The Microsoft Research - University of Trento
Centre for Computational
and Systems Biology

Technical Report CoSBi 20/2008

Algorithmic Systems Biology
An Opportunity for Computer Science

Corrado Priami
CoSBi
and
DISI, University of Trento
`priami@cosbi.eu`

This is the preliminary version of a paper that will appear in Communication of the ACM, 52(5):80-88, May 2009 available at `http://cacm.acm.org/magazines/2009/5/24650-algorithmic-systems-biology/abstract`

Algorithmic Systems Biology
An Opportunity for Computer Science

Corrado Priami
CoSBi and University of Trento
priami@cosbi.eu

Abstract. The convergence between computer science and biology occurred in successive waves, involving deeper and deeper concepts of computing. The current situation makes computer science a suitable candidate for becoming a philosophical foundation for systems biology with the same importance as mathematics, chemistry and physics. However, this significant opportunity is not a free lunch. New developments and a strong integration of different fields of computing are needed to face the challenges of systems biology. One of these developments is that of a complex and expanding applicative domain that can open entirely new avenues of research in computing and eventually help it become a natural, quantitative science.

Converging sciences

Since its early days, computer science has taken inspiration from nature, with the works of Turing, Von Neumann and Minsky. Throughout the history of computer science, many researchers looked at nature, and this inspiration led to extraordinary results, some of which recall biology even in their names: cellular automata, neural networks, genetic algorithms.

Computing and biology have been converging ever more closely for the past two decades, but with a vision of computing as a resource for biology that has propelled bioinformatics. Bioinformatics addresses structural and static aspects of biology and has produced databases, patterns manipulation and comparison, search tools and data mining techniques [S00,R01]. The most significant success has been the Human Genome Project, which was made possible by the selection of the correct language abstraction for representing DNA (a language with a 4-character alphabet) [S02].

Biology is now experiencing a heightening of interest in system dynamics interpreting living organisms as information manipulators [HG03] and is moving towards systems biology [K02]. There is no general agreement on a definition of systems biology, but whatever definition we select, it must embrace at least four characterizing concepts. Systems biology is a transition

1. from qualitative biology towards a quantitative science
2. from reductionism to system level understanding of biological phenomena
3. from structural, static descriptions to functional, dynamic properties
4. from descriptive biology to mechanistic/causal biology.

The above features highlight that causality between events, temporal ordering of interactions and spatial distribution of components within the reference volume of reactions are becoming essential in addressing biological questions at system level. This poses new challenges to describe the step-by-step mechanistic behavior enabling phenotypical phenomena that bioinformatics does not address [CA05].

Fig. 1. Algorithms describe how to move from pictures to films. The current practice is composed of taking pictures of biological systems and trying to model the variation of measures in the pictures through equations. No causal explanation is given (upper part of the figure). Algorithms describe the steps from one picture to the next in a causal continuum of actions that make the measures change, thus providing a dynamic view (a film) of the system under question.

Mathematical modeling is a philosophical foundation for systems biology for specifying and testing hypotheses about systems [B07]. Mathematical modeling is also a key aspect of computational biology that deals with the solution of systems of equations (models) through computer programs [N02]. This process is sometimes termed “simulation,” because the solution of the system of equations provides an average variation of the concentrations of the components that constitute the biological system considered. More generally, computational biology addresses the use of computers to handle biology-related issues, rather than the more fundamental principles of the computational strategies and capabilities implemented by living organisms that computing principles can unravel. The main concept to be exploited involves algorithms and the (programming) languages used to specify them. We can then recover temporal, spatial and causal information on the modeled systems by using well-established computing techniques that deal with program analysis, composition and verification, integrated software development environments and debugging tools as well as computational complexity and algorithm animation. The convergence between computing and systems biology on a peer-to-peer basis is then a valuable opportunity that can fuel the discovery of solutions to many of the current challenges in both fields, moving towards an algorithmic view of systems biology.

