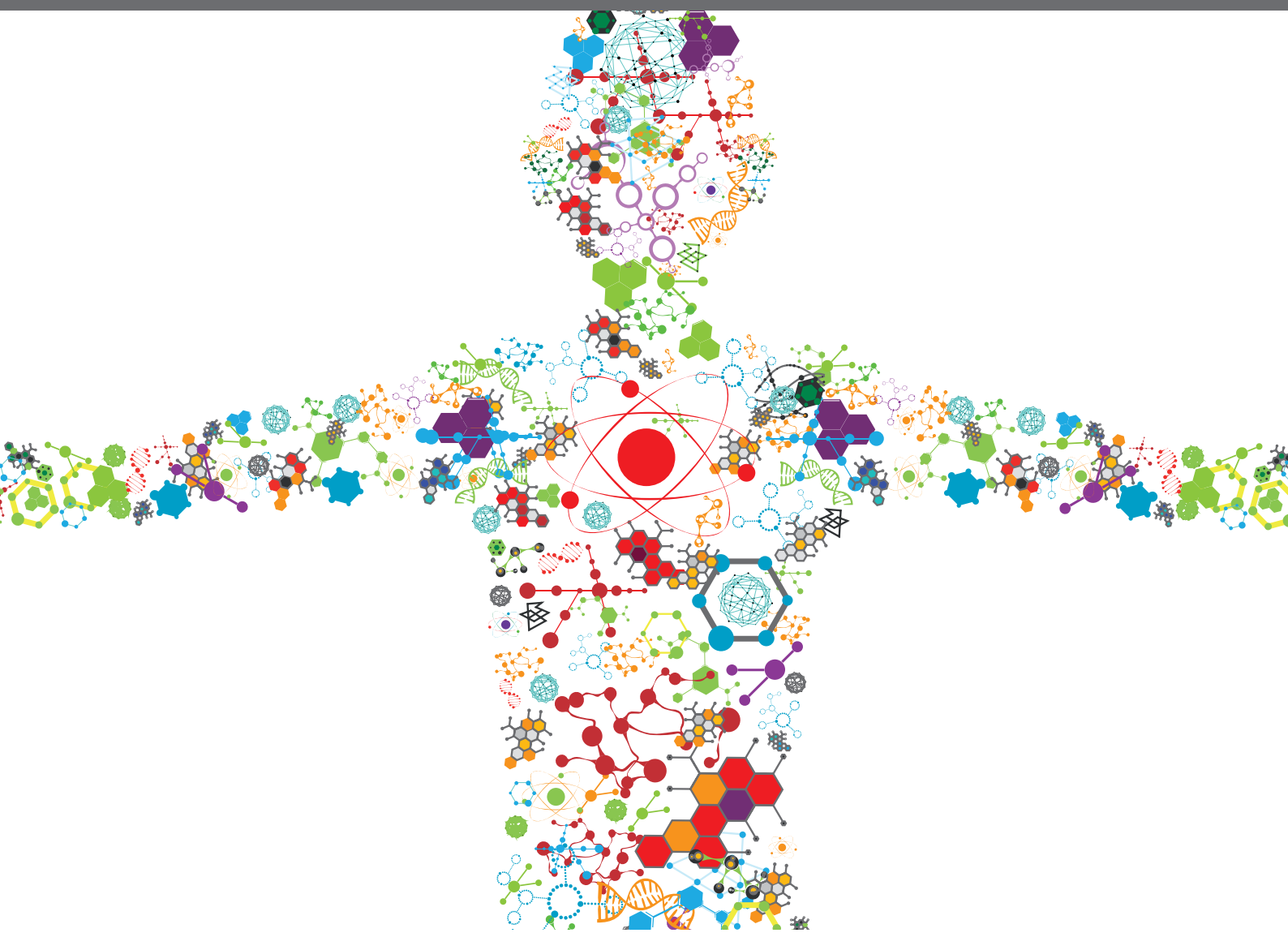
COMPUTER-AIDED BIODESIGN ACROSS SCALES

EDITED BY: Thomas E. Gorochowski, Fabio Parmeggiani, Jonathan Karr and Boyan Yordanov

PUBLISHED IN: Frontiers in Bioengineering and Biotechnology and Frontiers in Genetics



frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88971-150-5
DOI 10.3389/978-2-88971-150-5

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

COMPUTER-AIDED BIODESIGN ACROSS SCALES

Topic Editors:
Thomas E. Gorochowski, University of Bristol, United Kingdom
Fabio Parmeggiani, University of Bristol, United Kingdom
Jonathan Karr, Icahn School of Medicine at Mount Sinai, United States
Boyan Yordanov, Scientific Technologies Ltd, United Kingdom

Citation: Gorochowski, T. E., Parmeggiani, F., Karr, J., Yordanov, B., eds. (2021). Computer-Aided Biodesign Across Scales. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88971-150-5

Table of Contents

04 ***Editorial: Computer-Aided Biodesign Across Scales***
Thomas E. Gorochowski, Jonathan R. Karr, Fabio Parmeggiani and Boyan Yordanov

07 ***Toward Engineering Biosystems With Emergent Collective Functions***
Thomas E. Gorochowski, Sabine Hauert, Jan-Ulrich Kreft, Lucia Marucci, Namid R. Stillman, T.-Y. Dora Tang, Lucia Bandiera, Vittorio Bartoli, Daniel O. R. Dixon, Alex J. H. Fedorec, Harold Fellermann, Alexander G. Fletcher, Tim Foster, Luca Giuggioli, Antoni Matyjaszkiewicz, Scott McCormick, Sandra Montes Olivas, Jonathan Naylor, Ana Rubio Denniss and Daniel Ward

16 ***From Microbial Communities to Distributed Computing Systems***
Behzad D. Karkaria, Neythen J. Treloar, Chris P. Barnes and Alex J. H. Fedorec

38 ***Cell-Free Systems: A Proving Ground for Rational Biodesign***
Nadanai Laohakunakorn

46 ***SSRMMD: A Rapid and Accurate Algorithm for Mining SSR Feature Loci and Candidate Polymorphic SSRs Based on Assembled Sequences***
Xiangjian Gou, Haoran Shi, Shifan Yu, Zhiqiang Wang, Caixia Li, Shihang Liu, Jian Ma, Guangdeng Chen, Tao Liu and Yaxi Liu

56 ***Computer-Aided Whole-Cell Design: Taking a Holistic Approach by Integrating Synthetic With Systems Biology***
Lucia Marucci, Matteo Barberis, Jonathan Karr, Oliver Ray, Paul R. Race, Miguel de Souza Andrade, Claire Grierson, Stefan Andreas Hoffmann, Sophie Landon, Elibio Rech, Joshua Rees-Garbutt, Richard Seabrook, William Shaw and Christopher Woods

67 ***Novel Tunable Spatio-Temporal Patterns From a Simple Genetic Oscillator Circuit***
Guillermo Yáñez Feliú, Gonzalo Vidal, Macarena Muñoz Silva and Timothy J. Rudge

83 ***The Synthetic Biology Open Language (SBOL) Version 3: Simplified Data Exchange for Bioengineering***
James Alastair McLaughlin, Jacob Beal, Göksel Mısırlı, Raik Grünberg, Bryan A. Bartley, James Scott-Brown, Prashant Vaidyanathan, Pedro Fontanarrosa, Ernst Oberortner, Anil Wipat, Thomas E. Gorochowski and Chris J. Myers

98 ***Prediction of Protein Mutational Free Energy: Benchmark and Sampling Improvements Increase Classification Accuracy***
Brandon Frenz, Steven M. Lewis, Indigo King, Frank DiMaio, Hahnbeom Park and Yifan Song

106 ***Elfin UI: A Graphical Interface for Protein Design With Modular Building Blocks***
Chun-Ting Yeh, Leon Obendorf and Fabio Parmeggiani

frontiers in Bioengineering and Biotechnology

EDITORIAL
published: 15 June 2021
doi: 10.3389/fbioe.2021.700418

Editorial: Computer-Aided Biodesign Across Scales

Thomas E. Gorochowski[1,2*†‡], *Jonathan R. Karr*[3†‡], *Fabio Parmeggiani*[2,4†‡] *and Boyan Yordanov*[5,6†‡]

[1] School of Biological Sciences, University of Bristol, Bristol, United Kingdom, [2] BrisSynBio, University of Bristol, Bristol, United Kingdom, [3] Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY, United States, [4] School of Chemistry and School of Biochemistry, University of Bristol, Bristol, United Kingdom, [5] Scientific Technologies Ltd., London, United Kingdom, [6] Microsoft Research, Cambridge, United Kingdom

