Biological Hypercomputation: A New Research Problem in Complexity Theory

This article discusses the meaning and scope of biological hypercomputation (BH) that is to be considered as new research problem within the sciences of complexity. The framework here is computational, setting out that life is not a standard Turing Machine. Living systems, we claim, hypercompute, and we aim at understanding life not by what it is, but rather by what it does. The distinction is made between classical and nonclassical hypercomputation. We argue that living processes are nonclassical hypercomputation. BH implies then new computational models. Finally, we sketch out the possibilities, stances, and reach of BH. © 2014 Wiley Periodicals, Inc. Complexity 000: 00–00, 2014

Key Words: complex systems; biological information processing; nonclassical hypercomputation; theoretical biology; complexification of computation

CARLOS E. MALDONADO[1]
NELSON A. GÓMEZ CRUZ[2]

[1]School of Political Science and Government, Universidad del Rosario, Bogotá, Colombia (E-mail: carlos.maldonado@urosario.edu.co) and [2]Modeling and Simulation Laboratory, School of Management, Universidad del Rosario, Bogotá, Colombia

1. INTRODUCTION

Thanks to the development of computation the sciences of complexity were born [1] but the sciences of complexity have also contributed to the development of computation. The modeling and simulation of such processes and phenomena as nonlinearity, adaptation, information and communication, self-organization, and emergence—to name but just a few—made possible to better grasp and see characteristics such as learning, local and global interaction, birth and death, and complex networks, for instance. The workings went further to pervade biology at large to-date.

However, theoretical computer science has remained within the framework of classical science, namely the Turing machine (TM) and the (strong) Church–Turing thesis (CTt) [2]. Certainly, other alternatives have been considered, but engineering on-the-field in general and computer engineering in particular, remain all in all within the mainstream of the CTt and the TM.

The TM models no more and no less than algorithmic computation. It is clear that classic algorithmic computation is closed—the world is shut out—, its resources are finite, and the behavior (modeled) is fixed (conspicuously top-down) [3]. As a consequence, the TM does not represent a complete model of computation [4,5].

The computability (expressiveness and power) problem concerning most of the new paradigms and models inspired by biology have been reduced to the possibility

© 2014 Wiley Periodicals, Inc., Vol. 00 No. 00
DOI 10.1002/cplx.21535
Published online 00 Month 2014 in Wiley Online Library
(wileyonlinelibrary.com)

that they can equate the computational power of the universal TM, for they assume, from the outset, that in the Turing model there are limitations about what can and cannot be computed. Among the workings that can be placed in this direction, we encounter some models of membrane computing [6,7]; one-dimensional cellular automata (rule 110) [8] and two-dimensional cellular automata (the game of life, Conway) [9]; insertion–deletion systems [10]; ciliate computing [11]; peptide computing [12]; and some DNA computing models [13]. In other words, these can be taken as biological metaphors applied to models of computation.

Proving the Turing-completeness of the new computational paradigms—and particularly in those inspired by biology—has become a matter of routine and yet, according to Syropoulos [14], the computational TM-based paradigm "cannot and should not form the basis for proposing and studying new models of computation inspired by natural processes" (the same idea can be found in [5]).

Biological computation [15], moreover, is a field that has rarely been clearly and deeply worked out, especially within the sciences of complexity. The subject regarding biological computation has been, so far, about the computational models that can be adapted or developed to understand biological processes. However, biology is not to be taken exclusively as a metaphor for the aims of computation. On the contrary, the real and defiant task concerns the understanding of what living systems do to live. This, we claim leads us onto biological hypercomputation (BH), an argument that other authors would easily disclaim [16].

BH can be grasped as the explanation of computational processing the biological processes and dynamics ranging from the lowest scales, that is, genes and bacteria up to the biosphere. Hence, it encompasses both the developmental as well as the evolutionary stances in life. In this article, we introduce the concept of BH throughout four arguments, thus: (a) biological systems are not machines, and in which sense they are not; (b) evidences in the literature about the plausibility of the concept are brought out. As a consequence, the distinction between classical and nonclassical hypercomputation must be made; (c) the very concept is introduced and discussed. Although we bring out some experimental evidence shown in the literature, we also assess that radically different computational models must be conceived; and (d) the reach, stances, and possibilities of the concept introduced in this article are discussed. It is our contention that for the sciences of complexity the concept of BH helps magnificently to better understand what life is and what the biological processes are all about. At the end several conclusions are drawn.

2. LIFE IS NOT A (TURING) MACHINE

Modern science conceives of life—the human body, living systems, and even nature–as a machine very much along the framework of classical mechanics. For modern science a machine is a device clearly defined in anatomical and physiological terms. In the first decades of the 20th century machines were understood somehow according to the TM and hence as a computational device.

Within the computational framework the best characterization of a machine is the TM. Yet, life is not a TM, whence new models of computation different from the TM must be developed or introduced. These new models of computation are addressed to understanding what life does to live, and not to just to solve computational problems in the traditional sense. As it is well known, a TM is an algorithmic device, that is, machine that can solve P problems in terms of Cook and Karp's computational complexity.

Living systems develop and evolve. However, neither biological development nor evolution is performed in classical algorithmic terms. Moreover, the environment does not carry out processes and dynamics algorithmically. Evolution is not to be taken in classical Darwinian terms—which, by the way, might indeed be grasped as an algorithmic process—but rather, according to Jablonka and Lamb [17] as the entire process that encompasses the genetic, the epigenetic, the behavioral, and the symbolic levels.

