GUIDELINES

# From artificial evolution to computational evolution: a research agenda

*Wolfgang Banzhaf, Guillaume Beslon, Steffen Christensen, James A. Foster, François Képès, Virginie Lefort, Julian F. Miller, Miroslav Radman and Jeremy J. Ramsden*

Abstract | Computational scientists have developed algorithms inspired by natural evolution for at least 50 years. These algorithms solve optimization and design problems by building solutions that are 'more fit' relative to desired properties. However, the basic assumptions of this approach are outdated. We propose a research programme to develop a new field: computational evolution. This approach will produce algorithms that are based on current understanding of molecular and evolutionary biology and could solve previously unimaginable or intractable computational and biological problems.

Useful algorithms can be designed by imitating natural evolution[1–4]. For example, an algorithm inspired by evolution that aims to estimate the phylogeny of a set of amino-acid sequences might 'evolve' one or more sets of parameters that describe the topology and branch lengths of a phylogenetic tree. It might do so by simulating the data under each of several proposed trees, selecting trees that produce higher probabilities, varying those trees by exchanging or altering parameters, replacing older trees with new ones, and repeating the process. The analogy with evolution by reproduction, variation and natural selection is obvious.

We refer collectively to the current field of studying and applying such algorithms as artificial evolution (AE); this term groups approaches that are sometimes distinguished in the literature, such as evolutionary programming[5], genetic algorithms[6], evolutionary strategies[7] and genetic programming[8]. AE has traditionally had a strong engineering focus, and has been applied successfully in such disparate areas as the optimization of spacecraft trajectories[9], nano-transistor design[10] and quantum circuits[11], and has performed as well as or better than human designers in some areas such as designing analogue circuits[12,13].

The purpose of AE research is to develop new algorithms that solve difficult design and optimization problems. Many biological problems are of this type, such as inferring regulatory networks from microarray data, inferring phylogenies from taxa, classifying protein sequences and discovering remote homologies. AE, however, can do more. Significantly, AE provides a framework for a computational model of evolution that could help us to understand and quantify the mechanisms, trajectories and pace of natural evolution. By providing an *in silico* test bed for evolutionary biology, the AE approach is useful in testing hypotheses about mechanisms for speciation or in quantifying trade-offs among gene regulatory networks or metabolic networks, for example.

However, most current AE research proceeds from a restricted and dated understanding of natural evolution. For example, most AE 'genomes' are small, and are mapped directly into correspondingly simple 'phenotypes', both with little variation. In particular, AE rarely uses self-modification, or feedback of any kind, even though these features are pervasive in natural evolution. AE approaches typically drive relentlessly to an *a priori*, invariant objective and stop once they get there.

We propose a richer paradigm for algorithms inspired by evolution, which we call computational evolution (CE). The term has been used before, but in a less general sense[14–16]. We are a group of AE researchers and evolutionary biologists who met in the summer of 2005 to identify specific biological concepts that should be incorporated more fully into AE. Our hope is that CE will enable computational scientists to address new, difficult problems, including those of interest to natural scientists, and that the more sophisticated evolutionary modelling in CE will be directly useful to evolutionary biologists and ecologists as a basis for simulations. We also hope to promote a more sophisticated dialogue between computational and natural scientists about evolution.

In the following section, we give an overview of the limitations of AE. We then present some specific biological concepts that, if incorporated, would transform AE into CE. Some of these concepts have been explored individually in AE research, but none has been systematically applied. We also present some specific research areas that CE could address. In the final section, we discuss the difficulties one might face pursuing CE and suggest how they might be overcome.

## The problems with artificial evolution

***Material of evolution.*** In most AE applications, information, an abstraction represented in programming code, is the material of evolution. These AE applications assume a simple model of 'genotype', 'phenotype' and 'fitness', in which simple relationships exist between these concepts. The so-called 'central dogma of artificial evolution' was inspired by Crick's formulation of the 'central dogma of biology' (FIG. 1a,b), which states that information flows in one direction, from DNA to proteins. This simple model of information flow manifests itself in AE as an evolutionary model (FIG. 1c,d) in