The main difference between algorithmic systems biology and other techniques used to model biological systems stems from the intrinsic difference of algorithms (operational descriptions) and equations (denotational descriptions). An equation might be an elegant way of describing the result of the execution of an algorithm. Furthermore, equations specify dynamic processes by abstracting the steps performed by the executor, thus hiding from the user the causal, spatial and temporal relationships between the elementary steps. Equations describe the variation of variables (usually concentrations of species) from one state to another of a system, while algorithms highlight *why* and *how* a system moves from one state to another one. We could simplify the difference by stating that we move from the pictures described by equations to the film described by algorithms.

Algorithms precisely describe the behavior of systems with discrete state spaces, while equations describe an average behavior of systems with continuous state spaces. However, quantitative hybrid approaches exist that manipulates discrete state spaces annotated with continuous variables through algorithms [Aetal95, DPR08].

It is well-known in computer science that input-output relationships are not suitable for characterizing the behavior of concurrent systems where many threads of execution are simultaneously active (and in biological systems millions of interactions happen simultaneously). Concurrency theory was developed as a formal framework in which to model and analyze parallel, distributed and mobile systems, and this led to the definition of specific programming primitives and algorithms. Equations, instead, are sequential tools that attempt to model a system whose behavior is completely determined by input-output relations. The sequential assumption of equations also impacts the notion of causality that in a sequential setting coincides with the temporal ordering of events. In a parallel context causality is instead the complement of concurrency [DP99] and it may not coincide with the temporal ordering of the observed events. Therefore, relying on a sequential modeling style to describe a concurrent system immediately makes the modeler lose the connection with causality.

The full involvement of computer science in systems biology can be an arena in which to distinguish computing and mathematics, and clarify a discussion that has been going on for 40 years [K74, D74]. Algorithms and the coupling of executions/executors are the key point of the differentiation.

Algorithms force modelers/biologists to think about the mechanisms governing the behavior of the system under question. Therefore, they are a conceptual tool that helps to elucidate fundamental biological principles. Algorithms are a practical tool for expressing and favoring computational thinking [W06]. Similar ideas have been recently expressed in [C08].

Algorithms are quantitative when the selection mechanism of the next step is determined according to probabilistic/temporal distributions associated with either the rules or the components of the system modeled. Since the dynamics of biological systems are mainly driven by quantities such as concentrations, temperatures, gradients etc., we must clearly focus on quantitative algorithms and languages.

Algorithms can help in coherently extracting general biological principles that underlie the enormous amount of data produced by high-throughput technologies. Algorithms can also organize data in a clear and compact way, thus producing knowledge from information (data). This is actually aligned with the idea of Nobel Laureate Sydney Brenner that biology needs a theory able to highlight causality and abstract data into knowledge to elucidate the architecture of biological complexity.

Algorithms need a syntax to be described and a semantics to associate them with their intended meaning so that an executor can precisely perform the steps needed to implement the algorithms with no ambiguity. In this way, we are entering the realm of programming languages from both a theoretical and a practical perspective.

The use of programming languages to model biological systems is an emerging field that enhances current modeling capabilities (richness of aspects that can be described as well as easiness, composability and reusability of models) [PQ04]. The metaphor which inspires this idea is one where biological entities are represented as programs being executed simultaneously and the interaction of two entities is represented by the exchange of a message between the programs representing those entities in the model [RS03]. The biological entities involved in the biological process and the corresponding programs in the abstract model are in a 1:1 correspondence, thus coping by construction with the combinatorial explosion of variables needed in the mathematical approach to describe the whole set of states through which a single component can pass.