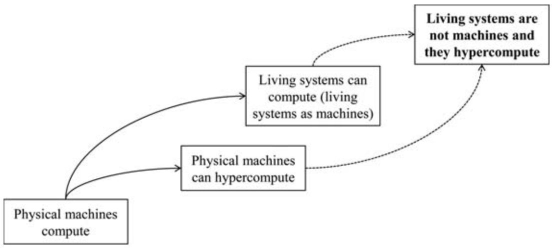
Living systems do compute though not in the sense of the classical TM. A tentative analog here could be the two sorts of machines Turing himself suggested, namely an automatic machine in respect to the mathematical Hilbert's tenth problem; additionally, a thinking machine, by which he was considering logical and philosophical contexts, that is, a machine that could not be reduced to mere algorithmic processes. There are, indeed, other models of computation that cannot be taken as mere algorithmic proposed by Turing most notably the oracle machine (o-machine), and the choice machine (c-machine). These machines do not compute: they hypercompute, this is, they interact with the environment, and make decisions on their own—provided an external operator (human, oracle).

Computation is for biological systems more than a matter of one tape that runs sequentially. It is matter that entails metabolizing, cellular, and homeostatic processes whence health and disease; after all, computation entails for the living systems the edge between life and death. Figure 1 shows the (evolutionary and computational) steps from physical machines to living systems.

However, it should be clearly pointed out that living systems are physical entities and yet cannot be reduced to sheer physics. This is where, according to Jablonka and Lamb [17],

DOI 10.1002/cplx

FIGURE 1


Living systems are not machines and they hypercompute
Living systems can compute (living systems as machines)
Physical machines can hypercompute
Physical machines compute

Living systems are not machines and they hypercompute.

the behavioral and symbolic stances are introduced in evolution. The behavioral stance means all capacities to adapt for a given individual whereas the symbolic ones are the outcome of cultural interactions. Hence, "physics" will not be understood as "material" which allows us to encompass "information" as well. Information is a physical entity, indeed, though not "material" (see for arguments [18,19]). If so, living systems process information as not just material stuff but as knowledge synthesis. We will come back to this.

3. HYPERCOMPUTATION: FROM CLASSICAL TO NONCLASSICAL

Hypercomputation is concerned with behaviors and phenomena that lie outside the capacities and interests of the TM. However, hypercomputation can be traced back to the workings by Turing [20]—particularly the c-machines [21] and o-machines [22,23], but remained abandoned or in oblivion for a long time. Isolated works were undertaken in-between, until in 1999 Copeland and Proudfoot [24] coined up the term hypercomputation (other synonymous concepts are super-Turing computation [25] and super-recursive algorithms [26]).

The theory of hypercomputation refers to the theoretical and practical feasibility of computing numbers and functions that the TM is unable to compute [20]. However, hypercomputation not only deals with the computation of numbers and functions that cannot be computed by a TM, but, more adequately and generically, it looks for finding new ways for solving problems and computing things [14]. Thus, in other words, hypercomputation widens up the framework of what is computable within the context of the CTt while trying to generalize the concept of computation to include dynamics, phenomena, and processes that were never before taken into account as essentially computable.

In this sense, according to Stepney [27], the models of hypercomputation that seek to solve the halting problem or to compute Turing-uncomputable numbers can be gathered under the generic name of classical hypercomputation (CH). Those models of hypercomputation that try to move beyond challenging the traditional notions of computation and the pillars established by the theory of computation are known as nonclassical hypercomputation (nCH).

CH intends transcend the limitations of the TM, wherein the basis is classical physics, by taking advantage of quantum theory and the general and special theory of relativity [27]. This form of hypercomputation implies the possibility of carrying out super-tasks, that is, for instance, carrying out an infinite number of operations in a finite time—and this would presumably allow for solving in principle Turing's halting problem [14]. In spite of the fact that all models of hypercomputation that go in this direction remain on the theoretical plane, there is nothing, so far, in the known mathematics and physics that impedes the implementation of such hypermachines [28]. Yet, CH remains within the boundaries imposed by the TM model.

As for nCH, it is concerned with widening up the limits of what classically has been understood by "computing" (a similar notion can be found in MacLennan as non-Turing computation [29]). The heuristics Stepney uses for introducing nCH consists in asking several questions, which show the tentative or exploring character of the issues, as well as the innovation of the field. The paradigms pointed out by Stepney as means to reach nCH are related to questions such as the physical embodiment, the use of parallelism as the unique mean to achieve a real time response, the interactive nature of computation [interactive computation (IC)], and the plausibility of some kind of emergent hypercomputation.

Within the paradigms brought out by Stepney, IC is the only one that has been formally constituted and broadly developed in the context of hypercomputation [4,5,14,30,31]. In fact, some of current computing systems such as Internet or sensing and acting robots [4] are better modeled in terms of interaction than in algorithms. This is, whereas the TM allows us to model closed systems, IC models open systems that interact with environments essentially uncomputable. Moreover, a TM does not know of time, whereas IC knows and works with the arrow of time [30]. Thanks to IC time is openly incorporated within computation. Life is the best example of an interactive system. We strongly believe that IC is the only field within hypercomputation that clearly allows opening the door toward BH, so far.

© 2014 Wiley Periodicals, Inc.
DOI 10.1002/cplx

All in all, hypercomputation does not know of biological computation, and vice versa: biological computation does not know of hypercomputation. Paradoxically, they are both stuck with the CTt and the TM's paradigm. To be sure, the openness of computation to complexity produces hypercomputation and, we claim, it is nCH that leads to BH.

4. BIOLOGICAL HYPERCOMPUTATION

Biological computation is a recent research area that studies the information processing that living systems carry out in their natural environment [15]. Within the sciences and disciplines that have contributed to consolidate this field there are natural computing, artificial life, systems biology, synthetic biology, computer science—most notable the nonclassical paradigms—, and the sciences of complexity. Biological computation differs conceptually and theoretically from such strands as computational biology, bioinspired computation, modeling, and computational syntheses of biological processes and the use of biological materials for computational aims. Nonetheless, the frontiers among these strands are currently blurred and hardly identifiable. According to Mitchell [15] "it is only the study of biological computation that asks, if, how, and why living systems can be viewed as fundamentally computational in nature."

