CELLULAR COMPUTING: TOWARDS AN ARTIFICIAL CELL

R. Lahoz-Beltra

***Abstract**: At present most point of views are in agreement with the idea that the similarity between cells and computers is a useful metaphor from which to obtain powerful predictions about life. In this paper we suggest that the analogy between computers and cells should be carefully reviewed when creating a silico artificial cell or whole-cell simulation, avoiding some common misconceptions derived from Cybernetics and the study of biological information processing based on a 'hardware + software' dualism.*

***Keywords**: artificial cell, minimal cell, virtual cell modelling and cellular computer models*

***ACM Classification Keywords**: I.6 Simulation and Modeling*

Introduction

Traditionally those sciences devoted to design artificial creations of biological systems, for instance Artificial Life as well as Artificial Intelligence, are based on the Cybernetics assumption that most biological entities can be represented as machines. One of the most interesting challenges is the creation of an artificial cell. An artificial cell is a minimal cell or *in silico* 'machine' composed by artificial parts, either wetware elements or software subroutines, respectively. The first wetware artificial cell was authored by T. Chang [Chang and Poznansky, 1968] consisting of a vesicle with a wall membrane and enzymes (Figure 1). Following, several other attempts were conducted creating artificial cells [Hotani et al., 1992; Noireaux et al., 2011] but far away of a fully operational cell. In 2010 Craig Venter successfully synthesized 'Synthia', a bacterial cell controlled by a chemically synthesized genome. On the other hand, M. Tomita [Tomita et al., 1999] designed E-Cell (Figure 2), one of the first *in silico* simulated artificial cell projects.

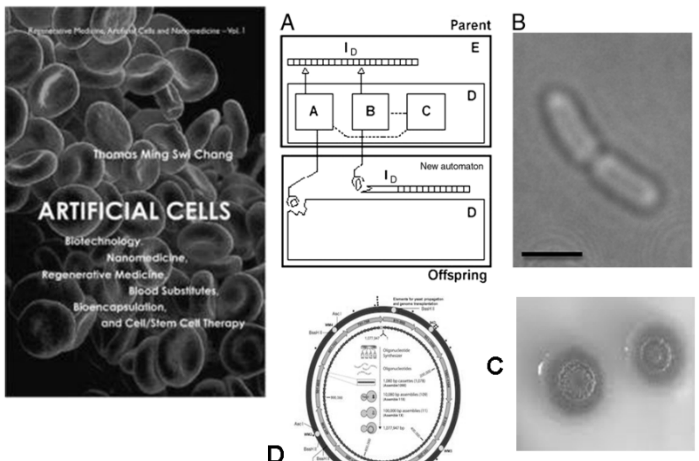
Thomas Ming Swi Chang
ARTIFICIAL CELLS
Biotechnology, Nanomedicine, Regenerative Medicine, Blood Substitutes, Bioencapsulation, and Cell/Stem Cell Therapy
A
Parent
E
A
B
C
D
New automaton
D
Offspring
B
C
D

FIGURE 1.- Artificial cells. (A, B) Von Neumann's scheme of cell-reproducing automata [Noireaux et al., 2011]. (C, D) Craig Venter's 'Synthia' and genome

One of the misleading assumptions applied to biological cells is the parallelism between cells and von Neumann architecture used to design computers [Bray, 1990; Sipper et al., 1997a, 1997b]. Usually terms like *CytoComputation* introduced by Paton [Paton, 1994] or the notion of computational architecture applied to cellular processing show the impact of Cybernetics modelling cells and cellular systems. Furthermore, this influence becomes apparent from the classical question about 'what is life?' [Klyce, 1999] to those conceptions derived from Artificial Life. For instance, the notion of 'A-Life avatar cell', thus models in which a cell is created in a computer memory with software. At present, the analogy between cells and computers has been accepted as a main assumption in other fields. For instance, in Medicine acupuncture functions through the Yin and Yang component of the cellular mediator, that is cyclic AMP and cyclic GMP, which activates the autonomic nervous feedback loop. In agreement with Kim [Kim, 1981] a computer works through Yin (0 binary digit) and Yang (1 binary digit) components which combinations represent symbols through a feedback loop of a central memory storage.

