Published online: July 30, 2018

EMBO reports

Science & Society

Will biologists become computer scientists?

A truly interdisciplinary effort by computer scientists and biologists to understand how cells process information may yield new insights for both fields

Anne Condon[1], Hélène Kirchner[2], Damien Larivière[3], Wallace Marshall[4], Vincent Noireaux[5], Tsvi Tlusty[6,7] & Eric Fourmentin[3]

The idea that living systems could be understood and described as information-processing systems has been around even before the first computers were built. From Alan Turing's considerable paper in 1936 to Erwin Schrödinger's work in 1944 and John von Neumann's work in 1948 [1], many scientists pondered about information storage and the possible existence of a logical processor within living cells. The discovery of the double-helical structure of DNA in 1953 provided the material basis for these intuitions as it finally revealed how cells store inheritable information in a "digital" format. The recent success of *genome transplantation* experiments into recipient host cells [2]—akin to transferring software to another computer—further strengthened the hypothesis that living cells can be regarded as *Turing Machines,* as was suggested by Sydney Brenner [3] (see Sidebar 1 for a glossary and Sidebar 2 for further readings).

> *"... many scientists pondered about information storage and the possible existence of a logical processor within living cells."*

> *"François Jacob and Jacques Monod were among the first biologists to understand gene expression as an algorithm."*

In light of these and other experimental results that would support the hypothesis that some parts of living systems could be understood as information-processing machines, the Fourmentin-Guilbert Scientific Foundation invited international scholars from the life sciences, computer sciences and physical sciences (see Sidebar 3) to the I2CELL (from Information to Cells) seminar in February 2018 near Oxford, UK, to discuss and identify new research areas. Over 3 days, they debated on a broad range of subjects from computation, information handling, algorithms, robotics and viruses (of the digital and biological varieties) to explore analogies between cells and computers that could inspire new research, while keeping a critical approach to the benefits of similarities. This article summarizes and analyses the presentations and debates.

Biomolecular computation

A crucial question is what does computation mean in the context of living systems? For a start, it means that cells host processes that manipulate symbolic information according to logical rules. The most obvious example is the genetic networks, which have many attributes associated with computing [4]. In fact, the extent of computations that cells and viruses perform seems to be very large. Eukaryotic cells, for example, make a decision of whether or not to divide by performing a "majority voting"-based computation. A similar process takes place during infection of a bacterium by a phage. The decision between lysis (reproduction of the phage to produce new viral particles at the expense of the host cell) and lysogeny (a Trojan horse-like state whereby the phage hides its genome in the host genome) is taken as an unanimous vote of the viral particles in the infected cell. Although this decision—as many other "decisions" of living systems—seems to be a stochastic event, it could be framed in a broader context, thereby yielding to a better understanding of how living systems make decisions if we could reveal the underlying *algorithms*.

Formalizing cellular algorithms

François Jacob and Jacques Monod were among the first biologists to understand gene expression as an algorithm [5]. The *lac* operon they discovered is a genetic

1 Department of Computer Science, University of British Columbia, Vancouver, BC, Canada
2 Inria Bordeaux – Sud-Ouest, Talence, France
3 Fourmentin-Guilbert Scientific Foundation, Noisy-Le-Grand, France
4 Department of Biochemistry and Biophysics, Center for Cellular Construction, University of California San Francisco, San Francisco, CA, USA
5 School of Physics and Astronomy, University of Minnesota, Minneapolis, MN, USA. E-mail: noireaux@umn.edu
6 Center for Soft and Living Matter, Institute for Basic Science, Ulsan, Korea
7 Department of Physics, Ulsan National Institute of Science and Technology, Ulsan, Korea