In biological computation there are usually three ways to explain biological phenomena as computational concerns. These are: theoretical computing models (in info), descriptions, and experimental observations and simulations. Several examples in the literature about biological computation include bacteria colonies [32,33]; gene assembly in ciliates [34–36]; biochemical reactions [37,38]; gene networks, protein–protein interaction networks, and transport networks [39,40]; cells [41]; information diffusion in the endocrine system; defense adaptation and coordination in the immune system [42,43]; information processing in swarm insects [44]; plants [45]; and evolution as computation [46,47]. A large part of the examples just mentioned assume implicitly or explicitly the TM as a final and complete model of computation [48] and, therefore, take the CTt as the upper bound of what can be computed.

To make the concept of biological computation possible Mitchell [15] asks four questions. They concern how is information represented in the system, how information is read and written by the system, how it is processed, and how this information acquires function (or "purpose" or "meaning"), and sets comparisons between TM-based computation and what life does, computationally speaking. Not far ago, Stepney [27] does include the term "biological hypercomputation" and yet she does not develop nor deepen into the concept. Therefore, it is our contention to justify and explain the concept of BH here.

There are, indeed, a number of theoreticians and researchers who are already working in terms of BH, even though they do not make use of the concept as such. Rosen [49], for instance, brings out the metabolism-repair systems [(M, R)-systems] to prove that life is not a mechanism whatsoever, that his models are nonsimulable, noncomputable (in terms of CTt and TMs), or also nondefinable by a classical recursive algorithm [50]. For Rosen [51] only those systems that have noncomputable, unformalizable models must be considered as complex. The result of such a claim has not been studied by the sciences of complexity to-date. Along the same argument, Kampis [52,53] introduces self-modifying networks/systems to explain the origins and increase of complexity in biological and cognitive systems and he claims that these systems cannot be expressed or explained in terms of a TM. Moreover, he says that the TM can be applied only to simple and complicated systems. Furthermore, Ben-Jacob [32] makes substantial differences between bacterial information processing and the Universal TM (UTM). Whereas in the UTM, the software and hardware are separated, within a bacterial colony the hardware changes according to the input, the information stored, the information processing, and the desirable output. Bacteria can exchange information with each other in the form of hardware (i.e., genetic material). Without having to go any further, this must be taken as an example of interactive biological (hyper)computation.

The most recent example of BH found in the literature is Siegelmann [48]. According to her, the TM computes only those tasks for which it has been programmed, but it is unable to adapt to new situations or learn from them. In response to that, Siegelmann introduces a model of super-Turing computation capable of modeling adaptive computations. The model, called analog recurrent neural network, allows both the efficient use of resources and the necessary flexibility to change its "program" according to the demands of the environment—hence, learning and adaptation [48]. From the conceptual standpoint, super-Turing computation is not only more expressive than the TM, but it is also radically different. Said in the language we are introducing here, super-Turing computation transcends CH and is placed as a particular case of nCH.

In any case, it should be clear that the TM works on a single scale, sequentially, and depending on an external operator. Life, on the other side, processes information locally and is able to synthesize (new, nontrivial) information on higher scales when dealing with given circumstances or solving relevant problems (e.g., acquiring nutrients, avoiding hazards, or

© 2014 Wiley Periodicals, Inc.
DOI 10.1002/cplx

reproduction [54]). That is, life does not possess all information needed to live, and that is why she has to synthesize it on the basis of genetic information, memory, and information about and from the environment [32]. Life's computation is self-organized and emergent.

Emergence in living systems is the outcome of self-organization, and by definition a TM is not a self-organized but a top-down controlled machine by an external operator. A TM is a closed system. Living systems, on the contrary are essentially open and incomplete systems. They need matter, energy, and information to live and they must seek them in the environment. The environment is intrinsically noncomputational and nonalgorithmic.

The TM is just one particular case of all possible types of computation (see, Ref. 55]). According to Dodig-Crnkovic [56], computation such as it appears in the natural world is much more general than what the TM is able to model—or simulate. As a matter of fact, in nature living beings process more than numbers and functions. Along this line, Wolfram [57] speaks of computational irreducibility. That is, there are phenomena which cannot be computationally solved and, therefore, the alternative is to simulate them and watch what happens. Among these kinds of phenomena are the biological systems and processes. In contrast, Mitchell [15] claims that the aim is certainly not just simulate biological systems but understanding the very phenomenon as computation.

It should be explicitly pointed out that the aim of BH does not consist in simulating life but to develop computational models capable of expressing the way life computes. In other words, the issue is not about making (computational) idealizations of life and the living processes but, on the contrary, understanding life and explaining via computation how life processes information. We just simply do not know how that computation is performed although good hints have been provided along the literature [58,59].

Let us put it straightforwardly: living systems lack an algorithmic solution. Presumably, living processes, for instance, biological, immunological, nervous, and others also lack an algorithmic solution. If so, BH is not only possible but necessary. In other words, biological processes—and henceforth living systems—operate on the basis of solving problems, adaptations, learning, and self-organization in such a way that life is made possible at the cost of normal algorithmic explanations. Life computes and creates always new functions hardly understandable ex ante. Life is concerned with highly complex computational problems such as protein folding, self-repair, interaction with the environment, metabolizing—well, development, and evolution. We find ourselves just on the eve of a nonmechanical and nonreductionist understanding and explanation of what life is and is about by what life does. To be sure, development and evolution are not about decidability/undecidability.