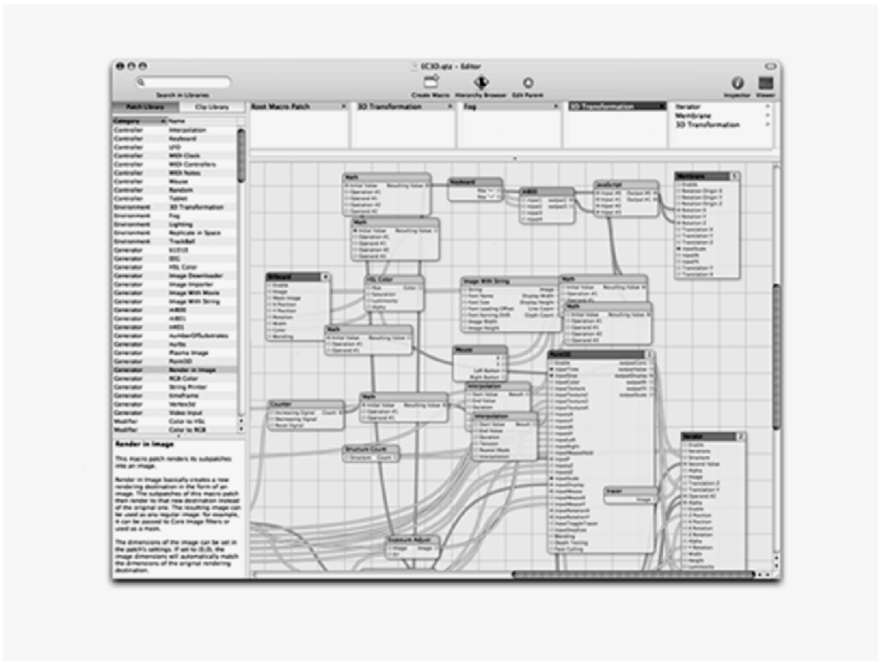

FIGURE 2. A whole-cell simulation model with E-Cell software

In this paper we suggest that the analogy between computers and cells should be carefully reviewed avoiding some common misconceptions derived from the (i) Cybernetics assumption that considers biological entities (i.e. cells, organs, etc.) as machines as well as (ii) the study of biological information processing based on a 'hardware + software' dualism that is a 'false dualism' when it is ingenuously applied in the analysis of biological systems. The above misleading assumptions are usual when creating *in silico* artificial cells or whole-cell simulations.

For instance, if DNA is the program - thus, the source code in a high level language - then are proteins the binary code? Furthermore, since proteins perform their task after their folding in a specific environment, i.e. cytoplasm, does the cytoplasm plays the role of the operating system in a computer? Which is the resemblance between protein biosynthesis and a compiler program? Is the Ray's TIERRA simulation [Ray, 1991] an analogy (Figure 3) to the Cambrian explosion fine enough and under what conditions?

Is cell circuitry, i.e. metabolic pathways, analogous to electronic circuits? [Lahoz-Beltra, 2001, Di Paola et al., 2004] Are biological molecules, i.e. proteins and enzymes, finite state machines? [Lahoz-Beltra, 1997] Are biological molecules, molecular automata with the capability to perform Boolean operations? [Lahoz-Beltra et al., 1993] These and other questions are critically analyzed in the present paper concluding that there is a 'likeness' between cells and computers but is this likeness between cells and computers a *metaphor* or it just arises from a real *similarity* between both systems?

The Analogy between Software Compilation and Biosynthesis of Proteins

Several analogies between computers and cells could be proposed considering that the main differences between both systems are given by their origin, hardware, software and the regulation of program execution. However, between cells and computers there are several analogies such as the (i) the linear topology of the program, (ii) the execution of the program is based on the translation of code, (iii) program code uses a few symbols, such as A, T, G, C in cells, and 0, 1 in computers, (iv) both systems, cells and computers, are susceptible of 'virus infection'.

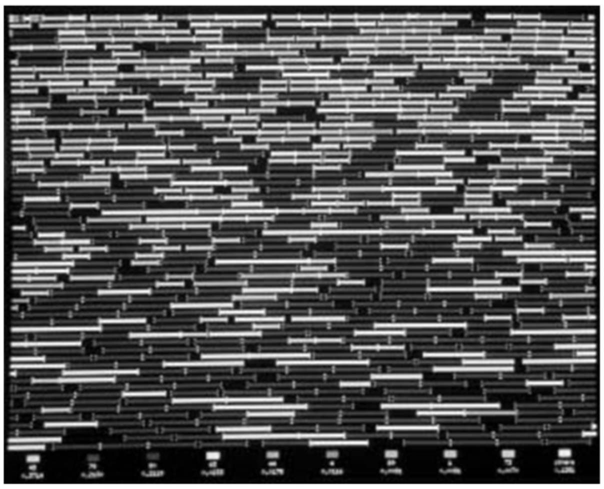

FIGURE 3. Thomas S. Ray's TIERRA simulator [Ray, 1991]. TIERRA is an example of an artificial life model or metaphor where evolvable computer programs can be considered as digital organisms which compete for energy (CPU time) and resources (main memory).