The metaphor above explicitly refers to concurrency. Indeed concurrency is endemic in nature, and we see this in examples ranging from atoms, molecules in living organisms, organisms themselves and populations to astronomy. If we are going to re-engineer artificial systems to match the efficiency, resilience, adaptability and robustness of natural systems, then concurrency must be a core design principle that, at the end of the day, will simplify the entire design and implementation process. Concurrency must not be considered a tool to improve the performance of sequential programming languages and architectures, which is the standard practice in most actual cases. Some programming languages that address concurrency as a core primitive issue and aim at modeling biological systems are emerging, e.g., [WB05, DPR08], from the field of process calculi [BPS01]. These concurrent programming languages are very promising for the establishment of a link between artificial concurrent programming and natural phenomena, thus contributing to the exposure of computer science to experimental, natural sciences. Furthermore, concurrent programming languages are suitable candidates for easily and efficiently expressing the mechanistic rules that propel algorithmic systems biology. The suitability of these languages is reinforced by their clean and formal definition that both supports the verification of properties and the analysis of systems and provides no engineering surprises as could happen with classical threads and lock mechanisms [WB05]

A recent paper by Nobel Laureate Paul Nurse advocates that a better understanding of living organisms requires "both the development of the appropriate languages to describe information processing in biological systems and the generation of more effective methods to translate biochemical descriptions into the functioning of the logic circuits that underpin biological phenomena" [N08]. This description perfectly adheres both to the need of a deeper involvement of computer science in biology and to the need of an algorithmic description of life based on a suitable language that makes analyses easier. Nurse's statement implicitly assumes that the modeling techniques adopted so far are not adequate to address the new challenges raised by systems biology.

Finally, it is important to note that process calculi are not the only theoretical basis for algorithmic systems biology. Petri nets, logic, rewriting systems and membrane computing are other relevant examples of formal methods applied to systems biology (for a collection of tutorials see [BDZ08]). Other approaches that are more closely related to software design principles are the adaptation of UML to biological issues (see www.biouml.org) and statecharts [H07]. Finally, cellular automata [G90] need to be considered as well with their game of life.

Fig. 2. The biological systems observed through the window showing the life sciences (green rectangle) can be closely and mechanistically modeled through the use of algorithms (written on the glass of the window) that add causal, spatial and temporal dimension to classical biological descriptions. Furthermore, algorithms can concisely and intentionally represent the large amount of data produced by high-throughput experiments (the river of numbers originating from biological elements within the window). Equations (considered the stars of modeling) are more abstract and hence more distant from living matter. The goal of algorithmic systems biology is to reach the moon, which is a complete mechanistic model of living systems (the lighted semi-sphere in the picture represents a cell under a digitalization process).

Conclusions

Quantitative algorithmic descriptions of biological processes add the causal, spatial and temporal dimensions to cartoons that disentangle the mechanistic behavior of molecular machineries that are usually hidden in the equations. Algorithmic systems biology allows us to take a step forward in our understanding of life by transforming collections of pictures (the cartoons) into spectacular films (the mechanistic dynamics of life). The languages and algorithms emerging from quantitative computing can be a further brick in the foundations of systems biology and the science of interactions in general.

Unraveling the very basic mechanisms adopted by living organisms in order to compute and manipulate information leads to the heart of computer science: computability. Life underwent billions of years of tests and was optimized during this time; we can now learn new computational paradigms to enhance our field. The same arguments apply to hardware architectures as well. Starting from the very basics, we can further build on top of these new computational paradigms to strengthen resource management and hence operating systems, primitives to instruct highly parallel systems and hence (concurrent) programming languages, up to software development environments that ensure higher quality and better properties than current software applications.

Algorithmic systems biology can contribute to the future of life and natural sciences through connecting models and experiments by means of new conceptual and computational tools integrated in a user-friendly environment equipped with templates of major biological components for drag-and-drop modeling of (artificial) organisms or populations and used by a large part of life scientists to predict the behavior of multi-level, multi-scale biological systems in a modular, composable, scalable and executable manner.

Algorithmic systems biology can also contribute to the future of computer science by developing a novel generation of operating systems and programming languages that enable simulation-based research within a quantitative reference framework that connects in-silico replica and actual systems by means of the new, biologically-inspired quantitative conceptual and computational tools.

Acknowledgements. The author thanks the CoSBi team for the numerous inspiring discussions.

