutUs.html>)

DOI: 10.1145/2480352.2480353