HSPI The Open Access Publisher

A B B ARCHIVES OF BIOTECHNOLOGY AND BIOMEDICINE
ISSN 2639-6777

Short Communication

# Computational Models in Systems and Synthetic Biology: Short Overview

**Marian Gheorghe***

Department of Computer Science, University of Bradford, Bradford, BD7 1DP, UK

## Abstract

Computational models used in specifying biological systems represent a complement and become an alternative to more widely used mathematical models. Amongst some of the advantages brought by these computational models, one can mention their executable semantics and mechanistic way of describing biological system phenomena. This short overview report enumerated some of the computational models utilised so far in systems and synthetic biology, the associated analysis and formal verification methods and tools, and a way of facilitating a broader use of this alternative approach.

More Information

*Address for correspondence: Marian Gheorghe, Department of Computer Science, University of Bradford, Bradford, BD7 1DP, UK,
Email: m.gheorghe@bradford.ac.uk

Submitted: January 02, 2024
Approved: January 17, 2024
Published: January 18, 2024

How to cite this article: Gheorghe M. Computational Models in Systems and Synthetic Biology: Short Overview. Arch Biotechnol Biomed. 2024; 8: 001-002.

DOI: 10.29328/journal.abb.1001037

https://orcid.org/0000-0002-2409-4959

Keywords: System and synthetic biology; Executable biology; Computational models; Formal methods; Model checking; User-friendly modelling; Healthcare; Cyber-physical systems

Computer science provides a huge range of applications, using specific models, methods, and tools, initially developed for investigations related to software design, implementation, formal verification, and validation [1]. The applications field has grown and these computer science developments have been used in the study of physical and engineering processes, including those occurring in systems and synthetic biology [2].

Although mathematical-based approaches play a very important role in modelling biological systems, *computational models* [3] have recently emerged as a way to complement and provide an alternative approach to mathematical formalism. This new approach provides a “deeper mechanistic understanding of biological systems” and is “more amenable to easier communication between the biologist and modeler” [3]. These models are directly executable, hence the term associated with them, *executable biology* [3]. A rich set of concepts and methods, developed for *analysing, formally verifying,* and *validating* software systems have been directly transferred into systems and synthetic biology modelling.

A few works presented and analysed various computational models such as process algebras, rule-based systems, Petri nets, Boolean networks, statecharts, hybrid systems, compartment-based, agent-based and lattice-based models, symbolic modelling, knowledge-based expert systems and membrane computing models [2,3]. Each of them has a specific operational semantics allowing to directly execute the specification that may be expressed through domain-specific languages associated with these models. The generated program can be used to investigate the behaviour of the system, but also for analysing it with specific computer science *formal verification tools*, such as model checking, runtime verification, and static analysis [2].

The use of computational models in specifying different biological systems has led to a continuous increase of well-established methods and tools developed initially for the analysis and verification of programs and programming languages, to the analysis and verification of the computational models built for such systems. It is well-known that despite the progress made in using formal methods in software verification and their applications in engineering [2], they faced serious problems in spreading their use in these areas given the high specialisation requested and lack of adequate user-oriented tools [2]. Some solutions to alleviate these problems have been provided through more *user-friendly* ways of specifying the mathematical formalism involved [4,5].

Subsequently, we illustrate the usage of *model checking* as an analysis method that helps reveal certain properties of the model. In order to formally analyse a biological system, a formal model of the system is provided and a formal specification of the properties to be verified against the formal model is given. The former may be expressed as a domain-specific language, whereas the latter appears as a temporal logic notation [3]. Although the domain-specific language requires some understanding of the concepts used by the model, its syntax, and semantics, it is, in general, easier to understand than a mathematical model. The temporal logic notation may appear to be quite tricky for a non-specialist and for this reason in engineering, where these approaches are applied, especially in safety-critical systems, some *patterns* that are encountered more often are expressed in natural language-like statements [3]. Something similar has been considered in the case of computational models used for biological systems [4,5]. This approach is illustrated by a synthetic biology example.