Our discussion about the likeness between cells and computers is illustrated (Figure 4) with the analysis of above analogy (ii).

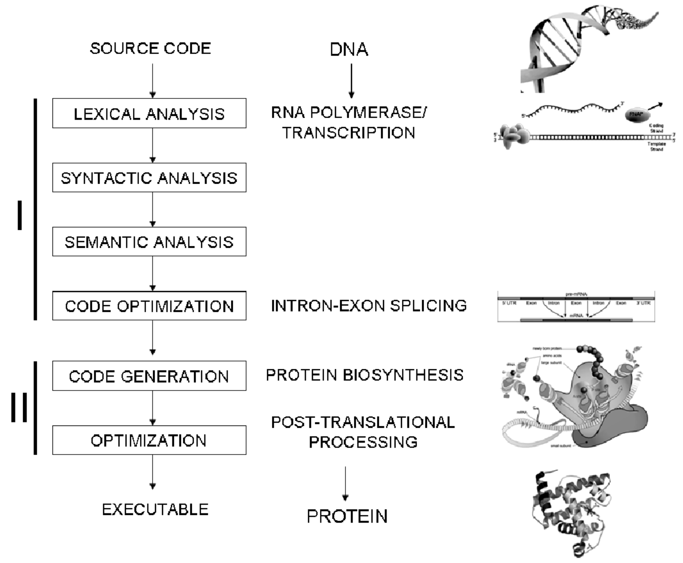

FIGURE 4. Analogy between cells and computers (for explanation see text)

The similarity between software compilation and the biosynthesis of proteins could be analyzed considering the following phases during the translation process:

(a) Decoding the source: DNA

Source code

Since DNA macromolecule is a sequence of four different subunits (A, T, G, C), the programming language in genetic terms is very simple consisting of A, T, G, C sequences of variable length. Program structures such as conditional if-then expressions, loops and sequential expressions (i.e. arithmetic expressions) are not apparently present in DNA, thus in the genetic source code.

Source analysis

The enzymatic complex RNA polymerase reads the DNA code in one-way direction (from the 5' end to 3') bearing a resemblance with a compiler during the lexical analysis phase. In such phase a compiler scans a source code from left to right, character per character. Furthermore, some DNA segments could be like arithmetic operators (+, -, *, /) or functions (i.e. exp(x), ln(x), Sin(x)) in a programming language but in such a case only understandable to cells promoting, repressing or terminating gene transcription. For instance, TATA box, -35 region, pause sites rich in GC and rho-dependent sequences could be the biological counterpart of above operators and functions but their representation may vary according to the compiler in use, thus the RNA polymerase complex which sets the parsing and syntax rules. In compiler terms, type checking is a strict process since the transcription from DNA to RNA is a precise and enzymatically controlled process where the semantic rules are usually respected and errors are not theoretically allowed (e.g. promoters cannot have a termination function).

Intermediate optimization

Most RNA resulting from DNA transcription contains both introns and exons. Through a complex enzymatic processing known as splicing introns are eliminated and exons are bound in a sequence. The final sequence form is the mature and optimized sequence of mRNA (messenger RNA). Also, further modifications may occur through nucleotide covalent changes and addition of caps or poly-A sequences. In the realms of compiler, this phase corresponds to an intermediate code generation and optimization where a simple and easily translatable source code representation is generated and unnecessary information is eliminated. It is interesting to note that the splicing event is relatively complex just as the compiler optimization phase is generally slow.

(b) Generating the active product: protein

Analysis of the optimizes source

Unlike a compilation process which at this step should generate the object code, the mature mRNA must be translated by ribosomes to a polypeptide or protein according to a dictionary or codon triplet table. DNA transcription as well as RNA translation comprises a lexical, syntactic and semantic analysis. In this case, A, G, C and U nucleotides must form three letter words or triplets (codons) which are associated through a degenerative translation table to specific amino acids (protein subunits) or to translation initiation or termination codons. Such a degenerative nature of codons counterbalances the effect of mutations which would form structurally or functionally different proteins. Likewise, a software program can be written in different ways performing all the programs the same task (e.g. 'while' and 'for' loops are different structures but both have similar function).