DOI 10.15252/embr.201846628 | EMBO Reports (2018) e46628

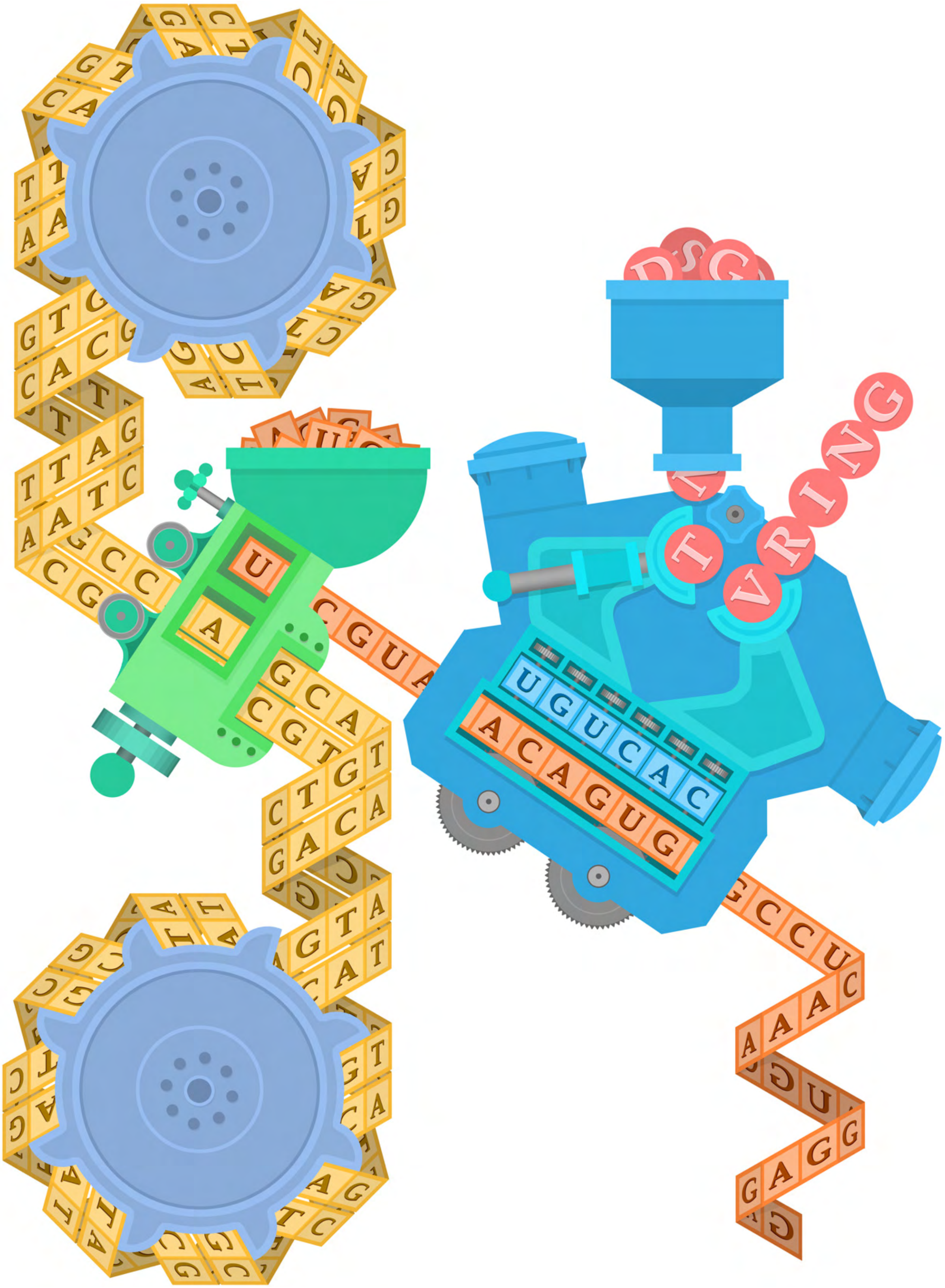

Fig 1. Steampunk illustration of a Turing Biomachine by David S. Goodsell (TSRI).
The translation and transcription machinery in a cell to convert information from DNA into proteins can be viewed and analysed as a Turing Machine. © Fourmentin-Guilbert Scientific Foundation.

Published online: July 30, 2018

Sidebar 1: Glossary

Algorithm
A description of a method to solve a problem in terms of elementary, precise operations. When expressed in a particular language, the algorithm is called a program. Cellular processes to make macromolecules are algorithmic in the form "begin, do: [if Condition then Action, check Control Points, repeat], end".