Science at large is, however, the story of understanding and making possible, via technological means, nonrecursive functions (and more adequately computable nonfunctions). Here is where the sciences of computation and biology play an important role.

As it happens many times in science, scientists and engineers make things without really knowing how things work in nature and why [60]. The examples go from physics to biology, from engineering to astronomy, for instance. To be sure, in the case of living beings, we have come to know that they are not machines, which is not a minor issue, and that the computation they carry out cannot by any means be reduced nor explained or understood in terms of the CTt. The computational architecture of livings systems is clearly not the Von Neumann architecture of classical computation. Within the framework of nonclassical computation other forms of better architectures have been tentatively introduced but usually in the context of the TM and the CTt (see, e.g., [61]).

BH steps aside of the "classical" idealizations that come to be in each case special cases of more general theories—very much in analogy as Euclidean geometry with respect to non-Euclidean geometries, classical logic vis-à-vis nonclassical logics, or classical mechanics in relation to the theory of relativity.

The arguments that support BH are:

i. Living beings are not just machines, in any concern; that is, even though they are physical entities they cannot be understood as mere machines or reduced to physics. New concepts and insights are needed;
ii. life is the complex phenomenon par excellence and it cannot be explained from the inferior or the lower stances. Life, it appears, is to be run and explained as it happens: it is not compressible, in terms of TM;
iii. the difference between the environment and the living being cannot by any means be traced. As the environment is an intrinsically open undetermined concept, there is no clear frontier between biotic and abiotic domains. Even though the environment consists in the atmosphere, the lithosphere, and the hydrosphere, the adaptability and manifold expressions of life do not exist apart from the work living organisms exert on the environment to which they adapt and they help transform. The concept of coevolution is most suitable here [62];

DOI 10.1002/cplx

TABLE 1

Historical/Relevant References in the Road toward BH

Field/Trend	Historical/Relevant References
Classical computation	[21,66]
Unconventional computation	[67–73]
Bioinspired models of computation	[74–78]
Biology as implementation substrate	[79–82]
Classical hypercomputation	[20–25,83]
Bioinspired hypercomputation	[84–88]
Nonclassical hypercomputation	[2,5,27,29,31,64]
Biological computation	[15,32–39,41,42,44–47]
Biological hypercomputation	[48,49,51–53]

iv. interaction—with the environment—is much more expressive than what an algorithm or a TM can be [2,5]; interactive computation whether from microorganisms on to human beings [63] or from one species to another to the environment is a matter that clearly opens up the door to BH;

v. life does not process in terms of classic logical functions—at least as they are known so far—; life is not worried by solving problems in the sense of mathematical functions, whatsoever, either. Other nonclassical logics are needed and can be developed and applied here. One of these nonclassical logics is paraconsistent logic [64];

vi. in computational terms, for the living systems the difference between the hardware and the software is irrelevant [32]. From a philosophical point of view, dualism can be superseded if not overcome;

vii. for life computation means living [54]; good computing means evolving and bad computing brings living systems to the edge of peril, danger, or extinction;

It follows that BH aims both at a rupture and a synthesis wherein the complexity—in the sense of the sciences of complexity—of life comes to the fore. The complexity is exactly the outcome of the asymmetry between the inside and the outside, that is, between the closed membrane and the autocatalytically closed system of components and the environment [64,65].

Figure 2 shows the historical development of computation models against a process of complexification. It should be taken as a general view, particularly regarding the time or historical/relevant references. It brings forth a comprehensive idea about the origins, ruptures, and synthesis wherein BH is born. We trace the very origins back to the 1930s, and the most important dates are mentioned. However, the core of the Figure lies in the developments that lead from classical computation on to BH.

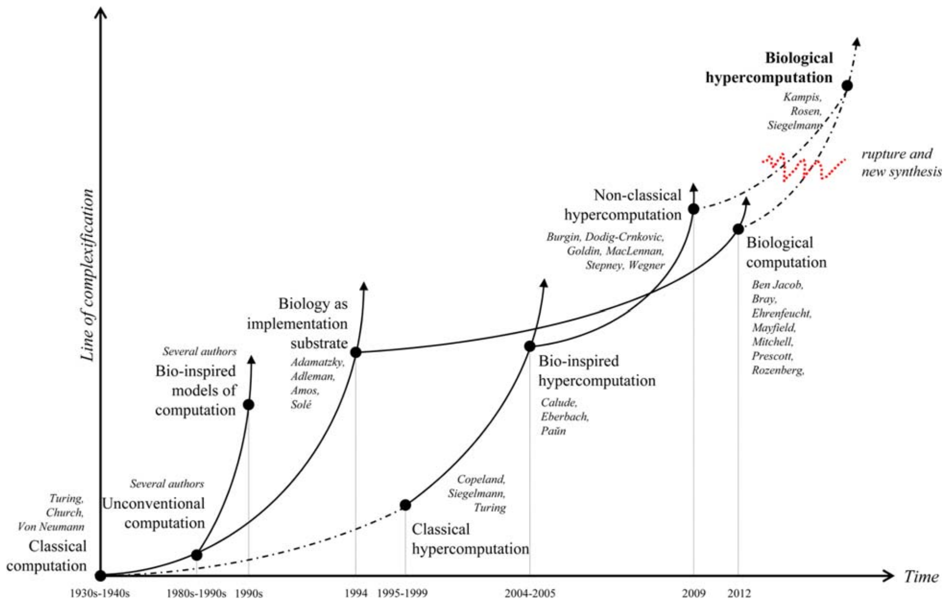
Table 1 presents a scheme of what can be properly named as the state-of-the art when considering the historical or relevant references that lead from classical computation to BH. It is a complement of Figure 2. The focus here is the key authors and years that support a given field or trend in the complexification of computation. Even though the main concern of this article is not the state-of-the-art the table is useful in that it provides a helpful insight that supports the main argument of our study here.