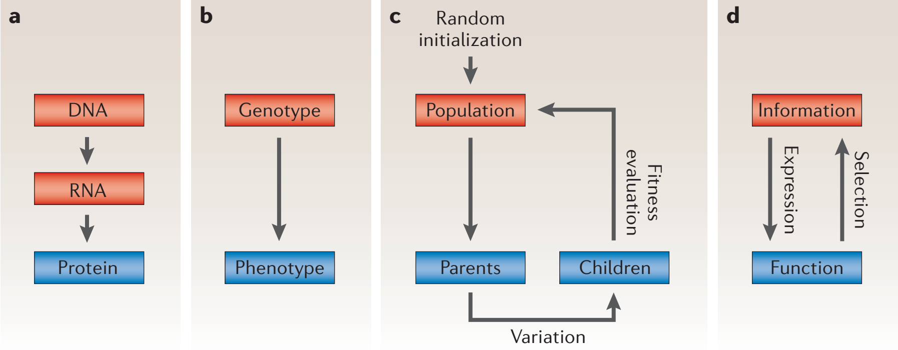

Figure 1 | **The biological 'central dogma' as implemented in artificial evolution. a** | A simplified view of the biological 'central dogma', namely that DNA is translated into RNA and then expressed as a phenotype by means of protein function, with selection operating on function. **b** | Artificial evolution (AE) assumes that, in biological systems, information flows from the genotype to a functional phenotype by being expressed as proteins; this view of a one-way flow of information is derived from the biological central dogma shown in panel **a**. **c** | The typical AE algorithm is an iterative loop in which information-carrying individuals in a population are randomly varied and then evaluated with respect to the objective of the algorithm, thereby assigning a fitness to the individual that determines which individuals are selected for the next generation. **d** | Note that there is no feedback from the evaluation of any individual to information other than through selection, which determines the next generation. In each case (panels **a**–**d**), information (red) is expressed as a function (blue), which forms the basis of selection.

which a population of individuals, each with a genotype encoding a single phenotype, undergoes variation and selection; changes in adjacent bits in the computer's representation of the genotype tend to change the same or closely related features of the phenotype. AE optimization algorithms typically make the following assumptions: that the genome is composed of discrete and independent genes, that there is no interesting role for non-genic material in the genome, that all genes are passed directly from parent to offspring, that variation occurs only at the genetic level, that selection operates only on the phenotype, and that the organization of the genome remains essentially unchanged throughout evolution.

Biological evolution is now understood to be much more intricate and nuanced, with evolution working at all levels of this complexity. DNA itself, the genetic code, and all control and feedback mechanisms are themselves products of evolution. RNA, alternative splicing, genetic regulation, retrotransposition and so on have made defunct the idea of a simple, direct relationship between gene and character. Natural evolution offers much richer possibilities for tuning, self-regulation, feedback and evolvability, yielding a level of complexity that was previously inconceivable[17]. AE must also evolve in ways that correspond to advances in evolutionary genetics.

***Mechanisms of evolution.*** Even Crick's dogma allows more complexity than AE. For example, transcription and translation[18,19] allow differential gene expression and a mapping of one genotype to many phenotypes; these processes operate simultaneously on many genomic loci, and modify templates, transcripts and products. By contrast, the most common implementation of genotype-to-phenotype mapping in AE is simple and direct (one-to-one), with some notable and encouraging exceptions[20–23].

Natural evolution is an open-ended process. But AE for optimization is closed, with precise invariant objectives. This approach reduces the amount of variation present during AE runs. It precludes emergent phenomena such as the re-use of genotypic and phenotypic structures for new purposes, complex networks that regulate component interactions, dynamic mapping between layers of information, interactions between individuals, and speciation.

***Problems for evolution to solve.*** The current approach to AE limits our vision of what is possible for algorithms that use evolution. Although current AE approaches might be adequate[24,25] for solving single, specific optimization or design problems, solving more difficult problems will require a richer evolutionary understanding. Complexity and robustness in nature is proof that biologically inspired processes can be powerful algorithmic tools. To ignore this is to seriously impoverish our algorithms. We present potential problems that CE might be able to address in BOX 1.

## Using biology to transform AE into CE

We recommend that AE be transformed into CE by incorporating algorithmic analogues of our current understanding of natural evolution. These recommendations are summarized in BOX 2. Such a transformation should be articulated by addressing three principal questions. The first concerns the material on which evolution acts, and requires revising the concept of what is evolving. The second question relates to the mechanisms of evolution: how does evolution happen? The third question relates to the problems that CE needs to address, and the practical uses that it can bring. These three questions are explored in turn in the following sections.

### Materials of evolution

***Physicality and embodiment.*** Natural evolution acts on physical systems[26–28]; it operates on a large complex of molecules, including DNA, that interact dynamically with themselves and a host of other molecules, therefore allowing complex interactions to emerge and cascade in unexpected ways. This process enables natural systems to discover new ways to represent and interpret information, which are then available as parts for further evolution. Evolution is a tinkerer[29].

By contrast, AE suffers largely from the curse of programmability[30]. It tends to assume that the genetic material is symbolic rather than physical, so that it is not influenced by physical properties such as electrostatic charge and temperature. AE implementations also tend to follow good programming practice by avoiding side effects, which could, for example, cause a program module to change the behaviour of an unrelated module.

CE research will include the evolution of physical systems, or simulated physical systems, rather than abstract entities. Productive steps towards this more biologically significant feature, physicality, have involved evolving circuits on actual silicon chips. Miller and Downing have argued that non-traditional 'hardware', such as liquid crystals rather than computer chips, might be even more suitable for computer-controlled evolution[31,32].

In short, just as natural evolution happens in a physical universe, and takes advantage of embodiment in surprising ways, CE will explore the creative possibilities of evolution in real or simulated[33–36] physical and chemical systems.

***Metabolism and dynamically transforming components.*** Living systems have evolved complex metabolic networks of molecular transformations, which are often mediated by enzymes[18]. These networks frequently incorporate sophisticated regulatory mechanisms, such as allosteric inhibition or substrate-induced enzyme clustering. As well as ensuring that all the requisite molecules are available to the cell, metabolism captures enough energy from the organism's environment to allow survival.