Automata
Abstract models of machines to perform computations from an input by moving through a series of intermediate states. When an automaton sees a symbol as input, it changes to another state according to an instruction (given by a transition function). The stored-program computer and the living cell are two concrete realizations of an automaton.

Turing Machine
An abstract machine with a virtual head to read and write symbols structured in words (sequences) from an infinite tape, focusing on one symbol at a time. A ribosome translating the information from a messenger RNA into a protein can be described as a Turing Machine, with striking physical similarity.

State machine
There is a whole hierarchy of state machines. The simplest one is a Finite State Machine (FSM) or Finite State Automaton (FSA). It has a finite number of states with an initial state, and transitions triggered by conditions (inputs). Apart from the states reflecting its current situation, the FSM has no mechanism for remembering past operations. Going up the hierarchy ladder, more sophisticated state machines are augmented with an increasingly versatile storage facility. The Turing Machine is on top with no restriction on its number of states. The information treated by a FSM resides in its states and its inputs. In the case of a Turing Machine, the information is also stored as symbols on the tape.

Maxwell's demon
James Clerk Maxwell originally conceived his thought experiment as a hypothetical being able to capture information about a system, thus reducing entropy, apparently against the standard laws of thermodynamics. Later on, his demon was "exorcised" when theoretical works suggested that it cannot use the information gained on the system without memorizing it and that the erasure of this information to reset the measurement device comes at an energetic cost, so as to preserve the laws of physics. Today, physicists and chemists try to prove the physical dimension of information by quantifying the relationship between information and energy—as did Einstein by unifying matter and energy—and try to implement concrete information-driven devices.

Genome transplantation
Experimental approach that consists of replacing entirely the genetic program of a bacterial cell by a synthetic new genetic program. The fact that the cells readily express the new chromosome, in the case of Mycoplasma [2], shows that the genetic program is separated from the cellular machine like in a computer and comes up as another proof of concept of the cell as a Turing Machine. Like in a computer, in which a program does not run if it is not properly recognized by the machine, one cannot expect any genome transplantation to be productive.

Kinetic proofreading
The proper functioning of protein synthesis depends on the ability of the ribosome to decode the messenger RNA with high fidelity. When a ribosome incorporates amino acids in a ratchet-like manner, it selects one wrong amino acid in 10,000. It achieves this low error rate thanks to specific proteins that, acting like Maxwell's demon, test (proofread) if the amino acid presented to the ribosome is the correct one.

system that allows *E. coli* cells to switch between lactose and glucose as a food source by executing a subroutine that uses "if–then" conditional statements common to rule-based programming languages. Such observations could be formalized more systematically by exploiting concepts from computer science, helping to answer fundamental questions about how living systems sense, process and react to information.

Computer sciences provide elaborate tools and methods to study and formulate algorithms. The *automata* model deserves particular attention as living cells can be considered as a material implementation of *state machines* or as housing such machines that transcribe, translate and replicate the genome (Fig 1). But only a few cellular processes have so far been described as state machines. Stahl and Goheen, for example, use a Turing Machine to model the "algorithmic enzymes" involved in the synthesis of mRNA and proteins [6]. Robert Landick and his colleagues mapped the whole transcription process onto a Turing Machine [7] (see Sidebar 2 for other examples).

> *"... living cells can be considered as a material implementation of state machines or as housing such machines that transcribe, translate and replicate the genome..."*

During I2CELL, the participants discussed other systems and observations—including, for instance, centrioles that play a crucial role in cell division—that could be analysed so as to classify cellular states and then explore the rules governing transitions between those states. Such approaches place cell biology into a *state machine* framework and, more generally, demonstrate that living systems could indeed be understood and analysed as housing information-processing devices.

Information storage, interpretation and measurement

Information storage is a key component of a Turing Machine or state machine. While DNA is a long-term memory to store and transmit information over generations, short-term memories also exist within the cell. The Turing Machine formalism suggests that proteins and RNA may act as a transient memory, together with other epigenetic information that is not coded in genes. The information contained in computer programs is subject to successive interpretations and translations from higher-level abstractions down to hardware instructions. In a living cell, similar computational processes can be observed at different levels of abstraction ranging from phenotype through genetic programs down to the underlying molecular interactions [8].