BH opens up a new research field that will transform and at the same time bring closer together biology and computing sciences, philosophy, and mathematics, to say the least. The goal consists in understanding and explaining the singularity of life—a singularity that remains within us and yet apart from us, so far. If the universe can be accounted for as a computational stance [89,90] then the consequence is unavoidable, namely the universe is alive—a conclusion many scientists have reached throughout other ways and roads [91–93]. The step must and can be made from the understanding of the universe as a computing system on as also and mainly a living phenomenon. This is the final or ultimate framework of BH but the departing gate is to be located in the study of proteins, bacteria, the immunological system, the DNA and RNA strands, on to the organic level, and the evolutionary scale.

Let us put it in analytical terms: the standard TM works with one tape on a sequential, that is, linear scale. Processing is performed as a black box in the sense that what is relevant is the translation of the Boolean logic into readable universal characters. This is what is called as a "general purpose machine." The processing is based on numerical characters. Processing means literally changing one thing into another, for instance Boolean characters into numbers, figures, letters, and the like. However, a TM does never synthesize information (the models of hypercomputation nearest to the TM that can overcome such a limitation are inductive TMs). In contrast, life is capable of synthesizing new information on different scales. Biological syntheses do transform one thing into another, indeed; however it produces brand new information that was not contained any means in the input, as in the TM. BH can adequately be grasped as metabolizing—information.

DOI 10.1002/cplx

FIGURE 2



The complexification of computation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In other words, saying that BH is not based on functions and numbers (the cornerstone of mathematics and physics) entails that current (mainstream) mathematics is insufficient. Explaining/understanding BH means a creation of new mathematics very much as also new ideas and concepts. A hint can be taken from IC. According to MacLennan, “some new models of computation have moved away from the idea of computation as the evaluation of a fixed function” [29]. Yet, it should be clear that IC cannot be expressed by or reduced to functions [3,30].

Life is a network of intertwined scales from bacteria to archaea to eukarya, and vice versa—on to the biosphere. In other words, from genes to cells to organs to systems on to the organism, and the interplay between homeostatic and homeorhetic processes. Life’s interaction with the environment is a question of resisting to the physical forces of the environment—homeorhesis. This is what Varela [94] dubs as the autonomy of living systems. However, at the same time once other living systems belong to the environment, the interaction is the interplay of cooperation and competition.

Life is a nonalgorithmic phenomenon. An algorithmic system works in three sequential steps, thus: input, processing, output, and they can never be mixed or intertwined. Computation literally happens in the second level, provided the input, and it takes place as a closed system (black box). Being essentially open, life never works in clearly stratified or differentiated sequential levels, for energy, matter, and information are constantly entering, being processed and produced in parallel, distributed, cross-linked dynamics. This is where BH happens and this is precisely what is to be understood. Hence, life does synthesize new information and new mechanisms for processing the new information—new information not present in original data.

All in all, BH is a cross concept that encompasses engineering, scientific—for instance the biological, physical, and computational sciences—, logical, and philosophical concerns. It aims at understanding life in computational terms, it shows the need for new computational paradigms, and explains multilevel, distributed, parallel, hybrid, and sequential (nonalgorithmic) information processes. BH links the microscopic view of life together with the macroscopic view of nature. As such it can be easily linked with one of the nonclassical logics, namely quantum logics (besides the call for paraconsistent logics we mentioned earlier). Thus, BH serves as a unitary although not a universal abstract concept. The concept might be considered as controversial, but it is a bet for a revolution in computer science. Furthermore, BH can bring brand new lights to the understanding of life and living beings. A revolution where the main participants are computer scientists working along with biologists, physicists, philosophers, and logicians.

DOI 10.1002/cplx

5. POSSIBILITIES, STANCES, AND REACH OF BH

BH entails from one extreme to the other enriching and deepening of the works and understanding of biology and computation. Moreover, biology and computation are, we believe, radically transformed by BH. In an applied sense, the sciences of complexity find in BH its most attractive if not feasible and open way for future research. In the theoretical standpoint, the question remains as to how exactly is BH performed in every level or scale and how that scale interacts, hypercomputationally, with the other levels of an organism, a species and the biosphere.

BH does not happen as a black box. First, the processing does not take place after the information is entered (input). Second, the output does not occur as a sequitur but is one and the same process as living. As a consequence, third, there are not two stances: the processing unit (PU) and the information, which is by definition external to the PU. There is no gap between the environment and a living system even though there is indeed an asymmetry and that is the key for BH. As it happens, in the TM and the CTt there is symmetry between input and output; that symmetry is linked and "processed" as a black box by the machine. In other words, the information that enters and the information gotten are computationally symmetric. This is exactly what the TM and the Von Neumann architecture are about. The tricky role is played, explained, and understood in terms of a black box, or is presumed to be on—tacitly.

More radically, there is no halting state for living beings, for such a state would be simple death. Complexity is about understanding how asymmetry is produced from symmetry, for life is asymmetric—whence, the arrow of time.

Conversely, life is an essentially asymmetric phenomenon that yet arises from symmetry—and breaks it down. This is what is usually called in the literature as dynamic equilibrium. The asymmetry life is sheds some new light on the following guess: BH is, correspondingly, an asymmetric processing phenomenon.

The main idea consists in highlighting how living "machines" and processes are not, in any concern, machine or mechanical processes. This claim might sound trivial, but mainstream science still considers—at least impliciter—biological dynamics and processes as mechanical. This can be seen throughout the language used, for instance, in neurology where many of the brain processes are still explained in terms of "on" and "off" switches; or in the immunological theory where many processes are understood in terms of "inhibitors" and "excititors." Many other cases can be easily brought out here. In classical terms, there is, although, a certain delay between the information processing and the actions performed as consequence. The traditional explanation has been in terms of instincts or impulses. The failure, however, is that they have been put out in terms of "on" and "off," which is quite mechanical and hence wrong.