Most algorithms, including AE, are based on abstract machines, such as Turing machines, that tacitly assume that sufficient energy exists to execute all necessary operations. CE will follow the lead of some artificial life research[37,38] and recent AE research[39] by taking into account finite energy and resource availability. For example, CE will provide simulations where adjacent individuals or subpopulations compete for different limited nutrients, in order to test hypotheses about how strategies for nutrient uptake, such as generalism versus specialism, emerge from subpopulations.

***Cellularization and development.*** Natural genomes orchestrate a vast network of interactions[40–42] that guide the development of living beings, which often have extensively compartmentalized units such as organelles, cells and tissues. Therefore, development turns a time-independent entity (the genome as an information source) into an autonomously adaptive system (the organism) in which time is an essential variable. The resulting organisms are hierarchical collections of structures within structures, with interactions between each level, which smooth out the effects of environmental fluctuations on the organism[43].

Similarly, in CE, the network of interactions between genetic information and evolving individuals will be dynamically and hierarchically organized, with different levels of organization able to affect each other. Achieving this goal requires a new approach to algorithms inspired by evolution, in which individual data structures can be nested and interact with each other.

***Ecosystems.*** Ecological interactions are ubiquitous at all scales, and change the nature of individuals and entire ecosystems. Such changes constrain the variety of available species, and alter the rate and effect of evolution. Some phenomena in nature, such as food webs, symbiosis and disease, require complicated ecosystems even to make sense.

CE incorporates adaptive ecosystems, which create new challenges for individuals, and thereby new ecosystems. In a sense, this is merely a restatement of our earlier call for emergent hierarchies of interactions and behaviours, but now at the population level rather than at the individual level. This is not surprising, as a population can be treated as an 'individual' for the purposes of modelling or understanding ecosystems.

### Mechanisms of evolution

***Genotype-to-phenotype mapping.*** In natural evolution, the transformation of genotype into phenotype is a multilayered process that operates on diverse elements, distributed throughout the cell[18], with many paths for feedback. For example, the activity of a protein might vary according to its phosphorylation state; the activity of RNA might vary because of alternative splicing or post-transcriptional modification; or the activity of a gene might vary as a result of methylation.

One important feature of CE is the incorporation of complex, evolvable relationships between genotype and phenotype[44,45], involving multiple interconnecting processes that have the potential for interactions with themselves and each other. The relationship between genotype and phenotype should preferably be emergent, not hard-wired (BOX 3).

***Exaptation and innovation.*** The recruitment of a component selected for one use to a new purpose (exaptation) could be an important source of innovation in natural morphogenesis[46] and a source of new functionality, such as the efficient uptake of oxygen in the eyes

### Box 1 | Representative challenges that are more likely to be met by CE than by current AE

The challenges listed below are not specific, and are far from exhaustive: they are suggestive of topics that could be beneficial to both artificial evolution (AE) and biological researchers. We do not claim that computational evolution (CE) will provide the final answer to these challenges. Rather, CE will provide a platform from which to discover and test hypotheses in an appropriately realistic way.

#### Genomic organization

CE will allow researchers to explore the possible interactions between networks at different levels of abstraction. For example, CE would enable simulations to:

- explore how complex metabolic networks might facilitate or inhibit the emergence of complex gene expression networks during evolution;
- explore how endosymbiosis or independently replicating transposable elements might affect the complexity of the genome;
- develop and test scenarios for the transition from an RNA world to a DNA world, or the emergence of cellularity.

#### Speciation

CE will provide a simulation environment that is rich enough to implement different models of speciation. This framework will allow researchers to formulate and test hypotheses about organismal and ecological interactions that lead to sympatric versus allopatric speciation, and to model the influence of co-evolution on speciation.

#### Environmental sensing

CE can enable individuals to discriminate between environmental cues. For example, autonomous robots, especially those operating in environments that they are themselves modifying, could use CE to select different sensory data to solve the same problem; a biological analogy would be the evolved use of hearing instead of sight to capture insects in both bats and swallows.

#### Continuous data mining

CE could autonomously explore databases that change significantly and rapidly, allowing researchers to periodically sample inferences for new insights that were not previously available. Potential targets are bioinformatics and terrestrial image databases.

#### Innovative design

CE allows for new variables to be created or old ones to be eliminated; this approach is unlike that implemented in routine design, which works within predetermined constraints. For example, innovative molecular design of antibiotics could require that new variables be introduced to account for the adaptive responses of both the host and target organisms, or variables that are considered important initially, such as the ability of a molecule to interact with non-organic molecules, could prove to be unnecessary and be eliminated.

#### Hypothesis generation

CE enables algorithms to vary which sets of data are relevant to a given problem, without relying on presuppositions about which data are important and which types of hypothesis should be explored. For instance, drug discovery is sometimes like a good mystery story, in that many molecular interactions are known, but it is unclear which are important for a specific pathology. CE could generate potentially interesting hypotheses about molecular interaction pathways.

and swim bladder of fishes[47]. The probable transition from RNA-based life to DNA-based and protein-based life, the emergence of multicellularity, and even genotype-to-phenotype mapping, might be examples of exaptation building on modules of molecular activities[48]. Exaptation presupposes a rich set of interactions between components and functions, so that a change in one item changes others; it might therefore happen that a side-effect is more valuable than the feature originally selected for, leading to amplified selection and eventual fixation of the new function. Another key to emergent, structured complexity and stability in nature is the way in which evolution acts on weakly coupled modular systems[49,50].