> *"The Turing Machine formalism suggests that proteins and RNA may act as transient memory, together with other epigenetic information that is not coded in genes."*

Further analogies might be useful to understand information handling in cells. In particular, the existence of a biological

Published online: July 30, 2018

Sidebar 2: Further reading

For a deeper understanding of how historically analogies between living cells and computers have been made, see videos by Sydney Brenner:
https://www.webofstories.com/play/sydney.brenner/45
https://www.webofstories.com/play/sydney.brenner/48

For a detailed analogy between living cells and computers, see:

- Danchin A (2008) Bacteria as computers making computers. *FEMS Microbiology Reviews* 33: 3–26

For a deeper analysis of various state machines, see figure 3 in:

- Benenson Y (2012) Biomolecular computing systems: principles, progress and potential. *Nat Rev Genetics* 13: 455–468

For a better understanding of the links between languages and state machines and how they are powerful tools for expressing biological messages, see:

- Searls DB (2002) The language of genes. *Nature* 420: 211–217

For a deeper understanding of biological computations and algorithms and examples thereof, see:

- Navlakha S, Bar-Joseph Z (2011) Algorithms in nature: the convergence of systems biology and computational thinking. *Mol Syst Biol* 7: 546
- Cardelli L, Hernansaiz-Ballesteros RD, Dalchau N, Csikasz-Nagy A (2017) Efficient switches in biology and computer science. *Plos Comput Biol* 13: e1005100
- Zeng L, Skinner SO, Zong C, Sippy J, Feiss M, Golding I (2010) Decision making at a subcellular level determines the outcome of bacteriophage infection. *Cell* 141: 682–691

For a deeper incursion into the relationships between information and energy and the possibility to create molecular devices making use of information, see:

- Serreli V, Lee CF, Kay ER, Leigh DA (2007) A molecular information ratchet. *Nature* 445: 523–527
- Lutz E, Ciliberto S (2015) Information: from Maxwell's demon to Landauer's eraser. *Physics Today* 68: 30

For an application of Maxwell's demon to biological problems, see:

- Binder P, Danchin A (2011) Life's demons: information and order in biology. What subcellular machines gather and process the information necessary to sustain life? *EMBO Reports* 12: 495–499

Chromatin reading and chromatin writing, transcription, bacterial chemotaxis or DNA recombination enzyme assembly have been framed as a Turing Machine. For details, see:

- Bryant B (2012) Chromatin computation. *PLoS One* 7: e35703
- Lan G, Tu Y (2016) Information processing in bacteria: memory, computation, and statistical physics: a key issues review. *Reports on progress in physics. Physical Society* 79: 052601
- Bar-Ziv R, Tlusty T, Libchaber A (2002) Protein-DNA computation by stochastic assembly cascade. *Proc Natl Acad Sci U S A* 99: 11589–11592

Computer scientist Charles Bennett compared the RNA polymerase with the read/write head of a Turing Machine and came up with considerations on energy dissipation during proofreading that shed new light on the physical dimension of the concept of information as a fifth category of Nature. For details, see:

- Bennett CH (1973) Logical reversibility of computation. *Ibm J Res Dev* 17: 525–532

For a deeper understanding of evolutionary algorithmics and robotics, and the proposition that real robots or hardware models can be a testing ground for biologists to investigate questions and hypothesis, for instance reproducing the selective pressure driving the evolution of biological organisms, see:

- Eiben AE, Smith, JE (2003) *Introduction to evolutionary computing*. Berlin: Springer
- Eiben AE, Smith, JE (2015) From evolutionary computation to the evolution of things. *Nature* 521: 476–482

counterpart to *Maxwell's demon*-like mechanisms could help to explain important observations in biology such as asymmetric cell division. Here, most of the damaged and old molecular components remain in the "mother" cell while the "daughter" receives the "younger" proteins and organelles. How does the cell achieve such asymmetry during division? One hypothesis suggests that proteins are filtered by *Maxwell's demons* that measure the state of encountered components—damaged or aged for instance—memorize this information, produce an action accordingly and reset their memory. *Kinetic proofreading*, which is an important mechanism to prevent errors during protein synthesis, can be also viewed as a process akin to *Maxwell's demon*. Such arguments would stress the important role of the concept of information as a measurable quantity.