One more key aspect is that by synthesizing new information and interacting with the environment living systems are capable of knowledge. Knowledge thus is to be grasped not just in terms of efficiency and useful assets, that is, information, but as a surplus wherein there is room for void and ambivalence, ambiguity and resilience, humor and irony—well, silence and patience, redundancy and doubt, for instance. The discussion about the centrality of the brain can be left aside here whereas we can reasonably argue in favor of mind, the very stance where knowledge arises from. BH sets forth a rationale for the distinction, not the opposition, between information and knowledge, brain and mind.

Several philosophical and social implications follow, then. Perhaps one of the most interesting is the promise that BH might transform technology, that is, engineering [95]. But we can also venture that BH is a multilevel, distributed, parallel, and sequential (but nonalgorithmic) processing, all at the same time, although not always with the same relevance. Indeed, even though BH remains on the theoretical ground for the time being, there are practical implications to be developed. These implications are, however, the subject of a different text.

6. CONCLUDING REMARKS

This article introduces the concept of BH. Whereas hypercomputation both opens and demands new computational models, the development of the concept and the reach and scope of the new computational models are issues that remain out of the scope here for reasons of time and space.

BH allows for a dynamic symmetry between computation and biology, for it transforms both in their current and classical sense into a brand new research field. This field is scientific as well as philosophical.

BH promises to open up the concept of computation and what it truly means and entails, and at the same time it makes possible to understand life in a more unified way. As there is no such a thing as a general theory of biology or of the living systems [15,63,91], BH, we believe, can pave the road toward such a theory.

The language about BH is set out in future tense in the literature. This, however, should not be taken in any sense as a mere desideratum or a wild idea or speculation. Scientific, technological, and philosophical research is being undertaken toward that future bringing it to the present.

So far, hypercomputation has not been capable of understanding biological processes, and develop a series of tools. In any case, to be sure, there is

DOI 10.1002/cplx

one more complex arena that arises here, namely how the mind hypercomputes. This issue raises questions and possibilities not just for the computation sciences and biology but also for philosophy, epistemology, and mathematics if not also for logics. We have aimed at understanding life from a computational point of view; but it should be clear that we do not pretend to reduce life to a computational framework.

ACKNOWLEDGMENTS

The authors have received very helpful comments on a previous version from E. Eberbach, particularly about Section 4. Authors hope they have improved the section as to the concern about why life cannot be modeled by machines, as well as to the issue pertaining the range and scope of mathematics. From M. Burgin, authors have carefully reflected about the physical character of information, something he is not quite convinced about. Here authors tried to make our arguments more convincing. From G. Dodig-Crnkovic authors learned to make more explicit the link between the universe as a computational and as a living stance. Finally, authors also want to thank L. F. Niño for his comments and advise on a previous draft. To them all authors express their highly gratitude. It goes without saying: if the article still presents some flaws they are only the authors' responsibility.

REFERENCES

1. Pagels, H. The Dreams of Reason: The Computer and the Rise of the Sciences of Complexity; Simon & Schuster: New York, 1988.
2. Goldin, D.; Wegner, P. The interactive nature of computing: Refuting the strong church-turing thesis. Minds Mach 2008, 18, 17–38.
3. Eberbach, E.; Goldin, D.; Wegner, P. Turing's ideas and models of computation. In: Alan Turing: Life and Legacy of a Great Thinker; Teuscher, C., Ed.; Springer: Berlin, 2004; pp 159–194.
4. Wegner, P.; Eberbach, E. New models of computation. Comput J 2004, 47, 4–9.
5. Wegner, P.; Eberbach, E.; Burgin, M. Computational completeness of interaction machines and Turing machines. In: Turing-100. The Alan Turing Centenary; Voronkov, A., Ed.; EasyChair: Manchester, 2012; pp 405–414.
6. Ionescu, M.; Martín-Vide, C.; Păun, A.; Păun, G. Unexpected universality results for three classes of P systems with symport/antiport. Nat Comput 2003, 2, 337–348.
7. Păun, G. Membrane computing: Power, efficiency, applications. In: New Computational Paradigms; Cooper, S.B.; Löwe, B.; Torenvliet, L., Eds.; Springer: Berlin, 2005; pp 396–407.
8. Cook, M. Universality in elementary cellular automata. Complex Syst 2004, 15, 1–40.
9. Goucher, A.P. Universal computation and construction in GoL cellular automata. In: Game of Life Cellular Automata; Adamatzky, A., Ed.; Springer: London, 2010; pp 505–517.
10. Takahara, A.; Yokomori, T. On the computational power of insertion-deletion systems. Nat Comput 2003, 2, 321–336.
11. Landweber, L.; Kari, L. Universal molecular computation in ciliates. In: Evolution as Computation; Landweber, L.; Winfree, E., Eds.; Springer: Berlin, 2002; pp 257–274.
12. Balan, M.S.; Jürgensen, H. On the universality of peptide computing. Nat Comput 2008, 7, 71–94.
13. Ignatova, Z.; Martínez-Pérez, I.; Zimmermann, K. DNA Computing Models; Springer: Berlin, 2008.
14. Syropoulos, A. Hypercomputation: Computing Beyond the Church-Turing Barrier; Springer: New York, 2008.
15. Mitchell, M. Biological computation. Comput J 2012, 55, 852–855.
16. Calude, C.S.; Păun, G. Computing with cells and atoms in a nutshell. Complexity 2001, 6, 38–48.
17. Jablonka, E.; Lamb, M.J. Evolution in Four Dimensions: Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life; MIT Press: Cambridge, MA, 2006.
18. Davies, P.; Gregersen, N.H., Eds. Information and the Nature of Reality: From Physics to Metaphysics; Cambridge University Press: Cambridge, 2010.
19. Vedral, V. Decoding Reality: The Universe as Quantum Information; Oxford University Press: Oxford, 2010.
20. Copeland, B.J. Hypercomputation. Minds Mach 2002, 12, 461–502.
21. Turing, A.M. On computable numbers, with an application to the entscheidungsproblem. Proc London Math Soc 1936, 42, 230–265.
22. Turing, A.M. Systems of logic based on ordinals. Proc London Math Soc 1939, 45, 161–228.
23. Copeland, B.J. Super Turing-machines: A review of universal Turing machines, AUTMs, and O-machines. Complexity 1998, 4, 30–32.
24. Copeland, B.J.; Proudfoot, D. Alan Turing's forgotten ideas in computer science. Sci Am 1999, 280, 99–103.
25. Siegelmann, H.T. Computation beyond the Turing limit. Science 1995, 268, 545–548.
26. Burgin, M. Super-Recursive Algorithms; Springer: New York, 2005.
27. Stepney, S. Non-classical hypercomputation. Int J Unconv Comput 2009, 5, 267–276.
28. Stannett, M. The case for hypercomputation. Appl Math Comput 2006, 178, 8–24.
29. MacLennan, B. Super-Turing or non-Turing computation? Extending the concept of computation. Int J Unconv Comput 2009, 5, 369–387.
30. Wegner, P. Interactive foundations of computing. Theor Comput Sci 1998, 192, 315–351.
31. Goldin, D.; Smolka, S. A.; Wegner, P., Eds. Interactive Computation: The New Paradigm; Springer: Berlin, 2006.