In nature, co-evolution of evolved units is the rule not the exception. CE encourages interactions between components and explicitly promotes emergent modularity, which allows slight changes in a variable to produce surprising innovations and novel selective advantages. For example, data structures that represent parts of a genome, or part of a genome translation pathway, might duplicate and diverge, or one such element might be translocated into another. Another potentially powerful CE strategy is to allow individuals to alter their environment and then to adapt both to the new conditions and to their original selective pressure.

*Enabling variety.* Living beings counter diversity and unpredictability in the challenges they face with corresponding variety in their parts[51]. Sensory organs provide a rich range of responses to stimuli by presenting a variety of detectors for diverse specific signals. Immune systems retain a vast memory of previously encountered antigens with diverse challenge-specific B cells. In some genomes, introns and intergenic sequences might provide a vast reservoir of dormant genetic material that can be mobilized when needed[52]. These are all networks that generate variety from a relatively small number of different specialist components.

CE overcomes the constraints that computational limits have historically forced on AE. Again, the key is to choose data structures that, by interacting, increase the number of internal degrees of freedom in the system. The target number of interacting components should be several orders of magnitude greater than current population sizes or the number of features described in current AE. Alternatively, it might be necessary to introduce noise into systems inspired by evolution, rather than relying on pseudo-random sources of variation[40,53].

## Problems to address

*Open-endedness.* In an open-ended setting — such as the one in which natural systems operate — different evolutionary trajectories can coexist, providing new degrees of freedom. This leads organisms to develop specific partial solutions to multiple challenges. For example, species with heavy parental investment in offspring might coexist with others with prodigious numbers of progeny. Moreover, while developing such partial solutions, organisms will modify and thereby restructure their environments, changing the challenges to which future generations must adapt.

CE systems would target open-ended problems, rather than those that are precisely specified *a priori*. Open-ended systems still have constraints, however. Just as natural organisms are constrained by spatial, energetic and other resource limitations in their environment, so CE systems are constrained by open-ended specifications and by the state of the evolving system. Evolvability is as important in CE[54–56] as it is in nature[57–59], because of the interplay between degrees of freedom and necessities imposed by constraints. To be evolvable, systems must not be overly constrained, or they will adapt and stop changing; nor can they be under-constrained, as that would lead to mere random variation.

*Interpretation of genome-encoded information.* DNA is more than just a passive information carrier: it is also an active participant in transcription, translation, and packaging and organizing the genome[60]. CE incorporates this insight. For example, by encoding key features of the translation machinery in the genome, such as faithfulness and specificity, these features become evolvable. Indeed, combining the evolution

### Box 2 | The road to computational evolution

Computational evolution (CE) originates from richer algorithmic approaches to three questions about evolution: what evolves, how does evolution happen, and what use is evolution?

**What evolves? The stuff of evolution**

- CE simulates evolution in an environment where physical features of the evolving entities can affect how they change. For example, just as the ‘sticky’ nature of oligonucleotides affects how DNA-based organisms evolve, an embodied CE system using liquid crystals or electrical circuits could take advantage of electromagnetic properties that are unavailable to mere simulations.
- CE will allow simulations in which different materials with different properties interact. Natural evolution often relies on this sort of interaction, as for example when DNA binding proteins interact with DNA molecules to regulate gene expression. With CE, hardware and software might co-evolve, which is analogous to how genomes and molecular interactions co-evolve in nature.
- With CE, nested data and control structures will interact simultaneously at multiple scales: basic data objects will interact with each other, populations of individuals will interact with other populations, populations will interact with atomic data structures, and so on. In particular, the environment in which evolution occurs in CE will itself evolve, and this will shape the evolutionary trajectories of other evolving entities. It is impossible to imagine natural evolution without interactions that scale from molecules to ecosystems.

**How does evolution happen? The mechanisms of evolution**

- Interactions between evolving objects and processes in CE will allow feedback loops to arise within or between objects and processes. This makes new evolutionary trajectories possible. Control mechanisms in nature, from methylation to horizontal gene transfer, are feedback loops by which evolving individuals, from molecules to populations, respond to change.
- As in nature, selection pressure in CE will be ‘felt’ at the level of basic data and control items (analogous to molecules and molecular interactions), collections of such items (analogous to cells, tissues and individuals), and even simulated species and ecosystems.
- CE will provide mechanisms for reproductive isolation and for merging and integrating evolving objects. After all, there is more to natural evolution than independent evolving organisms diverging cleanly into new species.

**Why bother with CE? The application of evolution**

- CE will make it possible to design algorithms that address open-ended problems. This is typically not possible with traditional algorithm design, which presupposes fixed specifications.
- CE will make it possible to design computational artefacts, from programs to special-purpose hardware, that must operate even when conditions and constraints vary in unpredictable ways.
- CE will provide a more life-like foundation for addressing problems in which the specifications might be unknown or too complicated to express (BOX 3).

of interpretation processes with physical materials that themselves evolve allows information to flow from the hardware (the genome) to the software (the machinery that interprets the information stored in the genome).