Beyond the concepts: how far can we draw an analogy between a living cell and a computer?

The architecture and programming of man-made computers should stimulate new questions for biologists: if the analogy holds, what does the cellular hardware do? How can we dissect biological functions from metabolic pathways to complex behaviours into their essential parts? For historical reasons, molecular biology has relied on a bottom-up approach from gene to protein and to function. Instead, a top-down approach or functional analysis [9] to identify and formulate the master functions of life and their underlying secondary functions, and then the objects associated with them, could provide a broader understanding of the essential processes in cells. Thus, current research on creating a genome coding for a "minimal set of functions" rather than a minimal set of genes could help to identify the still unknown functions of many essential genes.

Most computers run an Operating System (OS) with a reduced set of instructions that provides a common interface between hardware resources and applications. While the genome is a true program in the sense that it contains instructions and data, it is not clear yet whether there is also a cellular OS. Nevertheless, the metaphor could help to identify some of the homeostatic functions within a cell that would be part of a minimal genome.

Another analogy between computers and cells is suggested by *genome transplantation* experiments by which a recipient host cell's genome is replaced by a synthetic genome assembled exogenously. Beyond the essential compatibility of the transcription and translation machinery to "read" the new genome, it might need additional "boot programs" in order to restart the regulatory and metabolic functions encoded in the new genome and the host cell.

Like living systems, computers age; more precisely, transistors, the elementary building blocks of electronic circuits, age. This ageing decreases the switching speed over time and thereby overall performance. Processors and memory circuits are therefore equipped with components that

Published online: July 30, 2018

Sidebar 3: I2CELL participants and topics

Hugo Aguilaniu, Instituto Serrapilheira, Brasil
David Bikard, Institut Pasteur, France
Holger Breithaupt, EMBO, Germany
Anne Condon, University of British Columbia, Canada (in charge of the daily summaries)
Antoine Danchin, Institute of Cardiometabolism and Nutrition, Hôpital de la Pitié-Salpêtrière, France
Gilles Dowek, Inria/ENS Cachan, France
Agosto Eiben, Vrije Universiteit Amsterdam, The Netherlands
Eric Fourmentin, Fourmentin-Guilbert Scientific Foundation, France (organizer)
John Glass, J. Craig Venter Institute, USA
Ido Golding, Baylor College of Medicine, USA
Philipp Holliger, MRC, Cambridge University, UK
Paul Jardine, University of Minnesota, USA
Hélène Kirchner, Inria, France (organizer)
Damien Larivière, Fourmentin-Guilbert Scientific Foundation, France (organizer)
Carole Lartigue, Inra, France
Albert Libchaber, The Rockefeller University, USA
Jean-Yves Marion, IUF and Lorraine University, France
Wallace Marshall, UCSF, USA (in charge of the daily summaries)
Anthony Maxwell, John Innes Center, UK
Jean-Baptiste Mouret, Inria, France
Vincent Noireaux, University of Minnesota, USA (organizer)
Jordan Pollack, Brandeis University, USA
Olivier Sentieys, Inria/University of Rennes, France
Tsvi Tlusty, UNIST, Korea (in charge of the daily summaries)
Christoph Zechner, Max Planck Institute of Molecular Cell Biology and Genetics, Germany

The I2CELL seminar was organized into five sessions, each of which was a talk by a biologist, a physicist and a computer scientist followed by a debate.
Session 1: Cells as Turing Machines?
Session 2: Algorithm-driven cellular processes
Session 3: Where Operating System-like functions could be found?
Session 4: How far are we from making a self-replicating and self-reproducing machine?
Session 5: How to contain virus?

monitor transistor performance and progressively slow down the speed of computing to increase their lifespan. Do similar controls exist in cells that monitor the age of cellular components and adjust the speed of biological processors (RNA polymerases, ribosomes)? It might be valuable, when searching for biological *Maxwell's demons*, to identify protein networks that measure the intrinsic ageing of proteins—which manifests through chemical alterations—and that slow down the speed of transcription and translation to increase the cell's lifespan or that eventually trigger apoptosis or senescence to discard of or silence an aged cell. If it turns out that biological *Maxwell's demons* are central for renovating the cellular protein synthesis machinery or to synthesize a young cell, it would allow biologists to revisit ageing theories accordingly.