Keywords: biodesign, synthetic biology, bioengineering, multi-scale, systems biology, computational modeling, protein design, biocomputation

OPEN ACCESS

Edited and reviewed by:
Jean Marie François, Institut Biotechnologique de Toulouse (INSA), France

***Correspondence:**
Thomas E. Gorochowski
thomas.gorochowski@bristol.ac.uk

†These authors have contributed equally to this work

‡ORCID:
Thomas E. Gorochowski
orcid.org/0000-0003-1702-786X
Jonathan R. Karr
orcid.org/0000-0002-2605-5080
Fabio Parmeggiani
orcid.org/0000-0001-8548-1090
Boyan Yordanov
orcid.org/0000-0002-4149-6220

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *26 April 2021*
Accepted: *20 May 2021*
Published: *15 June 2021*

Citation:
Gorochowski TE, Karr JR, Parmeggiani F and Yordanov B (2021) Editorial: Computer-Aided Biodesign Across Scales. Front. Bioeng. Biotechnol. 9:700418. doi: 10.3389/fbioe.2021.700418

Editorial on the Research Topic

Computer-Aided Biodesign Across Scales

INTRODUCTION

Computer-aided design (CAD) has revolutionized many engineering fields, enabling the quick exploration and testing of designs *in silico,* that minimizes the need for expensive and laborious physical assembly and experimentation. We believe that CAD will become similarly important to synthetic biology. However, synthetic biology presents some unique challenges for CAD, including the multi-scale structure of biology, the combinatorial complexity of molecular systems due the low degree of insulation inside cells, the stochastic nature of many biological processes, and our limited ability to accurately characterize the components of these living systems. Despite these challenges, numerous advancements are being made toward CAD for many aspects of biodesign. These advances are accelerating our abilities to efficiently assemble synthetic biological systems and revealing underlying principles for their effective design. In this Research Topic we have collated a broad range of original research, perspectives, and reviews covering some of the current approaches to computer-aided biodesign across scales (**Figure 1**).

DE NOVO MOLECULAR PREDICTION AND DESIGN

At the molecular level, the design of proteins and DNA sequences pose many challenges, but reliable modeling at this level will offer the means to scale-up biodesign, moving our focus from single molecules to complex multi-component systems. To make this step, modeling and design tools need to be able to recapitulate experimental data.

To address this need, Frenz et al. analyze the ProTherm database, which contains thermodynamic information about large numbers of protein mutations, to build a broader understanding of potential biases and to develop a curated subset to improve the prediction of mutational effects. Such robust inference is at the core of traditional protein design methods and Yeh et al. push these approaches further by developing an interactive user interface (Elfin UI) to build proteins and protein complexes with arbitrary shapes from compatible structural building blocks. These architectures are much larger (in the range of thousands of amino acids) than routinely designed proteins and begin to cross the boundary from the molecular to the cellular scale. Modeling tools are also central to the prediction of desirable features at a genetic level.

Gou et al. describe SSRMMD, an algorithm for identification of microsatellites (or simple sequence repeats, SSRs) within genomes to allow for better navigation and comparison of genomes.

TOWARD THE DESIGN OF CELL-FREE SYSTEMS AND WHOLE CELLS

Due to the complexity of molecular biology, the molecular to cell scale is one of the most difficult scales of biology to design. Designing cells requires computational systems that help engineers navigate numerous challenges, including the large number of distinct molecular species involved in extant cells and the lack of rigid structures or insulation between parts, which gives rise to an extraordinary number of molecular interactions.

One potential way to circumvent these challenges is to focus on cell-free systems, which promise to be easier to construct, control, test, and model. Laohakunakorn highlights these advantages and describes how the reduced complexity of cell-free systems could be a powerful training ground for model-driven design of cells. Due to the limitations of cell-free systems, many problems will likely require engineering complete synthetic cells. For example, cells would likely be easier to deploy in open environments than cell-free systems. Designing entire cells requires confronting the complexity of molecular biology. Marucci et al. describe the models that will be needed to tackle this challenge and how such models could revolutionize synthetic biology. Achieving such models will likely require the collaborative efforts of numerous modelers, experimentalists, and engineers, which in turn will likely require standards for exchanging information about synthetic biological systems. Foreseeing this need, McLaughlin et al. report a new version of the Synthetic Biology Open Language (SBOL) which is substantially easier to use. McLaughlin et al. anticipate that the new version of SBOL will accelerate the adoption of SBOL and, in turn, the collaborative development of more sophisticated synthetic biological systems.

COMPUTER-AIDED BIODESIGN BEYOND SINGLE CELLS

Most synthetic biology efforts to date have focused on the design of individual cells with basic functionalities (e.g., implementing basic logic). However, outside the lab cells rarely exist in isolation and their ability to interact through chemical signaling and the inherent heterogeneity in cellular states across a population due to environmental perturbations can act as a basis for important collective behaviors. These emergent properties need to be understood even for simple synthetic circuits to function reliably and can even be exploited to create more robust or scalable distributed biological computations. However, designing individual cells to exhibit desired population-level behaviors is challenging, requiring novel computational and theoretical approaches.

Gorochowski et al. propose that multi-agent modeling could serve as a design framework for engineering living collectives and offer a way to better understand the underlying causes and driving factors of emergent properties in protocellular systems, developmental programs, disease states and industrial bioprocesses. Karkaria et al. present additional examples where the engineering of monocultures in synthetic biology has

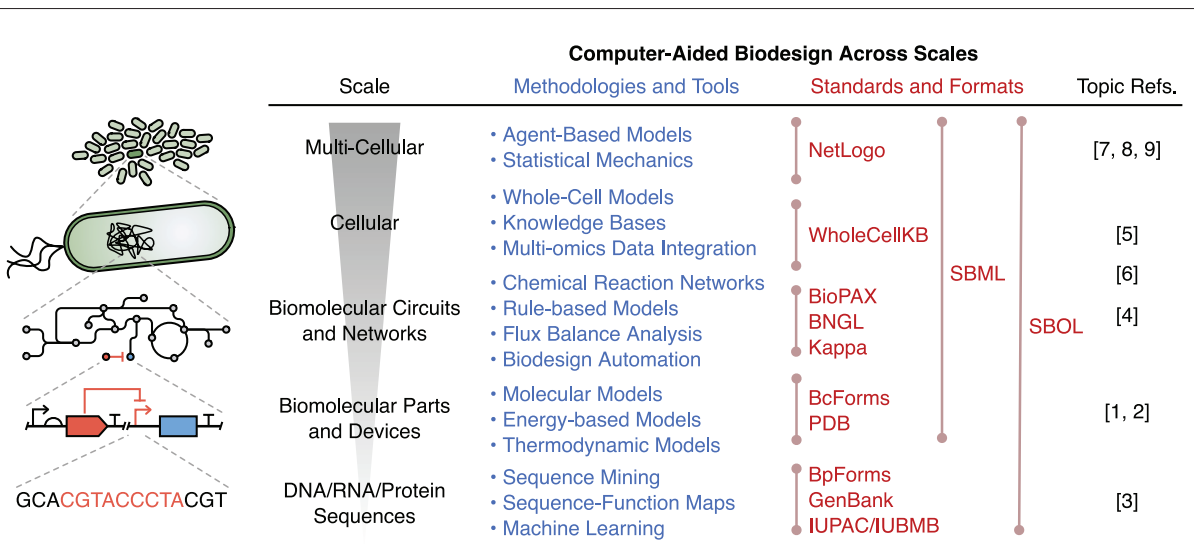


FIGURE 1 | Overview of computer-aided biodesign methodologies, tools, and standards used across scales. Although each methodology/tool has been placed at the specific scale it is commonly used, it should be noted that many can be applied across scales. Lines in the "Standards and Formats" section denote the range of scales over which the standard/format can be used. BioPAX, Biological Pathway Exchange; BNGL, BioNetGen Language; SBML, Systems Biology Markup Language; SBOL, Synthetic Biology Open Language; PDB, Protein Data Bank.

reached a bottleneck. Distributed computing is reviewed as a theoretical framework for understanding and designing distributed multicellular systems in biology to overcome these limitations. They consider a wide range of distributed algorithms used by biology covering the use of bet hedging, organism development and bacterial colony formation. Finally, Feliú et al. present original research spanning biological scales to understand and tune the patterns that emerge within a population of cells where each cell contains an identical synthetic oscillator circuit. They use computational modeling spanning multiple scales and show how a simplified cellular model coupled to varying environmental conditions can provide a convenient design tool that closely matches more complex multi-agent simulations.