***Repertoire exploitation.*** Some natural systems, such as the stress response in prokaryotes and the immune system, can retain a large repertoire of potential responses to support rapid responses to unusual stimuli. These systems effectively maintain counterselection by retaining a population of 'sleeping' cells or molecules that consume few resources, but which can be rapidly activated. Efficiency in these systems demands a division of effort between generalists that sample the space of possible stimuli and specialists that respond directly to individual challenges[61]. CE implements algorithmic analogues of biological repertoire exploitation with larger, more diverse, less constrained population sizes.

***Speciation.*** Speciation is the mechanism by which natural systems self-organize into groups of genomes that evolve (relatively) independently[62,63]. Reproductive barriers are not always absolute, as hybridization is common in plants, horizontal gene transfer is common in prokaryotes[64], and occasionally entire genomes can be acquired symbiotically[65]. In fact, genetic barriers between related species can be removed or established by the level of expression of single mismatch repair proteins[55].

CE allows reproductive barriers to arise and can change their permeability without constraint, as the barriers themselves could evolve, allowing speciation to emerge as a property of the system. This contrasts with 'speciation' in AE, which exists only to increase population diversity[66,67].

### What we can gain: solving new problems

CE will allow algorithm designers to address new challenges. Potential target problems will tend to have the following features: they cannot be completely specified *a priori*; they are dependent on temporal variations; the information needed to solve them is apparently available, but it is not clear which data are relevant; and their solution requires autonomy and cannot presume on-demand human intervention (see BOX 1 for examples).

### Challenges

There are both social and technical barriers to realizing our vision of CE.

***Social challenges.*** The main difficulties will be psychological. CE practitioners must be willing to be patient and generous with resources. Patience will be necessary given the complexity and robustness of living systems compared with the simplicity and brittleness of engineered systems.

AE researchers will also have to be willing to sacrifice precise specifications and a full understanding of how individual components work. But CE will allow us to explore realistically complicated evolutionary scenarios at machine speeds, and to ask and address new, complex and open-ended questions.

***Technical challenges.*** CE will require new data and control structures to implement interacting hierarchies of information and processes. AE researchers have already taken some steps in this direction by creating genomes that encode rules for translating the genome[21](known as 'indirect representation'), and by using new module synthesis techniques; for example, one in which segments of the genome are segregated to evolve and be used as a single entity (known as automatically defined functions[8]).

CE will require considerable computational power. AE was born a half century ago, when computing power was at a premium, and many AE practitioners still insist that run-times be 'reasonable' on readily available machines. Because CE is more complex than existing AE implementations, reasonable run-times will require significantly more power. Fortunately, available power continues to grow, compounded by new ways of combining existing facilities.

Many of the evolutionary processes that we suggest should be incorporated into CE are still poorly understood. But our current understanding far exceeds what is typically incorporated into AE. Much as AE was able to ride the wave of technological progress over the past 50 years, CE should incorporate new biological knowledge as it becomes available, so riding the wave of advances in biology as well as that of ongoing technological progress.

### Conclusions

We recognize that our proposals are suggestive, rather than precise. Note also that CE will not be a magical elixir and there will always be specific optimization and design problems that can be solved without the complexity of CE.

One example of a question that CE could address in future is whether it is possible to construct a program that functions like an organism, with interacting software objects (analogues of cells) that collectively perform a global function such as providing operating system services (such as file management), with the ability to respond gracefully to demand and damage (analogously to homeostasis).

The need for such reliable, self-maintaining software is great and growing quickly. For example, IBM is investing heavily in traditional engineering efforts to define and build 'autonomic' software systems[68].

### Box 3 | Complicating the genotype–phenotype relationship

Most artificial evolution (AE) approaches use a simple, unidirectional map from genotype to phenotype. Computational evolution (CE) would enrich the translation from an evolving substrate (the 'genotype') into a configuration that is evaluated (the 'phenotype'). For example, a regulatory feedback mechanism could emerge if genotypic information were expressed only when other parts of the genotype have already been expressed. This would generate a dynamic system that would allow environmental interactions to intervene and stabilize self-organization, much more like natural evolution[69].

For instance, artificial regulatory networks[45,70,71] have recently become a field of active exploration, because they lend themselves to the sort of dynamics and side-effects necessary for the emergence of novelty. Other types of network and interaction are being explored as well[72,73].

Multiple interactions occur between the various steps that lead from genotype to phenotype: many regions of the genome, transcriptome and proteome affect each other, and all work together to determine individual functionality, which forms the basis for selection.

How would one promote these emergent interactions in CE? One way would be by deliberately encouraging interactions between system components. Instead of isolating functions, as is usual in an engineering setting, one would allow for smooth variation of many components and cross-talk between them. If a sort of regulatory network would determine the mapping between genotype and phenotype, there would be a combinatorial explosion of possible interactions[74,75]. Given such a sufficient number of degrees of freedom, it would always be possible to suppress undesirable effects by adding an extra inhibitor or co-opting an existing one. However, if an unintended slight change in a variable results in a beneficial innovation, CE would be able to select for the new unfolding functionality.