DOI 10.1002/cplx

32. Ben-Jacob, E. Learning from bacteria about natural information processing. Ann NY Acad Sci 2009, 1178, 78–90.
33. Amos, M.; Hodgson, D.A.; Gibbons, A. Bacterial self-organisation and computation. Int J Unconv Comput 2007, 3, 199–210.
34. Prescott, D.M.; Rozenberg, G. How ciliates manipulate their own DNA – A splendid example of natural computing. Nat Comput 2002, 1, 165–183.
35. Ehrenfeucht, A.; Harju, T.; Petre, I.; Prescott, D.M.; Rozenberg, G. Computation in Living Cells: Gene Assembly in Ciliates; Springer: Berlin, 2004.
36. Brijder, R.; Daley, M.; Harju, T.; Jonoska, N.; Petre, I.; Rozenberg, G. Computational nature of gene assembly in ciliates. In: Handbook of Natural Computing; Rozenberg, G.; Bäck, T.; Kok, J.N., Eds.; Springer: Berlin, 2012; pp 1233–1280.
37. Ehrenfeucht, A.; Rozenberg, G. Reaction systems: A model of computation inspired by biochemistry. In: Developments in Language Theory; Gao, Y.; Lu, H.; Seki, S.; Yu, S., Eds.; Springer: Berlin, 2010; pp 1–3.
38. Ehrenfeucht, A.; Kleijn, J.; Koutny, M.; Rozenberg, G. Reaction systems: A natural computing approach to the functioning of living cells. In: A Computable Universe: Understanding and Exploring Nature as Computation; Zenil, H., Ed.; World Scientific: Singapore, 2012; pp 207–226.
39. Ben-Hur, A.; Siegelmann, H.T. Computation in gene networks. Chaos 2004, 14, 145–151.
40. Kari, L.; Rozenberg, G. The many facets of natural computing. Commun ACM 2008, 51, 72–83.
41. Bray, D. Wetware: A Computer in Every Living Cell; Yale University Press: New Haven, 2009.
42. Forrest, S.; Hofmeyr, S.A. Immunology as information processing. In: Design Principles for the Immune System and Other Distributed Autonomous Systems; Segel, L. A.; Cohen, I. R., Eds.; Oxford University Press: Oxford, 2001; pp 361–387.
43. Dasgupta, D.; Niño, L.F. Immunological Computation: Theory and Applications; CRC Press: Boca Raton, 2009.
44. Detrain, C.; Deneubourg, J. L.; Pasteels, J. M., Eds. Information Processing in Social Insects; Birkhäuser: Basel, 1999.
45. Peak, D.; West, J.; Messinger, S.; Mott, K. Evidence for complex, collective dynamics and emergent, distributed computation in plants. Proc Natl Acad Sci USA 2004, 101, 918–922.
46. Maynard-Smith, J.; Szathmáry, E. The Major Transitions in Evolution; Oxford University Press: Oxford, 1995.
47. Mayfield, J.E. The Engine of Complexity: Evolution as Computation; Columbia University Press: New York, 2013.
48. Siegelmann, H.T. Turing on super-Turing and adaptivity. Prog Biophys Mol Biol 2013, 113, 117–126.
49. Rosen, R. Life Itself; Columbia University Press: New York, 1991.
50. Louie, A.H. A living system must have noncomputable models. Artif Life 2007, 13, 293–297.
51. Rosen, R. Essays on Life Itself; Columbia University Press: New York, 2000.
52. Kampis, G. Self-Modifying Systems in Biology and Cognitive Science; Pergamon Press: Oxford, 1991.
53. Kampis, G. Self-modifying systems: A model for the constructive origin of information. Biosystems 1996, 38, 119–125.
54. Tsuda, S.; Zauner, K.P.; Gunji, Y.P. Computing substrates and life. In: Explorations in the Complexity of Possible Life: Abstracting and Synthesizing the Principles of Living Systems; Artmann, S.; Dittirch, P., Eds.; IOS Press: Amsterdam, 2006; pp 39–49.
55. Burgin, M.; Dodig-Crnkovic, G. From the closed classical algorithmic universe to an open world of algorithmic constellations. In: Computing Nature: Turing Centenary Perspective; Dodig-Crnkovic, G.; Giovagnoli, R., Eds.; Springer: Berlin, 2013; pp 241–253.
56. Dodig-Crnkovic, G. Dynamics of information as natural computation. Information 2011, 2, 460–477.
57. Wolfram, S. Computer software in science and mathematics. Sci Am 1984, 251, 188–203.
58. Gros, F. Les nouveaux mondes de la biologie; Odile Jacob: Paris, 2012.
59. Carlson, R.H. Biology is Technology: The Promise, Peril, and New Business of Engineering Life; Harvard University Press: Cambridge, MA, 2010.
60. Devlin, K. The Millennium Problems: The Seven Greatest Unsolved Mathematical Puzzles of Our Time; Basic Books: New York, 2002.
61. Sipper, M. Evolution of Parallel Cellular Machines: The Cellular Programming Approach; Springer: Heidelberg, 1997.
62. Thompson, J.N. The Coevolutionary Process; The University of Chicago Press: Chicago, 1994.
63. Margulis, L.; Sagan, D. Microscosmos: Four Billion Years of Microbial Evolution; The University of California Press: California, 1997.
64. Dodig-Crnkovic, G. Significance of models of computation, from Turing model to natural computation. Minds Mach 2011, 21, 301–322.
65. Hoffmeyer, J. Semiotic freedom: An emerging force. In: Information and the Nature of Reality: From Physics to Metaphysics; Davies, P.; Gregersen, N. H., Eds.; Cambridge University Press: Cambridge, 2010; pp 185–204.
66. Church, A. An unsolvable problem of elementary number theory. Am J Math 1936, 58, 345–363.
67. Adamatzky, A. Computing in Non-Linear Media and Automata Collectives; IOP Publishing: Bristol, 2001.
68. Benioff, P. Quantum mechanical Hamiltonian models of Turing machines. J Stat Phys 1982, 29, 515–546.
69. Calude, C.S.; Casti, J.; Dinneen, M.J., Eds. Unconventional Models of Computation; Springer: Singapore, 1998.
70. Crutchfield, J.P. The calculi of emergence: Computation, dynamics, and induction. Physica D 1994, 75, 11–54.
71. Cosenza, G.; Neri, F. Cellular automata and nonlinear dynamical models. Physica D 1987, 27, 357–372.
72. Forrest, S. Emergent Computation: Self-Organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks; MIT Press: Cambridge, 1991.
73. Toffoli, T.; Margolus, N. Cellular Automata Machines; MIT Press: Cambridge, MA, 1987.
74. Berry, G.; Boudol, G. The chemical abstract machine. Theor Comput Sci 1992, 96, 217–248.
75. Bishop, C.M. Neural Networks for Pattern Recognition; Oxford University Press: Oxford, 1995.
76. Dasgupta, D., Ed. Artificial Immune Systems and their Applications; Springer: Berlin, 1999.