CONCLUSION

Being able to scale our ability to harness biology will be crucial for addressing the many grand challenges we face, such as shifts toward sustainable manufacturing, clean energy production, and new forms of advanced medicine. CAD applied to synthetic biology is likely to play a key role in realizing these ambitions and the articles presented in this topic provide a broad introduction to CADs current role, in addition to a glimpse at its possible development and integration into the bioengineering practices of the future.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by a Royal Society University Research Fellowship (Grant no. UF160357; TG), a National Institutes of Health award (Grant no. R35GM119771; JK), a National Science Foundation award (Grant no. 1935372; JK), an EPSRC early career fellowship (Grant no. EP/S017542/1; FP), and BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre (Grant no. BB/L01386X/1; TG and FP).

Conflict of Interest: BY was employed by the companies Scientific Technologies Ltd. and Microsoft Research.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

PERSPECTIVE
published: 26 June 2020
doi: 10.3389/fbioe.2020.00705

Toward Engineering Biosystems With Emergent Collective Functions

Thomas E. Gorochowski[1], Sabine Hauert[2†], Jan-Ulrich Kreft[3†], Lucia Marucci[2†], Namid R. Stillman[2†], T.-Y. Dora Tang[4,5†], Lucia Bandiera[6], Vittorio Bartoli[2], Daniel O. R. Dixon[7], Alex J. H. Fedorec[8], Harold Fellermann[9], Alexander G. Fletcher[10], Tim Foster[3], Luca Giuggioli[2], Antoni Matyjaszkiewicz[11], Scott McCormick[2], Sandra Montes Olivas[2], Jonathan Naylor[9], Ana Rubio Denniss[2] and Daniel Ward[1]*

[1] School of Biological Sciences, University of Bristol, Bristol, United Kingdom, [2] Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom, [3] School of Biosciences and Institute of Microbiology and Infection and Centre for Computational Biology, University of Birmingham, Birmingham, United Kingdom, [4] Max Plank Institute of Molecular Cell Biology and Genetics, Dresden, Germany, [5] Physics of Life, Cluster of Excellence, Technische Universität Dresden, Dresden, Germany, [6] School of Engineering, University of Edinburgh, Edinburgh, United Kingdom, [7] School of Biochemistry, University of Bristol, Bristol, United Kingdom, [8] Division of Biosciences, University College London, London, United Kingdom, [9] School of Computing, Newcastle University, Newcastle upon Tyne, United Kingdom, [10] Bateson Centre and School of Mathematics and Statistics, University of Sheffield, Sheffield, United Kingdom, [11] The European Molecular Biology Laboratory, Barcelona, Spain*

OPEN ACCESS

Edited by:
Pablo Ivan Nikel, Novo Nordisk Foundation Center for Biosustainability (DTU Biosustain), Denmark

Reviewed by:
Dae-Hee Lee, Korea Research Institute of Bioscience and Biotechnology (KRIBB), South Korea
Irene Otero Muras, Consejo Superior de Investigaciones Científicas (CSIC), Spain

****Correspondence:***
Thomas E. Gorochowski
thomas.gorochowski@bristol.ac.uk

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *02 May 2020*
Accepted: *05 June 2020*
Published: *26 June 2020*

Citation:
Gorochowski TE, Hauert S, Kreft J-U, Marucci L, Stillman NR, Tang T-YD, Bandiera L, Bartoli V, Dixon DOR, Fedorec AJH, Fellermann H, Fletcher AG, Foster T, Giuggioli L, Matyjaszkiewicz A, McCormick S, Montes Olivas S, Naylor J, Rubio Denniss A and Ward D (2020) Toward Engineering Biosystems With Emergent Collective Functions. Front. Bioeng. Biotechnol. 8:705. doi: 10.3389/fbioe.2020.00705

Many complex behaviors in biological systems emerge from large populations of interacting molecules or cells, generating functions that go beyond the capabilities of the individual parts. Such collective phenomena are of great interest to bioengineers due to their robustness and scalability. However, engineering emergent collective functions is difficult because they arise as a consequence of complex multi-level feedback, which often spans many length-scales. Here, we present a perspective on how some of these challenges could be overcome by using multi-agent modeling as a design framework within synthetic biology. Using case studies covering the construction of synthetic ecologies to biological computation and synthetic cellularity, we show how multi-agent modeling can capture the core features of complex multi-scale systems and provide novel insights into the underlying mechanisms which guide emergent functionalities across scales. The ability to unravel design rules underpinning these behaviors offers a means to take synthetic biology beyond single molecules or cells and toward the creation of systems with functions that can only emerge from collectives at multiple scales.

Keywords: synthetic biology, multi-agent modeling, systems biology, emergence, multi-scale, bioengineering, consortia, collectives

INTRODUCTION

Many living organisms have evolved traits to exploit the capabilities that emerge from large interacting populations of molecules or cells, which go beyond those of the individual elements. From bacteria forming biofilms to fight antibiotic treatments to synchronizing their behaviors through quorum sensing during disease, emergent collective behaviors are pervasive in biology. Likewise, the engineering of emergent collective behaviors could offer an intriguing path to artificial biosystems with improved reliability, robustness and scalability. However, current approaches to biological design are ill-equipped for this task as they tend to focus on a single level of

organization and ignore potential feedbacks between different aspects/levels of a system. A common example is the design of transcriptional gene regulatory networks where it is assumed that the function of the entire system can be understood solely by the steady state input-output transcriptional response of genetic devices (Nielsen et al., 2016). While this simplification is useful and powerful in some cases, if the genes regulated link to metabolic processes there is a chance that feedback via metabolism could break circuit function. Focusing purely on transcriptional networks makes it impossible to capture such behaviors.

In physics, great strides have been made through techniques from statistical mechanics to understand emergent phenomena. These include the Ising model used to capture magnetic phase transitions (Taroni, 2015) and the application of renormalization to understand how physical and biological constraints might underpin scaling laws that guide evolution (West et al., 2002; Kempes et al., 2019). There has also been growing interest over the past few decades in the field of complexity theory (Nicolis and Prigogine, 1989) and whether laws might exist that govern self-organization and emergence across diverse types of complex system composed of many interacting parts (Prigogine and Nicolis, 1985; Ashby, 1991; Goldstein, 1999; West et al., 2002).

An approach to capture and explore the emergent features of complex systems is multi-agent modeling (also termed agent-based or individual-based modeling) (Hellweger et al., 2016). This considers key components of a system as explicit entities/agents and allows for large and diverse interacting populations of these (**Figure 1A**). Specifically, a multi-agent model consists of autonomous agents that represent the lowest level components of the system. Common types of agent in biological systems include molecules, cells and whole multicellular organisms. Each agent is assigned a specific set of rules governing how it interacts with other agents and the local environment. The way these rules are modeled is flexible with the option to use basic finite state-machines, Boolean logic governing stimuli-response relationships, or more complex representations like differential equation models (e.g., capturing the biochemical reaction networks within a cell). Populations of these agents are then placed in a simulated environment that encompasses physical processes of relevance to the system. In biology, this might include the diffusion of chemicals, the flow of fluids, and the mechanical forces that cells can exert on one another. Again, the way that these environmental processes are modeled can vary, resulting in a final model that could potentially combine stochastic, deterministic, dynamic, discrete and continuous representations for different aspects of a system. The integration of such diverse modeling approaches allows for the most appropriate form of representation to be used for each aspect and helps simplify the specification of the multi-scale system, but often comes at the cost of reduced analytical tractability. Even so, multi-scale modeling has been shown capable of discovering some of the core ingredients needed for collective behaviors to emerge (Hellweger et al., 2016; Gorochowski and Richardson, 2017), but its use to date in synthetic biology has been limited (Gorochowski, 2016).

Here, we aim to highlight some of the key areas of synthetic biology where multi-agent modeling offers a unique way to tackle longstanding problems (**Figure 1B**). While the examples we cover are diverse, they all share a core characteristic: the emergence of behaviors in the systems cannot be explained by looking solely at their basic parts in isolation. This essence makes such systems special yet difficult to engineer via traditional means. We propose to extend bioengineering methods to encompass principles gleaned from multi-agent models and use them to guide the design of synthetic biological systems displaying emergent phenomena. We end by discussing some of the practical challenges when using multi-agent modeling in synthetic biology and future directions for the marriage of these fields.