We argue that CE will be required to define and perfect such a system. Traditional software engineering breaks down problems into highly constrained interactions between modules, until the modules are simple enough to be implemented in computer code. But this approach abhors emergence and avoids surprises. There is also a natural ceiling in the program size achievable by a limited number of human programmers with traditional engineering.

Multicellularity, hierarchically defined expressions of the genome and massive parallelism will benefit a CE approach to designing an autonomic operating system. In living systems the genetic code is copied in full with each cell replication, and full organismal complexity emerges through physical interactions of cells and cell products. Similarly, a CE approach would allow the complexity of the software system to emerge from massively distributed interactions between hierarchically organized data and control structures, described by an evolving genome.

Pursuing our proposal, even imperfectly, will help to close the loop between biology and computing, for the benefit of both: CE could point to a new way of computing and a new way of understanding the living world, including how evolution works.

*Wolfgang Banzhaf is at the Department of Computer Science, Memorial University of Newfoundland, St John's, Newfoundland and Labrador A1B 3X5, Canada.*
*Guillaume Beslon and Virginie Lefort are at the Departement d'Informatique, INSA, 69621 Villeurbanne Cedex, France.*
*Steffen Christensen is at the School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario K1S 5B6, Canada.*
*James A. Foster is at the Department of Biological Sciences, University of Idaho, Moscow, Idaho 83844-3051, USA.*
*François Képès is part of the Epigenomics Project, Genopole, 523 Terrasses de l'Agora, 91000 Evry, France.*
*Julian F. Miller is at the Department of Electronics, University of York, York YO10 5DD, UK.*
*Miroslav Radman is at the Université de Paris-Descartes, Faculté de Médecine, INSERM Site Necker, U571, 156 Rue de Vaugirard, 75015 Paris, France, and the Mediterranean Institute for Life Sciences, Mestrovicevo setaliste bb, 2100 Split, Croatia.*
*Jeremy J. Ramsden is at the School of Applied Sciences, Cranfield University, Cranfield, Bedfordshire MK43 0AL, UK.*
*Correspondence to W.B., F.K. and J.A.F.*
*e-mails: banzhaf@cs.mun.ca; Francois.Kepes@epigenomique.genopole.fr; foster@uidaho.edu*