References

[A06] Alon, U. *An introduction to systems biology: design principles of biological circuits.* Chapman and Hall, 2006.

[Aetal95] Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S. The algorithmic analysis of hybrid systems. Theoretical Computer Science, 138(1):3-34,1995.

[AN08] Auffray, C., Nottale, L. *Scale relativity theory and integrative systems biology: founding principles and scale laws.* Progress in Biophysics and Molecular Biology, 97:79-114, 2008.

[B02] Brekling, B. *Individual-based modelling: potentials and limitations.* Scientific World Journal, 2:1044-1062, 2002.

[B07] Boogerd, F et al. *Systems Biology: Philosophical Foundations.* Elsevier, 2007.

[BPS01] Bergstra, J.A., Ponse, A., Smolka, S.A. *Handbook of Process Algebras.* Elsevier, 2001.

[BS05] Benner, S.A., Sismour, A.M. *Synthetic Biology.* Nature Reviews Genetics, 6:533-544, 2005.

[BDZ08] Bernardo, M., Degano, P., Zavattaro, G. *Formal Methods for Computational Systems Biology.* LNCS 5016, Springer, 2008.

[CA05] Cassman, M., Arkin, A., Doyle, F., Katagiri, F., Lauffenburg, D., Stokes, C. *International Research and Development in Systems Biology.* WTEC Panel on Systems Biology final report, October 2005.

[Cetal07] Chiarugi, D., Degano, P., Marangoni, R. (2007) A Computational Approach to the Functional Screening of Genomes. PLoS Comput Biol 3(9): e174. doi: 10.1371/journal.pcbi.0030174, 2007.

[C08] Cohen, J. *The crucial role of CS in systems and synthetic biology.* Comm. ACM, 51(5):15-18, 2008.

[CH08] Ciocchetta, F., Hillston, J. Process algebras in systems biology. In Formal methods for computational systems biology, LNCS 5016: 313-365, Springer, 2008.

[detal01] de Alfaro, L., Henzinger, T.A., Jhala, R. Compositional methods for probabilistic systems. In CONCUR01, LNCS 2154, 2001.

[D74] Dijstra, E.W. *Programming as a Discipline of Mathematical Nature*. American Mathematical Monthly, 81:608-612, 1974.

[D03] Denning, P.J. *Great principles of computing.* Comm. ACM, 46(11):15-20, 2003.

[D05] Denning, P.J. *Is Computer Science Science?* Comm. ACM, 48(4):27-31, 2005.

[D05a] Denning, P.J. *Recentering Computer Science.* Comm. ACM, 48(11):15-19, 2005.

[D07] Denning, P.J. *Computing is a natural science.* Comm. ACM, 50(7):13-18, 2007.

[DP99] Degano, P., Priami, C. Non-interleaving semantics of mobile processes. Theoretical Computer Science, 216(1-2):237-270 , 1999.

[DPR08] Dematté, L., Priami, C., Romanel, A. *The BlenX Language: a Tutorial.* In Formal methods for computational systems biology, LNCS 5016: 313-365, Springer, 2008.

[DPR08a] Dematté, L., Priami, C., Romanel, A. *The Beta Workbench: A tool to study the dynamics of biological systems.* Briefings in Bioinformatics, 2008.

[FB07] Forrest, S., Beauchemin, C. *Imm. Reviews* 216 (1), 176.197, 2007

[FC06] Foster, A.C., Church, G.M. *Towards synthesis of a minimal cell.* Molecular Systems Biology, doi:10.1038/msb4100090, 2006.

[FH07] Fisher, J., Henzinger, T. *Executable Cell Biology*. Nature Biotechnology, 25:1239-1249, 2007.

[G02] Goldbeter, A. *Computational approaches to cellular rhythms.* NATURE, 420:238-245, 2002.

[G77] Gillespie, D.T. Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry, 81: 2340-2361, 1977.

[G90] Gutowitz, H. *Introduction (to cellular automata)*. Physica D, Vol. 45, 1990.