UNDERSTANDING THE EMERGENCE OF LIFE

When considering emergent phenomena, the quintessential example is the emergence of life. Putting aside the difficulty of defining precisely what life is, the ability of living systems to self-replicate and create order/information out of chaos is an inspiration for many engineers. Bottom-up synthetic biology attempts to build chemical systems that display life-like behaviors using a minimal set of components. The hope is that these simplified systems might help us understand how life emerged from first principles.

One attempt to reach this goal has been via the synthesis of artificial cells (protocells) with life-like properties. This requires the ability to bridge length scales by harnessing molecular self-assembly to create micron-sized compartments (Bayley et al., 2008; Li et al., 2014) and the intricate interactions between molecules and enzymes to form biochemical reaction networks (Hasty et al., 2002). The incorporation of these reaction networks within protocells has also been demonstrated (Adamala et al., 2017; Joesaar et al., 2019) and although chemically simple, such systems display an array of dynamical behaviors including pattern formation (Niederholtmeyer et al., 2015; Zadorin et al., 2017) and replication via controlled growth and division (Chen et al., 2004). By combining these systems with additional chemical modules and parts, this may offer a route to creating other key behaviors of living systems.

Building on these capabilities, functionalities can be scaled further by constructing systems composed of populations of protocells or through interacting natural and artificial cellular communities (Lentini et al., 2014; Adamala et al., 2017; Tang et al., 2018). While such extensions offer a promising platform for probing emergent behaviors using simple self-contained chemical units, it is difficult to know what parameters to engineer into these systems and the level of complexity required to drive a desired collective behavior. This is where multi-agent modeling, in combination with more traditional models of chemical reaction systems, could lead to a quantitative understanding of the key elements needed for the emergence of life-like behaviors. In particular, multi-agent models would allow for the rapid exploration of potential systems using physically

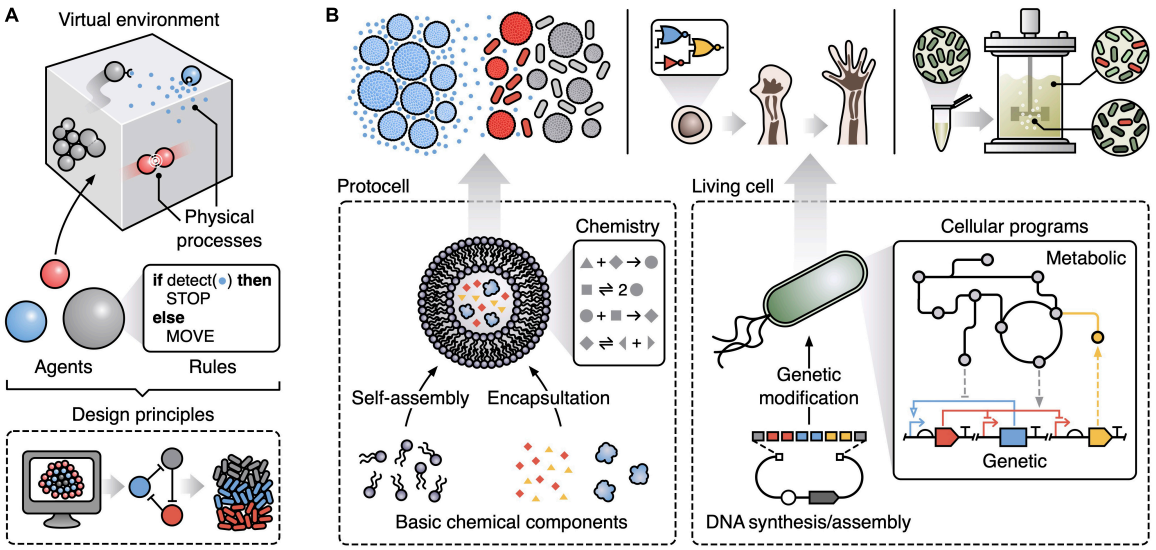


FIGURE 1 | Multi-agent modeling can support the design of emergent collective functions in synthetic biology. **(A)** Key components of a multi-agent model. Populations of autonomous agents following user-prescribed rules are placed in a virtual environment that simulates relevant physical processes (e.g., physical collisions, chemical diffusion, movement, and fluid flows). Simulations of multi-agent models can be used to derive design principles that capture the basic ingredients (e.g., specific patterns of interactions) needed for a particular emergent behavior. **(B)** Potential applications of multi-agent modeling within synthetic biology and the underlying agents (bottom, dashed boxes) used to generate specific emergent collective behaviors: (top left) exploring how to create life-like behaviors from basic chemical components with sender protocells (blue) able to spatially propagate a signal to receiver protocells and bacteria (gray when inactive, red when active) using a small diffusive chemical (small blue dots); (top middle) understanding the developmental programs used during morphogenesis as a step toward the creation of synthetic multi-cellular life; (top right) improving scale-up of microbial fermentations by accounting for heterogeneity across a bioreactor and designing engineered microbes able to robustly function under these conditions.

realistic parameters until the right combination of parts was found that resulted in a desired emergent functionality.

Historically, mathematical models developed using differential equations have proved effective for understanding the dynamics of minimal chemical systems (Rovinskii and Zhabotinskii, 1984) and are widely and successfully used for modeling all types of biological system (Ellner and Guckenheimer, 2011; Raue et al., 2013). Furthermore, the application of bifurcation analysis to these dynamical models enables the rigorous characterization of emergent phenomena such as bi-stability, symmetry breaking, non-linear oscillations, chaos, and pattern formation (Kuznetsov, 2004). However, while it is possible to use partial differential equations (PDEs) to capture spatial aspects of a system, the high levels of heterogeneity in the complex environments of many biological system (e.g., cellular tissues) and the ability of both agents and the rules to change over time, can make practical use of PDEs a challenge (Hellweger et al., 2016; Perez-Carrasco et al., 2016; Glen et al., 2019).

In comparison, multi-agent modeling is able to explicitly capture such variation and consider simplified rules to express internal chemical reactions altering specific characteristics of each component. Due to the chemical simplicity and programmability of minimal protocells, this abstraction is a good fit, allowing iterative refinement of model and experimental system. For example, due to the limited number of possible chemical reactions present in a minimal system, comprehensive direct measurements can be made to create highly predictive rules for how a protocell's chemical state will change over time. These can then drive simulations of accurate protocell behaviors in a multi-agent model to explore the specific combination of reactions required for the emergence of higher population-level functionalities. This two-way cycle of development would be difficult, if not impossible, when using natural cells where complex evolutionary baggage masks those features essential for emergence.

DISTRIBUTED COMPUTATION DURING DEVELOPMENT

Living cells continually monitor their environment and adapt their physiology in order to survive. This requires the processing of information gathered from sensors to make suitable changes to gene expression. Synthetic biology enables us to reprogram cells by writing our own genetic programs to exploit the cells' computational capabilities in new ways (Greco et al., 2019; Grozinger et al., 2019). So far, the majority of research in biological computation has revolved around the concept of genetic circuits and attempted to repurpose tools and methodologies from electronic circuit design (Nielsen et al., 2016; Gorochowski et al., 2017) and automatic verification (Dunn et al., 2014). While this approach has enabled the automated design of cellular programs able to perform basic logic, much of the information processing in native biological systems is distributed,

relying on collective decision making (e.g., quorum sensing) and interactions between large numbers of cells.

This feature is most evident in developmental biology where robust genetic programs must ensure that a complex multi-cellular organism emerges from a single cell. Cell growth, differentiation, migration and self-organization are coordinated by a developmental program with dynamics at both the intra- and inter-cellular levels. These enable the generation of precise deterministic patterns from stochastic underlying processes (Glen et al., 2019). In contrast to simple logic circuits, the complexity of the molecular interactions and mechanical forces underpinning these processes motivate the use of multi-agent modeling to better understand how developmental programs work in morphogenetic systems. In particular, multi-agent models are able to capture the role of cellular heterogeneity, proliferation and morphology, mechanical and environmental cues, movement of cells as well as the integration of multiple processes at diverse scales and the feedback between these (Montes-Olivas et al., 2019). Such models have helped deepen our understanding of early mammalian embryogenesis (Godwin et al., 2017), as well as the formation of vascular networks (Perfahl et al., 2017) and other complex structures and organs, including the skin, lung (Stopka et al., 2019), kidney (Lambert et al., 2018), and brain (Caffrey et al., 2014).