DOI 10.1002/cplx

77. Gibbons, A.; Amos, M.; Hodgson, D. Models of DNA computation. In: Proceedings of Mathematical Foundations of Computer Science (MFCS), Cracow, Poland, 1996; Springer: Berlin, 1996; pp 18–36.
78. Păun, G. Computing with membranes: An introduction. Bull EATCS 1999, 68, 139–152.
79. Adleman, L. Molecular computation of solutions to combinatorial problem. Science 1994, 266, 1021–1024.
80. Benenson, Y.; Paz-Elizur, T.; Adar, R.; Keinan, E.; Livneh, Z.; Shapiro, E. Programmable and autonomous computing machine made of bio-molecules. Nature 2001, 414, 430–434.
81. Regot, S.; Macia, J.; Conde, N.; Furukawa, K.; Kjellén, J.; Peeters, T., Hohomann, S.; de Nadal, E.; Posas, F.; Solé, R. Distributed biological computation with multicellular engineered networks. Nature 2011, 469, 207–211.
82. Tolmachiev, D.; Adamatzky, A. Chemical processor for computation of Voronoi diagram. Adv Mater Opt Electr 1996, 6, 191–196.
83. Turing, A.M. Intelligent machinery. In: Collected works of AM Turing: Mechanical intelligence; Ince, D. C., Ed.; Elsevier: Amsterdam, 1992 (original 1948); pp 107–128.
84. Păun, G. (DNA) computing by carving. Soft Comput 1999, 3, 30–36.
85. Calude, C.S.; Păun, G. Bio-steps beyond Turing. Biosystems 2004, 77, 175–194.
86. Eberbach, E. Toward a theory of evolutionary computation. Biosystems 2005, 82, 1–19.
87. Burgin, M.; Eberbach, E. Evolutionary automata: Expressiveness and convergence of evolutionary computation. Comput J 2012, 55, 1023–1029.
88. Gheorghe, M.; Stannett, M. Membrane system models for super-Turing paradigms. Nat Comput 2012, 11, 253–259.
89. Lloyd, S. Programming the Universe; Alfred A. Knopf: New York, 2006.
90. Barrow, J.D. The Artful Universe; Oxford University Press: New York, 1995.
91. Kauffman, S. Investigations; Oxford University Press: New York, 2000.
92. Griffin, J. In the Beginning: The Birth of the Living Universe; Little Brown & Co: New York, 1993.
93. Smolin, L. The Life of the Cosmos; Oxford University Press: Oxford, 1997.
94. Varela, F. El fenómeno de la vida; Santiago de Chile: Dolmen, 2000.
95. Maldonado, C.E.; Gómez-Cruz, N. The complexification of engineering. Complexity 2012, 17, 8–15.

DOI 10.1002/cplx