Synthesis

The tRNA-ribosomal system produces the protein sequence (primary structure) resembling a compiler during object code generation. Afterwards, specific enzymes (i.e. proteases) may modify the protein which could be considered as object code optimization during the compilation process. Finally, even after the biosynthetic process is finished a protein may be inactive until the protein is under the right conditions (pH, temperature, peptide cleavage, etc.). For this reason, the environment (i.e cytoplasm, membrane, etc.) in which proteins are embedded could bear a resemblance with the operating system which allows the compiler to work. As a result, functional proteins or enzymes could be considered as executable programs but proteins are able to carry out their function only under the right environment, just like a program compiled under a specific operating systems.

The Computational View of Cell: Metaphor *versus* Similarity

At present most point of views are in agreement with the idea that the similarity between cells and computers is a useful *metaphor* from which to obtain powerful predictions about life. In agreement with David Baltimore [Baltimore, 2011] *modern biology is a science of information* but people is not satisfied with the computer code metaphor of DNA. According to Baltimore there are two main reasons: (i) the meaning of a computer code (with only 2 possible symbols at each position, 0 or 1) is not a common knowledge and because (ii) the metaphor does not communicate the richness of coding systems buried in a monotonous string of symbols. For instance, the dynamic behaviour of cellular automata in computer simulations (i.e. propagation, interaction, storing, computing information, self-organization, chaos) bear a strong resemblance to those behaviours shown by biomolecules in the interior of real cells [Lahoz-Beltra, 1998]. Molecular automata, thus biomolecules simulated as finite automata, are able to capture the biological functionality exhibited by biomolecules inside cells [Lahoz-Beltra, 1997]. In consequence, such approach represents a modelling framework for computer simulation of complex activities exhibited inside cells.

Based on above considerations, our point of view about the analogy between cells and computers is as follows:

(1) Cells as well as their biological molecules (i.e. proteins, enzymes, etc.) are able of information processing but information processing in biological systems is not like in digital computers. That is to say digital computers are a reason for finding inspiration about how cells perform information processing and self-organize their *wetware* (e.g. biomolecules, membrane, cytoplasm, etc.). For details about the differences between cells and computers we suggest the reading of 'Circuits of a Cell' published by the Berkeley Lab Research Review [Preuss, 2000].

(2) The 'hardware + software' dualism leads to a misleading conception of cell. In cells 'biomolecule + function' are inseparable elements in contrast with computers. For instance, it is possible to have a computer without operating system and software. Of course, it is not useful but in cells it is impossible to have biomolecules without biological function.

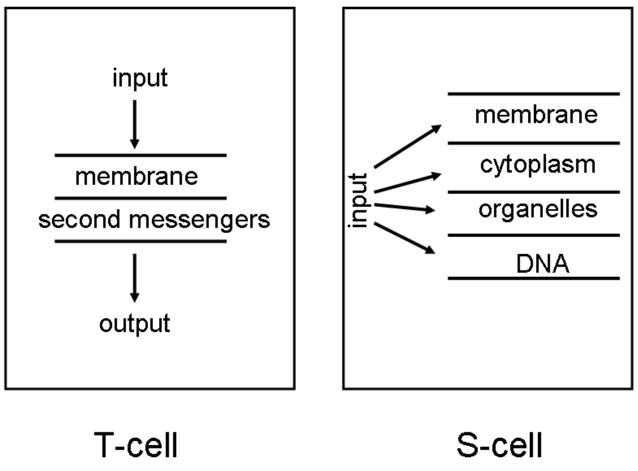

FIGURE 5. Organization of T-cell and S-cell

(3) Most views about information processing inside cells consider the organization of cell in layers as in Figure 5. Conceptually, such organization model finds its inspiration in the organization of software in layers (i.e. MS-Dos, Unix /Linux operating systems, Internet Protocol, etc.). We suggest that in agreement with this model of organization two different type of cells or architectures could be defined: T-cell or *transduction cell* and S-cell or *subsumption cell*. The transduction cell type is a model of cell that is similar to the senso-motor architecture used in simple autonomous agents (i.e. simple mobile robots). On the other hand, the subsumption cell should be similar to an operating system model in layers - where DNA and membrane represent the kernel and the shell respectively of the operating system - or to a complex autonomous agent (i.e. the robots designed by Brooks, see [Brooks, 1986]) where cells have the possibility to choose a layer depending on the problem to solve.