A quorum-sensing device produces a transcriptional factor, LasR, that binds a signalling molecule, N-Acylhomoserine Lactone (AHL), forming a complex LasR-AHL. The LuxR promoter, to which the LaxR-AHL activator complex binds, serves as the inducible promoter in the sensing system, leading to the production of Green Fluorescent Protein (GFP) as the reported protein [6]. The model is defined as a membrane computing model, using a domain-specific language and a query checking the relationship between the presence of the AHL molecule and the production of GFP. This query can be stated in a precise probabilistic temporal logic formalism, but this may be formulated through the following natural language-like pattern [6]: "What is the probability that the system produces GFP if there is no AHL?". As expected, the answer to this query is that the probability is 0.0, which ensures that the system will not produce GFP when AHL is absent. This pattern-based approach in formulating various properties makes the use of model-checking more user-friendly to non-specialists. More complex patterns can be built in a compositional and systematic manner by using a design pattern language for *biological verification* [3]. Based on this a coherent methodology of using, in a *user-friendly* manner, model-checking tools in the formal verification of biological systems, is presented in [3].

Numerous biological systems have been investigated using computational modelling approaches and formal analysis methods and techniques. Recently, tools, combining computational models with formal analysis approaches have been built. Another step forward has been made by automatically selecting the most appropriate model-checking tool when more than one may be available to be used to verify a system [3]. These new developments will make the use of computational methods "more efficient, effective, and easy to use, even for non-experts in formal methods" [3].

The technological progress in *healthcare* that has led to devices called *cyber-physical systems* [2], where smart devices interact with biological systems, may benefit from the use of computational modelling in an integrative and uniform manner. The cyber-physical systems use quite extensively computational modelling approaches in their design, implementation, testing, and formal verification, and these will be smoothly integrated with similar models used in the biological component of the healthcare system.

## Conclusion

In this short overview report, we have presented some of the most used executable biology methods, including computational models and formal analysis methods and techniques. They complement and become an alternative approach to more widely used mathematical models and methods. Executable methods may be integrated within cyber-physical systems that start being more widely used in healthcare and other medical areas.

## References

1. Garavel H, Beek MHT, Van de Pol J. The 2020 Expert Survey on Formal Methods, Proc. Formal Methods for Industrial Critical Systems (FMICS 2020). LNCS 12327, Springer 2020.
2. Bartocci E, Lió P. Computational Modeling, Formal Analysis, and Tools for Systems Biology. PLoS Comput Biol. 2016 Jan 21;12(1):e1004591. doi: 10.1371/journal.pcbi.1004591. PMID: 26795950; PMCID: PMC4721667.
3. Konur S, Gheorghe M, Krasnogor N. Verifiable biology. J R Soc Interface. 2023 May;20(202):20230019. doi: 10.1098/rsif.2023.0019. Epub 2023 May 10. PMID: 37160165; PMCID: PMC10169095.
4. Mitra ED, Suderman R, Colvin J, Ionkov A, Hu A, Sauro HM, Posner RG, Hlavacek WS. PyBioNetFit and the Biological Property Specification Language. iScience. 2019 Sep 27; 19:1012-1036. doi: 10.1016/j.isci.2019.08.045. Epub 2019 Aug 28. PMID: 31522114; PMCID: PMC6744527.
5. Hall BA, Fisher J. Constructing and Analyzing Computational Models of Cell Signaling with BioModelAnalyzer. Curr Protoc Bioinformatics. 2020 Mar; 69(1):e95. doi: 10.1002/cpbi.95. PMID: 32078258.
6. Konur S, Gheorghe M, Dragomir C, Mierla L, Ipate F, Krasnogor N. Qualitative and quantitative analysis of systems and synthetic biology constructs using P systems. ACS Synth Biol. 2015 Jan 16;4(1):83-92. doi: 10.1021/sb500134w. Epub 2014 Aug 14. PMID: 25090609.