doi:10.1038/nrg1921

1. Foster, J. A. Evolutionary computation. *Nature Rev. Genet.* **2**, 428–436 (2001).
2. Fogel, D. (ed.) *Evolutionary Computation: the Fossil Record* (IEEE Press, New York, 1998).
3. Langdon, W. B. & Gustafson, S. Genetic programming and evolvable machines: five years of reviews. *Genet. Programming Evolvable Machines* **6**, 221–228 (2005).
4. Cotta, C. & Merelo, J. J. The complex network of evolutionary computation authors: an initial study. *Physics Abstracts* [online], < http://arxiv.org/abs/physics/0507196 > (2006).
5. Fogel, L. J., Owens, A. J. & Walsh, M. J. *Artificial Intelligence through Simulated Evolution* (Wiley, New York, 1966).
6. Holland, J. H. *Adaptation in Natural and Artificial Systems* 2nd edn (MIT Press, Massachusetts, 1993).
7. Rechenberg, I. *Evolutionsstrategie* 2nd edn (Frommann, Stuttgart, Germany, 1993) (in German).
8. Koza, J. *Genetic Programming* (MIT Press, Massachusetts, 1992).
9. Cage, P. G., Kroo, I. M. & Braun, R. D. Interplanetary trajectory optimization using a genetic algorithm. *J. Astronautical Sci.* **43**, 59–75 (1995).
10. Stoica, A., Klimeck, G., Salazar-Lazaro, C., Keymeulen, D. & Thakoor, A. Evolutionary design of electronic devices and circuits. *Proc. 1999 Congress Evol. Computation* 1271–1278 (IEEE Press, New York, 1999).
11. Barnum, H., Bernstein, H. J. & Spector, L. Quantum circuits for OR and AND of ORs. *J. Phys. A* **33**, 8047–8057 (2000).
12. Koza, J. *et al. Genetic Programming IV: Routine Human–Competitive Machine Intelligence* (Kluwer Academic, Massachusetts, 2003).
13. Koza, J., Keane, M., Yu, J., Bennett, F. H. III & Mydlowec, W. Automatic creation of human–competative programs and controllers by means of genetic programming. *Genet. Programming Evolvable Machines* **1**, 121–164 (2000).
14. Vaario, J. From evolutionary computation to computational evolution. *Informatica* **18**, 417–434 (1994).
15. Yeh, E.-C., Venkata, S. S. & Sumic, Z. Improved distribution system planning using computational evolution. *Proc. Power Industry Comp. App. Conf.* 530–536 (IEEE Press, Pittsburgh, 1995).
16. Judson, R. Computational evolution of a model polymer that folds to a specified target conformation. *Proc. Pacific Symp. Biocomputing* 423–437 (World Scientific, Singapore, 1996).
17. Morange, M. Gene function. *C.R. Acad. Sci. Paris* **323**, 1147–1153 (2000).
18. Ramsden, J. J. *Bioinformatics: An Introduction* (Kluwer Academic, Dordrecht, 2004).
19. Kapranov, P., Cawley, S. E. & Drenkow, J. Large-scale transcriptional activity in chromosomes 21 and 22. *Science* **296**, 916–919 (2002).
20. Gruau, F. Genetic synthesis of modular neural networks. *Proc. 5th Int. Conf. Genet. Algorithms* 318–325 (Morgan Kaufmann, San Francisco, 1993).
21. O'Neill, M. & Ryan, C. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language* (Kluwer Academic, Dordrecht, 2003).
22. Koza, J. R., Andre, D., Bennett, F. H. & Keane, M. *Genetic Programming 3: Darwinian Invention and Problem Solving* (Morgan Kaufman, San Francisco, 1999).
23. Keller, R. E. & Banzhaf, W. The evolution of genetic code in genetic programming, *Proc. Genet. Evol. Computation Conf.* 1077–1082 (Morgan Kaufmann, San Francisco, 1999).
24. Goldberg, D. E. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms* (Kluwer Academic, Dordrecht, 2002).
25. De Jong, K. *Evolutionary Computation: a Unified Approach* (MIT Press, Cambridge, Massachusetts, 2006).
26. Ramsden, J. J. Paracelsus: the measurable and the unmeasurable. *Psyche Problems Persp.* **4**, 52–88 (2004).
27. Ramsden, J. J. Computational aspects of consciousness. *Psyche Problems Persp.* **1**, 93–100 (2001).
28. Képès, F. Periodic transcriptional organization of the *E. coli* genome. *J. Mol. Biol.* **340**, 957–964 (2004).
29. Coen, E. *The Art of Genes* (Oxford Univ. Press, Oxford, 1999).
30. Conrad, M. in *A Half-Century Survey on the Universal Turing Machine* 285–307 (Oxford Univ. Press, New York, 1988).
31. Miller, J. F. & Downing, K. L. Evolution *in materio*: Looking beyond the silicon box. *Proc. NASA / DoD Conf. Evolvable Hardware* 167–176 (IEEE Computer Society, Virginia, 2002).
32. Harding, S. & Miller, J. F. Evolution *in materio*: A tone discriminator in liquid crystal. *Proc. Congress Evol. Comp. 2004* **2**, 1800–1807 (2004).
33. Dittrich, P., Ziegler, J. & Banzhaf, W. Artificial chemistries — a review. *Artificial Life* **7**, 225–275 (2001).
34. Sims, K. Evolving virtual creatures. *Proc. 21st Annu. Conf. Comp. Graphics Interactive Technol.* 15–22 (ACM Press, New York, 1994).
35. Pollack, J. B., Lipson H., Hornby, G. & Funes, P. Three generations of automatically designed robots. *Artificial Life* **7**, 215–223 (2001).
36. Rieffel, J. & Pollack, J. The emergence of ontogenic scaffolding in a stochastic development environment. *Lecture Notes Comp. Sci.* **3102**, 804–815 (Springer, Berlin, 2004).
37. Ray, T. An approach to the synthesis of life. *Proc. Artificial Life II* 371–408 (Addison-Wesley, Boston, 1991).
38. Adami, C. *An Introduction to Artificial Life* (Springer, Berlin, 1999).
39. Lones, M. A. & Tyrrell, A. Biomimetic representation with enzyme genetic programming. *Genet. Programming Evolvable Machines* **3**, 193–217 (2002).
40. Érdi, P. & Barna, G. Self-organizing mechanism for the formation of ordered neural mappings. *Biol. Cybern.* **51**, 93–101 (1984).
41. Luthi, P. O., Preiss, A., Chopard, B. & Ramsden, J. J. A cellular automaton model for neurogenesis in drosophila. *Physica D* **118**, 151–160 (1998).
42. McKinney, M. Heterochrony: Beyond words. *Paleobiology* **25**, 149–153 (1999).
43. Ramsden, J. J. in *Creatine Kinase and Brain Energy Metabolism* (eds Kekelidze, T. & Holtzmann, D.) 55–58 (IOS Press, Amsterdam, 2003).
44. Eggenberger-Hotz, P. Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. *Proc. Congress Evolutionary Computation* (IEEE Press, New York, 2003).
45. Kuo, P. D., Leier, A. & Banzhaf, W. Evolving dynamics in an artificial regulatory network model. *Lecture Notes Comp. Sci.* **3242**, 571–580 (Springer, Berlin, 2004).
46. Gould, S. J. & Vrba, E. Exaptation: a missing term in the science of form. *Paleobiology* **8**, 4–15 (1982).
47. Berenbrink, M., Koldkjaer, P., Kepp, O. & Cossins, A. R. Evolution of oxygen secretion in fishes and the emergence of a complex physiological system. *Science* **307**, 1752–1757 (2005).
48. Szathmáry, E. & Maynard-Smith, J. The major evolutionary transitions. *Nature* **374**, 227–232 (1995).
49. Hartwell, L. H., Hopfield, J. J., Leibler, S. & Murray, A. W. From molecular to modular cell biology. *Nature* **402**, C47–C52 (1999).
50. Simon, H. A. How complex are complex systems? *Proc. 1976 Biennial Meeting Philos. Sci. Assoc.* **2**, 507–522 (Edwards Bros, Ann Arbor, 1977).
51. Ashby, W. R. Requisite variety and its implications for the control of complex systems. *Cybernetica* **1**, 84–99 (1958).
52. Muotri, A. R. *et al.* Somatic mosaicism in neuronal precursor cells mediated by L1 retrotransposition. *Nature* **435**, 903–910 (2005).
53. Meysenburg, M. M., Hoelting, D., McElvain, D. & Foster, J. A. How random generator quality impacts genetic algorithm performance. *Proc. Genet. Evolutionary Computation Conf.* 480–483 (Morgan Kaufmann, San Francisco, 2002).
54. Altenberg, L. in *Advances in Genetic Programming* (ed. Kinnear, K. E.) 47–74 (MIT Press, Cambridge, Massachusetts, 1994).
55. Radman, M., Matic, I. & Taddei, F. Evolution of evolvability. *Ann. NY Acad. Sci.* **870**, 146–155 (1999).
56. Earl, D. & Deem, M. Evolvability is a selectable trait. *Proc. Natl Acad. Sci. USA* **101**, 11531–11536 (2004).
57. Dawkins, R. in *Artificial Life* (ed. Langton, C.) 201–220 (Addison Wesley, Boston, 1989).
58. Wagner, G. P. & Altenberg, L. Complex adaptations and the evolution of evolvability. *Evolution* **50**, 967–976 (1996).
59. Kirschner, M. W. & Gerhard, J. C. *The Plausibility of Life* (Yale Univ. Press, New Haven, 2005).
60. Shapiro, J. A. A 21st century view of evolution: genome system architecture, repetitive DNA, and natural genetic engineering. *Gene* **345**, 91–100 (2005).