Although such work has provided insights into the computational architecture of native developmental programs, it has been difficult to apply this information to the creation of *de novo* morphogenetic systems because of a limited toolkit of parts available to build such systems. Synthetic biology may help solve this issue by facilitating the engineering of simplified multi-cellular systems (Velazquez et al., 2018) that implement developmental programs encompassing distributed feedback regulation (Ausländer and Fussenegger, 2016) and cell-to-cell communication (Bacchus et al., 2012), to better understand how these factors can be used to contribute to emergent self-organization (Morsut et al., 2016).

COLLECTIVE PHENOMENA DRIVING DISEASE

Many of the challenges treating diseases result from the malfunction of emergent multi-cellular properties, be it carcinogenesis (Deisboeck and Couzin, 2009; Ward et al., 2020), viral infection (Jacob et al., 2004), bacterial biofilm formation (Wu et al., 2020) or microbiome imbalances (Shreiner et al., 2015; Kumar et al., 2019). Multi-agent modeling of these conditions has helped demystify how the collective behavior of large numbers of diverse cells and their interactions with each other and their environment can lead to negative clinical outcomes.

Cancer is a complex multi-scale disease that includes environmental factors, genetic mutations and clonal selection, and complex interactions with the immune and vascular system. As a result, computational models of cancer need to account for many of these factors considering the heterogeneity and interactions of single cells, yet contain sufficient numbers of them to predict emergent phenomena at a tumor scale (Metzcar et al., 2019). Using this approach, multi-agent models have been used to help understand metastasis (Waclaw et al., 2015) and show that cancer cells with stem cell-like properties can be a key determinant in cancer progression with fatal consequences (Scott et al., 2016, 2019).

Beyond understanding the emergence of some diseases, multi-agent models can also identify novel ways of fixing their dynamics by considering how to disrupt cellular behaviors, and their interactions in space and time (Waclaw et al., 2015; Gallaher et al., 2018). Treatments themselves can even be designed to have collective emergent properties. For example, bacteria have already been engineered to use quorum sensing to trigger their delivery of drugs (Din et al., 2016) or they can be controlled using magnetic fields to penetrate cancerous tissue (Schuerle et al., 2019). Other collective behaviors used in cancer nanomedicine include self-assembly of nanoparticles to anchor imaging agents in tumors, disassembly of nanoparticles to increase tissue penetration, nanoparticles that compute the state of a tumor, nanoparticle-based remodeling of tumor environments to improve secondary nanoparticle transport, or nanoparticle signaling of tumor location to amplify the accumulation of nanoparticles in tumors (Hauert et al., 2013; Hauert and Bhatia, 2014).

The emergent properties inherent in many diseases, and the potential to harness such behaviors for new treatments, highlight the need for multi-scale modeling tools. Moreover, with the rapidly expanding field of "systems medicine," integrated modeling pipelines able to predict multi-scale disease dynamics and assess novel synthetic biology treatments via large-scale simulation and machine learning are positioned to revolutionize many areas of medicine (Stillman et al., 2020).

CHALLENGES IN SCALING-UP BIOTECHNOLOGY

The ability for synthetic biology to reprogram cellular metabolisms offers an opportunity to convert cheap substrates (or even waste) into valuable chemicals and materials via microbial fermentation (Nielsen and Keasling, 2016). To make this economically viable, large bioreactors are often used. However, while our use of fermentation stems back millennia (McGovern et al., 2004), we still struggle to reliably scale-up many processes from shake flasks in the lab to industrial-sized bioreactors (Lee and Kim, 2015).

A major reason for this problem is the increasing difficulty and power consumption of mixing or aerating reactors as their volume increases, causing pockets to form where nutrient concentration, temperature, oxygen, pH and other factors differ (Alvarez et al., 2005). As a microbe travels through the bioreactor, it becomes exposed to a wide variety of environments, each causing changes in its physiology. Because the path of each cell is unique, a population of cells will display a wide variety of physiological states. This differs from lab-scale experiments where environments are well-mixed and homogeneous, and causes predictions made from these conditions to significantly deviate from those observed during scale-up.

Capturing the combined environmental and cellular variability present in a large bioreactor is difficult using standard differential-equation models. In contrast, multi-agent models are able to explicitly capture and link gene regulation, metabolism, and the cells' local environment (Nieß et al., 2017; Haringa et al., 2018), as well as differences between individual cells and how cells change over time (González-Cabaleiro et al., 2017). In particular, hybrid models in which continuous descriptions of complex physical processes like fluid flows have been coupled with multi-agent models to allow for the efficient simulation of these systems. This approach can accurately predict the emergence of population heterogeneity and overall production rates and help guide bioreactor design to further improve yields (Haringa et al., 2018). Some attempts have also been made to use control engineering principles to design cellular systems able to adapt to fluctuating environments (Hsiao et al., 2018). To date, these attempts have mostly focused on the basic genetic parts and regulatory motifs (e.g., negative feedback) needed to implement control algorithms (Ceroni et al., 2018; Aoki et al., 2019; Pedone et al., 2019; Bartoli et al., 2020). Moving forward, multi-agent models offer a means to make simulations of these systems more realistic by accurately capturing how individual cells and their complex environment change over time.

Another challenge faced during large-scale fermentation is the opportunity for mutants to arise or unwanted microbes to contaminate a process and out-compete their engineered counterparts (Kazamia et al., 2012; Louca and Doebeli, 2016). Multi-agent models of these complex environments and local competition when multiple types of organism are present, could help support the development of evolutionarily stable strategies (ESSs) that prevent the replacement of an engineered population by competitors (Schuster et al., 2010).

ENGINEERING SYNTHETIC ECOLOGIES

At an even larger organizational level, synthetic biologists have begun to explore how to engineer interactions between communities to enable the future construction of synthetic ecologies (Ben Said and Or, 2017). With climate change, pollution and many other factors leading to the degradation of ecological systems, understanding how these systems emerge and function is crucial. Such knowledge would allow for effective restoration strategies (Solé et al., 2015) and potentially offer means to terraform other planets like Mars for future human inhabitation (Conde-Pueyo et al., 2020).

These applications require an understanding of how diverse organisms interact to create stable communities (Widder et al., 2016). This is difficult because the interactions that take place at the level of a population are governed by choices made by single organisms (Kreft et al., 2017). By using multi-agent modeling to rapidly test combinations of cell types, behaviors and interactions, and synthetic biology tools to engineer real-world microbial communities, it might become possible to design and test hypotheses regarding the principles for robust ecosystem design. For example, multi-agent modeling has been used to help understand how signaling and mutual cooperation can stabilize microbial communities (Kerényi et al., 2013). Furthermore, from a synthetic biology perspective many of the tools needed to engineer these systems already exist, e.g., biological parts able to implement cooperation (Shou et al., 2007), signaling (Bacchus et al., 2012), targeted death (Fedorec et al., 2019), and collective decision making (e.g., quorum sensing).

Beyond engineering interactions between organisms, spatial structure can also play a crucial role in the functionalities of microbial communities. Multi-agent modeling has demonstrated the significant impact that spatio-temporal organization can have on soil microbes and the success of auxotrophic interactions (Jiang et al., 2018). Such interactions are particularly important for engineering minimal functional synthetic communities as plant seed treatments and for vertical farming under defined conditions. In this context, whether or not a single cell or division of labor is the evolutionarily stable solution depends on the metabolic flux through the system, with high flux favoring division of labor (Kreft et al., 2020). Extending this modeling approach further to consider the thermodynamics of microbial growth and redox biochemistry could help ensure that resultant systems are ecologically and evolutionarily stable (Zerfaß et al., 2018). Alternatively, external control of the environment could be used to forcibly maintain a desired community structure (Treloar et al., 2020). In all cases, a combination of multi-agent modeling and engineerable biological systems provides a unique means to unravel how these complex systems function.

External feedback control has been proposed as another approach to control of cellular communities. By employing real-time single cell measurements (e.g., by time-lapse microscopy or flow-cytometry) and experimental systems able to send control signals to the cells via optogenetics (Toettcher et al., 2011) or chemical release in microfluidics (Menolascina et al., 2014), a computer can monitor and signal to a population of cells in order to maintain a desired behavior (e.g., the expression rate of a protein). More recently, it has been proposed to implement these control algorithms directly into cells, with the key aim of distributing tasks among different strains (Fiore et al., 2017; McCardell et al., 2017). Multi-agent modeling can be instrumental in the design of robust feedback mechanisms across multicellular populations, as it can reveal non-obvious effects of cell density, proliferation dynamics and spatial constraints on the effectiveness of control actions (Fiore et al., 2017).