[H07] Harel, D. *Statecharts in the Making: A Personal Account.* Proc. 3rd ACM SIGPLAN History of Programming Languages Conference (HOPL III), 2007.

[HG03] Hood, L., Galas, D. *The digital code of DNA.* NATURE, 421:444-448, 2003.

[Hetal08] Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T., Weitzner, D. *Web Science: An Interdisciplinary Approach to Understanding the Web.* Comm. ACM, 51(7): 60-69, 2008.

[IPL01] Priami, C., Regev, A., Shapiro, E., Silvermann, W. *Application of a stochastic name-passing calculus to representation and simulation of molecular processes.* Information Processing Letters, 80:25-31, 2001.

[K02] Kitano, H. *Systems Biology: A Brief Overview*. SCIENCE, 295:1662-1664, 2002.

[K74] Knuth, D. *Computer Science and its Relation to Mathematics.* American Mathematical Monthly, 81:323-343, 1974.

[KM08] Kuwahara, H., Mura, I. An efficient and exact stochastic simulation method to nalyze rare events in biological systems. Journal of chemical physics, 129, 2008.

[KS05] Klawe, M., Shneiderman, B. *Crisis and Opportunity in Computer Science.* Comm. ACM 48(11):27-28, 2005.

[Letal08] Ludtke, N., Panzeri, S., Brown, M., Broomhead, D.S., Knowles, J., Montemurro M.A., Kell, D.B. *Information-theoretic sensitivity analysis: a general method for credit assignment in complex networks.* J. R. Soc. Interface, 5:223-235, 2008.

[M70] Minsky, M. *ACM Turing Lecture: Form and Content in Computer Science. Journal of the ACM* 17(2):197-215, 1970.

[N02] *Nature insight: Computational Biology*. NATURE, 420:206-251, 2002.

[N08] Nurse, P. *Life, Logic and Information.* NATURE, 454:424-426, 2008.

[NA91] Nussbaum, D. Agarwal, A. *Scalability of parallel machines.* Comm. ACM, 34(3): 57-61, 1991.

[OD05] O'Malley, M., Dupre', J. *Fundamental Issues in Systems Biology.* BioEssays, 27:1270-1276, 2005.

[P06] Palsson, B.O. *Systems biology: properties of reconstructed networks.* Cambridge University Press, 2006.

[PQ04] Priami, C., Quaglia, P. *Modeling the dynamics of biosystems.* Briefings in Bioinformatics, 5:259-269, 2004.

[R01] Roos, D. *Bioinformatics -- Trying to swim in a sea of data.* SCIENCE, 291:1260-1261, 2001.

[Retal06] Rapin, N., Kesmir, C., Frankild, S., Nielsen, M., Lundegaard, C., Brunak, S., Lund, O. *Modelling the Human Immune System by Combining Bioinformatics and Systems Biology Approaches.* Journal Biol. Phys., 32:335-353, 2006.

[RS03] Regev, A., Shapiro, E. *Cells as computation*. NATURE, 419:343, 2002.

[RWA02] Rao, C.V., Wolf, D.M., Arkin, A.P. *Control, exploitation and tolerance of intracellular noise.* NATURE, 420:231-237, 2002.

[S00] Spengler, S.J. *Bioinformatics in the information age.* SCIENCE 287, 1221-1223, 2000.

[S02] Searls, D. *The Language of Genes*. NATURE, 420:211-217, 2002.

[T06] Teuscher, C. *Biologically Uninspired Computer Science.* Comm. ACM, 49(11): 27-29, 2006.

[V26] Volterra, V. *Fluctuations in the abundance of species considered mathematically.* NATURE, 118:558-560, 1926.

[W06] Wing, J. *Computational thinking.* Comm. ACM, 49(3):33-35, 2006.

[WB05] Welch, P.H., Barnes, F.R.M. *Communicating Mobile Processes: Introducing occam-pi.* In CSP25, LNCS 3525:175-210, Springer, 2005.