61. Ramsden, J. J. & Vohradský, J. Zipf-like behavior in procaryotic protein expression. *Phys. Rev. E* **58**, 7777–7780 (1998).
62. Mayr, E. What is a species and what is not? *Philos. Sci.* **63**, 262–277 (1996).
63. Avise, J. *Phylogeography: the History and Formation of Species* (Harvard Univ. Press, Massachusetts, 2000).
64. Arber, W. in *Frontiers in Biology* (eds Chou, C.-H. & Shao, K.-T.) 19–24 (Academia Sinica, Taipei, 1998).
65. Margulis, L. & Sagan, D. *Acquiring Genomes: a Theory of the Origins of Species* (Basic Books, New York, 2002).
66. Deb, K. & Goldberg, D. An investigation of niche and species formation in genetic function optimization. *Proc. 3rd Int. Conf. Genet. Algorithms* 42–50 (Morgan Kaufmann, San Francisco, 1989).
67. Fonseca, C. M. & Fleming, P. J. Multiobjective genetic algorithms made easy: selection, sharing, and mating restriction. *Proc. 1st Int. Conf. Genet. Algorithms Engineering Systems Innovations Appl.* 45–52 (IEEE Press, New York, 1995).
68. International Business Machines. Autonomic computing. *IBM Research | Autonomic Computing* [online], < http://www.research.ibm.com/autonomic/ > (2006).
69. Ridley, M. *Evolution* (Blackwell Science, Oxford, 1996).
70. Reil, T. Dynamics of gene expression in an artificial genome-implications for biological and artificial ontogeny. *Adv. Artificial Life — Proc. 5th European Conf. Artificial Life* **1674**, 457–466 (1999).
71. Watson, J., Geard, N. & Wiles, J. Towards more biological mutation operators in gene regulation studies. *BioSystems* **76**, 239–248 (2004).
72. Volkert, L. G. & M. Conrad. The role of weak interactions in biological systems: the dual dynamics model. *J. Theo. Bio.* **193**, 287–306 (1998).
73. Bongard, J. Evolving modular genetic regulatory networks. *Proc. 2002 Congress Evolutionary Computation* **2**, 1872–1877 (IEEE Press, New York, 2002).
74. Eggenberger, P. Evolving morphologies of simulated 3D organisms based on differential gene expression. *Proc. 4th European Conf. Artificial Life* 205–213 (MIT Press, Cambridge, Massachusetts, 1997).
75. Volkert, L. G. Enhancing evolvability with mutation buffering mediated through multiple weak interactions. *Biosystems* **69**, 127–142 (2003).

**Acknowledgements**
We thank the people at Genopole Recherche, Évry, France, for generously sponsoring the meeting that initiated this paper. We also thank the anonymous referees for helpful suggestions.

**Competing interests statement**
The authors declare no competing financial interests.

## FURTHER INFORMATION

**Authors' web site:** http://www.computational-evolution.org
**François Képès's homepage:** http://stat.genopole.cnrs.fr/~kepes
**Guillaume Beslon's homepage:** http://prisma.insa-lyon.fr/~gbeslon
**James Foster's homepage:** http://www.ibest.uidaho.edu/~foster
**Jeremy Ramsden's homepage:** http://www.cranfield.ac.uk/~wajr
**Julian Miller's homepage:** http://www.elec.york.ac.uk/intsys/users/jfm7
**Miroslav Radman's homepage:** http://www.necker.fr/tamara
**Wolfgang Banzhaf's homepage:** http://www.cs.mun.ca/~banzhaf
**Access to this links box is available online.**