DISCUSSION

We have shown how multi-agent models can be applied to many areas of synthetic biology. The core features of these models provide insight into some of the basic building blocks and mechanisms needed for collective behaviors to emerge and, we believe, may offer a means to support the future predictive design of collective behaviors.

A major hurdle to the widespread use of multi-agent modeling is the need to define and simulate complex models (Grimm et al., 2006). Although computational frameworks have been available since the 1980s to support this process, it is only during the past decade that tools have been

tailored for synthetic biology applications and reached sufficient performance (Gorochowski et al., 2012; Oishi and Klavins, 2014; Goñi-Moreno and Amos, 2015). More recently, the effective use of highly parallel computing resources has expanded the complexity of biological models that can be simulated (Rudge et al., 2012; Naylor et al., 2017; Li et al., 2019; Cooper et al., 2020). Automated coarse-graining of representations enable faster simulation without impacting on the accuracy of predictions (Graham et al., 2017), while advanced tools allow verification, validation and uncertainty quantification for such simulations (Richardson et al., 2020).

Improved simulations do not only speed up the time to an answer but may open up opportunities to create new types of computational design environments. For example, high-performance models coupled to virtual reality allow for multiple researchers to interactively manipulate a system and immediately observe the outcomes of their design decisions. Such capabilities have already begun to be used for molecular design (O'Connor et al., 2018) and when coupled to machine learning, offer a unique setting in which to explore complex high-dimensional datasets that are common in biology. They also allow for essential features to be distilled that can then be used to guide predictive design. Furthermore, hybrid approaches become possible where computational models dynamically augment an experimental setup by controlling physical features such as light (Rubio Denniss et al., 2019) or magnetism (Carlsen et al., 2014). If agents within the experimental system are responsive to these stimuli, then various forms of interaction can be externally programmed and rapidly explored to better understand the necessary conditions for a particular collective behavior to emerge. Once a desired set of rules for the interactions is found, the agents can be modified to implement these autonomously, removing the need for external control.

As synthetic biology moves beyond simple parts and circuits, and toward large-scale/multicellular systems, the available repertoire of design tools must also expand to support new requirements. Multi-agent modeling is perfectly placed to help make this leap and usher in new biological design methods focused on the engineering of emergent collective behaviors. Not only will this allow functionalities to span length scales, but it will also provide a way to engineer across the organizational levels of life through hierarchical composition of multi-scale models, from basic molecules and cells through to entire ecosystems.

AUTHOR CONTRIBUTIONS

TG, SH, J-UK, LM, NS, and T-YT wrote the manuscript. All other authors helped with editing and provided the feedback.

FUNDING

This work captured discussions between participants at the "Multi-agent modeling meets synthetic biology" workshop held on the 16–17 May 2019 at the University of Bristol, UK and funded by BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre (Grant No. BB/L01386X/1). TG was supported by a Royal Society University Research Fellowship (Grant No. UF160357). DD and VB were supported by the University of Bristol and the EPSRC & BBSRC Centre for Doctoral Training in Synthetic Biology (Grant No. EP/L016494/1). LB was supported by EPSRC (Grant No. EP/P017134/1-CONDSYC). AF received funding from the European Research Council under the European Union's Horizon 2020 Research and Innovation Programme (Grant No. 770835). LM was supported by the Medical Research Council (Grant No. MR/N021444/1), and the Engineering and Physical Sciences Research Council (Grant Nos. EP/R041695/1 and EP/S01876X/1). SM was supported by a Mexico Consejo Nacional de Ciencia y Tecnología (CONACYT) Ph.D. scholarship. TF and J-UK are grateful to the UK National Centre for the Replacement, Refinement & Reduction of Animals in Research (NC3Rs) for funding their development of individual-based models (IBMs) for the gut environment (eGUT Grant No. NC/K000683/1 and Ph.D. training Grant No. NC/R001707/1). SH, NS, and SM received funding from the European Union's Horizon 2020 FET Open programme (Grant No. 800983). T-YT acknowledged financial support from the MaxSynBio Consortium (jointly funded by the Federal Ministry of Education and Research, Germany and the Max Planck Society) and the MPI—CBG and the Cluster of Excellence Physics of Life of TU Dresden and EXC-1056 for funding.

REFERENCES

Adamala, K. P., Martin-Alarcon, D. A., Guthrie-Honea, K. R., and Boyden, E. S. (2017). Engineering genetic circuit interactions within and between synthetic minimal cells. *Nat. Chem.* 9, 431–439. doi: 10.1038/nchem.2644

Alvarez, M. M., Guzmán, A., and Elías, M. (2005). Experimental visualization of mixing pathologies in laminar stirred tank bioreactors. *Chem. Eng. Sci.* 60, 2449–2457. doi: 10.1016/j.ces.2004.11.049

Aoki, S. K., Lillacci, G., Gupta, A., Baumschlager, A., Schweingruber, D., and Khammash, M. (2019). A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* 570, 533–537. doi: 10.1038/s41586-019-1321-1

Ashby, W. R. (1991). "Principles of the self-organizing system," in *Facets of Systems Science*, ed. G. J. Klir (Boston, MA: Springer US), 521–536. doi: 10.1007/978-1-4899-0718-9_38

Ausländer, S., and Fussenegger, M. (2016). Engineering gene circuits for mammalian cell–based applications. *Cold Spring Harb. Perspect. Biol.* 8:a023895. doi: 10.1101/cshperspect.a023895

Bacchus, W., Lang, M., El-Baba, M. D., Weber, W., Stelling, J., and Fussenegger, M. (2012). Synthetic two-way communication between mammalian cells. *Nat. Biotechnol.* 30, 991–996. doi: 10.1038/nbt.2351

Bartoli, V., Meaker, G. A., di Bernardo, M., and Gorochowski, T. E. (2020). Tunable genetic devices through simultaneous control of transcription and translation. *Nat. Commun.* 11:2095. doi: 10.1038/s41467-020-15653-7

Bayley, H., Cronin, B., Heron, A., Holden, M. A., Hwang, W. L., Syeda, R., et al. (2008). Droplet interface bilayers. *Mol. Biosyst.* 4, 1191–1208. doi: 10.1039/B808893D

Ben Said, S., and Or, D. (2017). Synthetic microbial ecology: engineering habitats for modular consortia. *Front. Microbiol.* 8:1125. doi: 10.3389/fmicb.2017.01125

Caffrey, J. R., Hughes, B. D., Britto, J. M., and Landman, K. A. (2014). An in silico agent-based model demonstrates Reelin function in directing lamination of neurons during cortical development. *PLoS One* 9:e110415. doi: 10.1371/journal.pone.0110415

Carlsen, R. W., Edwards, M. R., Zhuang, J., Pacoret, C., and Sitti, M. (2014). Magnetic steering control of multi-cellular bio-hybrid microswimmers. *Lab. Chip* 14, 3850–3859. doi: 10.1039/C4LC00707G

Ceroni, F., Boo, A., Furini, S., Gorochowski, T. E., Borkowski, O., Ladak, Y. N., et al. (2018). Burden-driven feedback control of gene expression. *Nat. Methods* 15, 387–393. doi: 10.1038/nmeth.4635

Chen, I. A., Roberts, R. W., and Szostak, J. W. (2004). The emergence of competition between model protocells. *Science* 305:1474. doi: 10.1126/science.1100757

Conde-Pueyo, N., Vidiella, B., Sardanyés, J., Berdugo, M., Maestre, T. F., de Lorenzo, V., et al. (2020). Synthetic biology for terraformation lessons from mars, earth, and the microbiome. *Life* 10:14. doi: 10.3390/life10020014

Cooper, F. R., Baker, R. E., Bernabeu, M. O., Bordas, R., Bowler, L., Bueno-Orovio, A., et al. (2020). Chaste: cancer, heart and soft tissue environment. *J. Open Source Softw.* 5:1848. doi: 10.21105/joss.01848

Deisboeck, T. S., and Couzin, I. D. (2009). Collective behavior in cancer cell populations. *BioEssays* 31, 190–197. doi: 10.1002/bies.200800084

Din, M. O., Danino, T., Prindle, A., Skalak, M., Selimkhanov, J., Allen, K., et al. (2016). Synchronized cycles of bacterial lysis for in vivo delivery. *Nature* 536, 81–85. doi: 10.1038/nature18930

Dunn, S.-J., Martello, G., Yordanov, B., Emmott, S., and Smith, A. G. (2014). Defining an essential transcription factor program for naïve pluripotency. *Science* 344:1156. doi: 10.1126/science.1248882

Ellner, S. P., and Guckenheimer, J. (2011). *Dynamic Models in Biology*. Princeton, NJ: Princeton University Press.