Finally, robotics is also exploiting concepts from nature and evolution to develop self-evolving non-organic systems with the ability to explore, sense and process information about their environment. Though this work is at early stages, robots should eventually become capable of autonomous evolution to adapt their functions and programming to their environment. Robots are already able to learn, sometimes from scratch, to improve their movement for instance, or to adapt their capabilities in case of degradation. Overall, there is a rich history of evolutionary computation and robotics that borrows ideas and inspiration from biology to design new machinery and software [10].

Conclusion

While cells and computers are made of very different materials—carbon versus silicon—both are nonetheless information-processing systems. The living world has for long been a source of inspiration for computer scientists and has led to the development of neural networks, evolutionary algorithms or self-learning systems. Many insights from living systems are still to be explored, for example to decrease the energetic cost of computation, to make artificial systems more autonomous, to design a computer immune system or to allow computers to benefit from viral infection.

Vice versa, there is also a lot to gain for biologists from ideas and concepts in computer science. Synthetic biology in particular is already benefiting in its attempts to create artificial cells designed to perform a particular task. While synthetic biology has a large inventory of available "parts"–metabolic and regulatory genes that can be combined into circuits—and the software program, that is the genome, can be tailored to specifications, it still remains impossible to bootstrap a cell from a set of molecules. Like computers, living cells are synthesized by algorithmic machines based on a construction scheme. Computer sciences could help to identify key characteristics of such processes and therefore allow biologists to better understand how the machinery of the cell evolved.

> *"There is a lot to gain for biologists from ideas and concepts in computer science."*

The Fourmentin-Guilbert Scientific Foundation has created the I2CELL Seed Award to emphasize the importance of this interdisciplinary field and to stimulate experimentally relevant ideas that exploit the concept of information in biology (www.i2cell.science).

Acknowledgements

The organizers are grateful to the 21 participants listed in sidebar 3, with a particular mention to Antoine Danchin for inspiring ideas during the preparation of the meeting. The Fourmentin-Guilbert Foundation is grateful to Hélène Kirchner and Vincent Noireaux for their invaluable contribution to the organization of I2CELL, their chairmanship and their help after the meeting to make the most of it.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Cobb M (2013) 1953: when genes became "information". *Cell* 153: 503–506
2. Lartigue C, Glass JI, Alperovich N, Pieper R, Parmar PP, Hutchison CA III, Smith HO, Venter JC (2007) Genome transplantation in

bacteria: changing one species to another. *Science* 317: 632–638

3. Brenner S (2012) Turing centenary: life's code script. *Nature* 482: 461
4. McAdams HH, Arkin A (1998) Simulation of prokaryotic genetic circuits. *Annu Rev Biophys Biomol Struct* 27: 199–224
5. Jacob F, Monod J (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J Mol Biol* 3: 318–356
6. Stahl WR, Goheen HE (1963) Molecular algorithms. *J Theor Biol* 5: 266–287
7. Mooney RA, Artsimovitch I, Landick R (1998) Information processing by RNA polymerase: recognition of regulatory signals during RNA chain elongation. *J Bacteriol* 180: 3265–3275
8. Condon A, Harel D, Kok NJ, Salomaa A, Winfree E (2009) *Algorithmic bioprocesses*. Berlin: Springer
9. Danchin A, Sekowska A, Noria S (2018) Functional requirements in the program and the cell chassis for next-generation synthetic biology. In *Synthetic Biology: Parts, Devices and Applications*, Smolke C (ed.), 1st edn. pp. 81–106. Weinheim: Wiley-VCH Verlag GmbH & Co. KGaA
10. Lipson H, Pollack JB (2000) Automatic design and manufacture of robotic lifeforms. *Nature* 406: 974–978