Conclusion

In conclusion, our point of view has been resumed in the above three statements (1) (2) (3) concluding that there is a 'likeness' between cells and computers. Cells are able of information processing but bearing in mind that bioinformation processing involves many features that are distinct from digital computers.

Bibliography

[Baltimore, 2011] D. Baltimore. 2011. Caltech and the Human Genome Project. DNA is a reality beyond metaphor. http://pr.caltech.edu/events/dna/dnabalt2.html

[Bray, 1990] D. Bray. 1990. Intracellular signalling as parallel distributed process. J. theor. Biol. 143: 215-231.

[Brooks, 1986] R.A. Brooks. 1986. A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation 2: 14.

[Chang and Poznansky, 1968] T.M. Chang, M.J. Poznansky. 1968. Semipermeable microcapsules containing catalase for enzyme replacement in acatalsaemic mice. Nature 218: 242-245.

[Di Paola et al., 2004]. V. Di Paola, P.C. Marijuan, R. Lahoz-Beltra. 2004. Learning and evolution in bacterial taxis: an operational amplifier circuit modeling the computational dynamics of the prokaryotic 'two component system' protein network. BioSystems. 74: 29-49.

[Hotani, et al., 1992] H. Hotani, R. Lahoz-Beltra, B. Combs, S. R. Hameroff, S. Rasmussen. 1992. Microtubule dynamics, liposomes and artificial cells: in Vitro observation and cellular automata simulation of microtubule assembly/disassembly and membrana morphogenesis. Nanobiology 1: 61-74.

[Kim, 1981] S.S. Kim. 1981. The mediator theory of acupuncture and its practical application in bronchial asthma and myasthenia gravis. Amer. J. Acupuncture 9(2): 101-116.

[Klyce, 1999] B. Klyce. 1999. What is Life? http://www.panspermia.org/whatis2.htm

[Lahoz-Beltra et al. 1993] R. Lahoz-Beltra, S.R. Hameroff, J.E. Dayhoff. 1993. Cytoskeletal logic: a model for molecular computation via Boolean operations in microtubules and microtubule associated proteins. BioSystems 29: 1-23.

[Lahoz-Beltra, 1998] R. Lahoz-Beltra. 1998. Molecular automata modeling in structural biology. Advances in Structural Biology 5: 85-101.

[Lahoz-Beltra, 1997] R. Lahoz-Beltra. 1997. Molecular automata assembly: principles and simulation of bacterial membrane construction. BioSystems 44: 209-229.

[Lahoz-Beltra, 2001] R. Lahoz-Beltra. 2001. Evolving hardware as model of enzyme evolution. BioSystems 61: 15-25.

[Noireaux et al., 2011] V. Noireaux, Y.T. Maeda, A. Libchaber. 2011. Development of an artificial cell, from selforganization to computation and self-reproduction. PNAS 108: 3473–3480.

[Paton, 1994] R. Paton, (Ed.). 1994. Computing with Biological Metaphors, Chapman and Hall.

[Preuss, 2000]. P. Preuss. Circuits of a Cell 2000. http://www.lbl.gov/Science-Articles/Research-Review/Magazine/2000/Winter/features/06circuits_splash.html

[Ray, 1991]. T.S. Ray. 1991. An approach to the synthesis of life, in: Artificial Life II, C.G. Langton et al. Eds., Addison-Wesley: 371-408.

[Sipper et al., 1997a] M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Pérez-Uribe, A. Stauffer. 1997a. The POE model of bio-inspired hardware systems: A short introduction, in: Genetic Programming 1997: Proceedings of the Second Annual Conference, J.R. Koza, J.R. et al. Eds., Morgan Kauffman: 510-511.

[Sipper et al., 1997b] M. Sipper, D. Mange, A. Stauffer. 1997b. Ontogenetic hardware. BioSystems 44: 193-207.

[Tomita et al., 1999] M. Tomita, K. Hashimoto, K. Takahashi, T.S. Shimizu, Y. Matsuzaki, F.Miyoshi, K. Saito, S. Tanida, K. Yugi, J.C. Venter, C.A. Hutchison. 1999. E-CELL: software environment for whole-cell simulation. Bioinformatics 15: 72-84.

Authors' Information

Rafael Lahoz-Beltra *– Dept. of Applied Mathematics, Faculty of Biological Sciences, Complutense University of Madrid, 28040 Madrid, Spain ; e-mail: lahozraf@bio.ucm.es*

Major Fields of Scientific Research: evolutionary computation, embryo development modeling and the design of bioinspired algorithms