Fedorec, A. J. H., Karkaria, B. D., Sulu, M., and Barnes, C. P. (2019). Killing in response to competition stabilises synthetic microbial consortia. *bioRxiv* [Preprint]. doi: 10.1101/2019.12.23.887331

Fiore, G., Matyjaszkiewicz, A., Annunziata, F., Grierson, C., Savery, N. J., Marucci, L., et al. (2017). In-Silico analysis and implementation of a multicellular feedback control strategy in a synthetic bacterial consortium. *ACS Synth. Biol.* 6, 507–517. doi: 10.1021/acssynbio.6b00220

Gallaher, J. A., Enriquez-Navas, P. M., Luddy, K. A., Gatenby, R. A., and Anderson, A. R. A. (2018). Spatial heterogeneity and evolutionary dynamics modulate time to recurrence in continuous and adaptive cancer therapies. *Cancer Res.* 78:2127. doi: 10.1158/0008-5472.CAN-17-2649

Glen, C. M., Kemp, M. L., and Voit, E. O. (2019). Agent-based modeling of morphogenetic systems: advantages and challenges. *PLoS Comput. Biol.* 15:e1006577. doi: 10.1371/journal.pcbi.1006577

Godwin, S., Ward, D., Pedone, E., Homer, M., Fletcher, A. G., and Marucci, L. (2017). An extended model for culture-dependent heterogenous gene expression and proliferation dynamics in mouse embryonic stem cells. *Npj Syst. Biol. Appl.* 3:19. doi: 10.1038/s41540-017-0020-5

Goldstein, J. (1999). Emergence as a construct: history and issues. *Emergence* 1, 49–72. doi: 10.1207/s15327000em0101_4

Goñi-Moreno, A., and Amos, M. (2015). "DiSCUS: a simulation platform for conjugation computing," in *Unconventional Computation and Natural Computation*, eds C. S. Calude and M. J. Dinneen (Cham: Springer International Publishing), 181–191. doi: 10.1007/978-3-319-21819-9_13

González-Cabaleiro, R., Mitchell, A. M., Smith, W., Wipat, A., and Ofiţeru, I. D. (2017). Heterogeneity in pure microbial systems: experimental measurements and modeling. *Front. Microbiol.* 8:1813. doi: 10.3389/fmicb.2017.01813

Gorochowski, T. E. (2016). Agent-based modelling in synthetic biology. *Essays Biochem.* 60:325. doi: 10.1042/EBC20160037

Gorochowski, T. E., Espah Borujeni, A., Park, Y., Nielsen, A. A., Zhang, J., Der, B. S., et al. (2017). Genetic circuit characterization and debugging using RNA-seq. *Mol. Syst. Biol.* 13:952. doi: 10.15252/msb.20167461

Gorochowski, T. E., Matyjaszkiewicz, A., Todd, T., Oak, N., Kowalska, K., Reid, S., et al. (2012). BSim: an agent-based tool for modeling bacterial populations in systems and synthetic biology. *PLoS One* 7:e42790. doi: 10.1371/journal.pone.0042790

Gorochowski, T. E., and Richardson, T. O. (2017). "How behaviour and the environment influence transmission in mobile groups," in *Temporal Network Epidemiology*, eds N. Masuda and P. Holme (Singapore: Springer), 17–42. doi: 10.1007/978-981-10-5287-3_2

Graham, J. A., Essex, J. W., and Khalid, S. (2017). PyCGTOOL: automated generation of coarse-grained molecular dynamics models from atomistic trajectories. *J. Chem. Inf. Model.* 57, 650–656. doi: 10.1021/acs.jcim.7b00096

Greco, F. V., Tarnowski, M. J., and Gorochowski, T. E. (2019). Living computers powered by biochemistry. *Biochemist* 41, 14–18. doi: 10.1042/bio04103014

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., et al. (2006). A standard protocol for describing individual-based and agent-based models. *Ecol. Model.* 198, 115–126. doi: 10.1016/j.ecolmodel.2006.04.023

Grozinger, L., Amos, M., Gorochowski, T. E., Carbonell, P., Oyarzún, D. A., Stoof, R., et al. (2019). Pathways to cellular supremacy in biocomputing. *Nat. Commun.* 10:5250. doi: 10.1038/s41467-019-13232-z

Haringa, C., Tang, W., Wang, G., Deshmukh, A. T., van Winden, W. A., Chu, J., et al. (2018). Computational fluid dynamics simulation of an industrial *P. chrysogenum* fermentation with a coupled 9-pool metabolic model: towards rational scale-down and design optimization. *Chem. Eng. Sci.* 175, 12–24. doi: 10.1016/j.ces.2017.09.020

Hasty, J., McMillen, D., and Collins, J. J. (2002). Engineered gene circuits. *Nature* 420, 224–230. doi: 10.1038/nature01257

Hauert, S., Berman, S., Nagpal, R., and Bhatia, S. N. (2013). A computational framework for identifying design guidelines to increase the penetration of targeted nanoparticles into tumors. *Nano Today* 8, 566–576. doi: 10.1016/j.nantod.2013.11.001

Hauert, S., and Bhatia, S. N. (2014). Mechanisms of cooperation in cancer nanomedicine: towards systems nanotechnology. *Trends Biotechnol.* 32, 448–455. doi: 10.1016/j.tibtech.2014.06.010

Hellweger, F. L., Clegg, R. J., Clark, J. R., Plugge, C. M., and Kreft, J.-U. (2016). Advancing microbial sciences by individual-based modelling. *Nat. Rev. Microbiol.* 14, 461–471. doi: 10.1038/nrmicro.2016.62

Hsiao, V., Swaminathan, A., and Murray, R. M. (2018). Control theory for synthetic biology: recent advances in system characterization, control design, and controller implementation for synthetic biology. *IEEE Control Syst. Mag.* 38, 32–62. doi: 10.1109/mcs.2018.2810459

Jacob, C., Litorco, J., and Lee, L. (2004). "Immunity through swarms: agent-based simulations of the human immune system," in *Artificial Immune Systems*, eds G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis (Berlin: Springer), 400–412. doi: 10.1007/978-3-540-30220-9_32

Jiang, X., Zerfaß, C., Feng, S., Eichmann, R., Asally, M., Schäfer, P., et al. (2018). Impact of spatial organization on a novel auxotrophic interaction among soil microbes. *ISME J.* 12, 1443–1456. doi: 10.1038/s41396-018-0095-z

Joesaar, A., Yang, S., Bögels, B., van der Linden, A., Pieters, P., Kumar, B. V. V. S. P., et al. (2019). DNA-based communication in populations of synthetic protocells. *Nat. Nanotechnol.* 14, 369–378. doi: 10.1038/s41565-019-0399-9

Kazamia, E., Aldridge, D. C., and Smith, A. G. (2012). Synthetic ecology – A way forward for sustainable algal biofuel production? *Photosynth. Microorg. Bio Fuel Prod. Sun Light* 162, 163–169. doi: 10.1016/j.jbiotec.2012.03.022

Kempes, C. P., Koehl, M. A. R., and West, G. B. (2019). The scales that limit: the physical boundaries of evolution. *Front. Ecol. Evol.* 7:242. doi: 10.3389/fevo.2019.00242

Kerényi, Á, Bihary, D., Venturi, V., and Pongor, S. (2013). Stability of multispecies bacterial communities: signaling networks may stabilize microbiomes. *PLoS One* 8:e57947. doi: 10.1371/journal.pone.0057947

Kreft, J.-U., Griffin, B. M., and González-Cabaleiro, R. (2020). Evolutionary causes and consequences of metabolic division of labour: why anaerobes do and aerobes don't. *Curr. Opin. Biotechnol.* 62, 80–87. doi: 10.1016/j.copbio.2019.08.008

Kreft, J.-U., Plugge, C. M., Prats, C., Leveau, J. H. J., Zhang, W., and Hellweger, F. L. (2017). From genes to ecosystems in microbiology: modeling approaches and the importance of individuality. *Front. Microbiol.* 8:2299. doi: 10.3389/fmicb.2017.02299

Kumar, M., Ji, B., Zengler, K., and Nielsen, J. (2019). Modelling approaches for studying the microbiome. *Nat. Microbiol.* 4, 1253–1267. doi: 10.1038/s41564-019-0491-9

Kuznetsov, Y. A. (2004). *Elements of Applied Bifurcation Theory*. New York, NY: Springer.

Lambert, B., MacLean, A. L., Fletcher, A. G., Combes, A. N., Little, M. H., and Byrne, H. M. (2018). Bayesian inference of agent-based models: a tool for studying kidney branching morphogenesis. *J. Math. Biol.* 76, 1673–1697. doi: 10.1007/s00285-018-1208-z

Lee, S. Y., and Kim, H. U. (2015). Systems strategies for developing industrial microbial strains. *Nat. Biotechnol.* 33, 1061–1072. doi: 10.1038/nbt.3365

Lentini, R., Santero, S. P., Chizzolini, F., Cecchi, D., Fontana, J., Marchioretto, M., et al. (2014). Integrating artificial with natural cells to translate chemical messages that direct E. coli behaviour. *Nat. Commun.* 5:4012. doi: 10.1038/ncomms5012

Li, B., Taniguchi, D., Gedara, J. P., Gogulancea, V., Gonzalez-Cabaleiro, R., Chen, J., et al. (2019). NUFEB: a massively parallel simulator for individual-based modelling of microbial communities. *PLoS Comput. Biol.* 15:e1007125. doi: 10.1371/journal.pcbi.1007125

Li, M., Huang, X., Tang, T.-Y. D., and Mann, S. (2014). Synthetic cellularity based on non-lipid micro-compartments and protocell models. *Synth. Biol. Synth. Biomol.* 22, 1–11. doi: 10.1016/j.cbpa.2014.05.018

Louca, S., and Doebeli, M. (2016). Transient dynamics of competitive exclusion in microbial communities. *Environ. Microbiol.* 18, 1863–1874. doi: 10.1111/1462-2920.13058

McCardell, R. D., Huang, S., Green, L. N., and Murray, R. M. (2017). Control of bacterial population density with population feedback and molecular sequestration. *bioRxiv* [Preprint]. doi: 10.1101/225045

McGovern, P. E., Zhang, J., Tang, J., Zhang, Z., Hall, G. R., Moreau, R. A., et al. (2004). Fermented beverages of pre- and proto-historic China. *Proc. Natl. Acad. Sci. U.S.A.* 101:17593. doi: 10.1073/pnas.0407921102

Menolascina, F., Fiore, G., Orabona, E., De Stefano, L., Ferry, M., Hasty, J., et al. (2014). In-Vivo real-time control of protein expression from endogenous and synthetic gene networks. *PLoS Comput. Biol.* 10:e1003625. doi: 10.1371/journal.pcbi.1003625

Metzcar, J., Wang, Y., Heiland, R., and Macklin, P. (2019). A review of cell-based computational modeling in cancer biology. *JCO Clin. Cancer Inform.* 3, 1–13. doi: 10.1200/CCI.18.00069

Montes-Olivas, S., Marucci, L., and Homer, M. (2019). Mathematical models of organoid cultures. *Front. Genet.* 10:873. doi: 10.3389/fgene.2019.00873

Morsut, L., Roybal, K. T., Xiong, X., Gordley, R. M., Coyle, S. M., Thomson, M., et al. (2016). Engineering customized cell sensing and response behaviors using synthetic notch receptors. *Cell* 164, 780–791. doi: 10.1016/j.cell.2016.01.012

Naylor, J., Fellermann, H., Ding, Y., Mohammed, W. K., Jakubovics, N. S., Mukherjee, J., et al. (2017). Simbiotics: a multiscale integrative platform for 3D modeling of bacterial populations. *ACS Synth. Biol.* 6, 1194–1210. doi: 10.1021/acssynbio.6b00315

Nicolis, G., and Prigogine, I. (1989). *Exploring Complexity: An Introduction*. New York, NY: W.H. Freeman and Company.

Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., et al. (2015). Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* 4:e09771. doi: 10.7554/eLife.09771

Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., et al. (2016). Genetic circuit design automation. *Science* 352:aac7341. doi: 10.1126/science.aac7341

Nielsen, J., and Keasling, J. D. (2016). Engineering cellular metabolism. *Cell* 164, 1185–1197. doi: 10.1016/j.cell.2016.02.004

Nieß, A., Löffler, M., Simen, J. D., and Takors, R. (2017). Repetitive short-term stimuli imposed in poor mixing zones induce long-term adaptation of *E. coli* cultures in large-scale bioreactors: experimental evidence and mathematical model. *Front. Microbiol.* 8:1195. doi: 10.3389/fmicb.2017.01195

O'Connor, M., Deeks, H. M., Dawn, E., Metatla, O., Roudaut, A., Sutton, M., et al. (2018). Sampling molecular conformations and dynamics in a multiuser virtual reality framework. *Sci. Adv.* 4:eaat2731. doi: 10.1126/sciadv.aat2731

Oishi, K., and Klavins, E. (2014). Framework for engineering finite state machines in gene regulatory networks. *ACS Synth. Biol.* 3, 652–665. doi: 10.1021/sb4001799

Pedone, E., Postiglione, L., Aulicino, F., Rocca, D. L., Montes-Olivas, S., Khazim, M., et al. (2019). A tunable dual-input system for on-demand dynamic gene expression regulation. *Nat. Commun.* 10:4481. doi: 10.1038/s41467-019-12329-9

Perez-Carrasco, R., Guerrero, P., Briscoe, J., and Page, K. M. (2016). Intrinsic noise profoundly alters the dynamics and steady state of morphogen-controlled bistable genetic switches. *PLoS Comput. Biol.* 12:e1005154. doi: 10.1371/journal.pcbi.1005154

Perfahl, H., Hughes, B. D., Alarcón, T., Maini, P. K., Lloyd, M. C., Reuss, M., et al. (2017). 3D hybrid modelling of vascular network formation. *J. Theor. Biol.* 414, 254–268. doi: 10.1016/j.jtbi.2016.11.013

Prigogine, I., and Nicolis, G. (1985). "Self-organisation in nonequilibrium systems: towards a dynamics of complexity," in *Bifurcation Analysis: Principles, Applications and Synthesis*, eds M. Hazewinkel, R. Jurkovich, and J. H. P. Paelinck (Dordrecht: Springer Netherlands), 3–12. doi: 10.1007/978-94-009-6239-2_1

Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelke, M., Kaschek, D., et al. (2013). lessons learned from quantitative dynamical modeling in systems biology. *PLoS One* 8:e74335. doi: 10.1371/journal.pone.0074335

Richardson, R. A., Wright, D. W., Edeling, W., Jancauskas, V., Lakhlili, J., and Coveney, P. V. (2020). EasyVVUQ: a library for verification, validation and uncertainty quantification in high performance computing. *J. Open Res. Softw.* 8:11. doi: 10.5334/jors.303

Rovinskii, A. B., and Zhabotinskii, A. M. (1984). Mechanism and mathematical model of the oscillating bromate-ferroin-bromomalonic acid reaction. *J. Phys. Chem.* 88, 6081–6084. doi: 10.1021/j150669a001

Rubio Denniss, A. M., Gorochowski, T. E., and Hauert, S. (2019). "Augmented reality for the engineering of collective behaviours in microsystems," in *Proceedings of the 2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, Helsinki.

Rudge, T. J., Steiner, P. J., Phillips, A., and Haseloff, J. (2012). Computational modeling of synthetic microbial biofilms. *ACS Synth. Biol.* 1, 345–352. doi: 10.1021/sb300031n

Schuerle, S., Soleimany, A. P., Yeh, T., Anand, G. M., Häberli, M., Fleming, H. E., et al. (2019). Synthetic and living micropropellers for convection-enhanced nanoparticle transport. *Sci. Adv.* 5:eaav4803. doi: 10.1126/sciadv.aav4803

Schuster, S., Kreft, J.-U., Brenner, N., Wessely, F., Theißen, G., Ruppin, E., et al. (2010). Cooperation and cheating in microbial exoenzyme production – Theoretical analysis for biotechnological applications. *Biotechnol. J.* 5, 751–758. doi: 10.1002/biot.200900303

Scott, J. G., Fletcher, A. G., Anderson, A. R. A., and Maini, P. K. (2016). Spatial metrics of tumour vascular organisation predict radiation efficacy in a computational model. *PLoS Comput. Biol.* 12:e1004712. doi: 10.1371/journal.pcbi.1004712

Scott, J. G., Maini, P. K., Anderson, A. R., and Fletcher, A. G. (2019). Inferring tumor proliferative organization from phylogenetic tree measures in a computational model. *Syst. Biol.* syz070. doi: 10.1093/sysbio/syz070

Shou, W., Ram, S., and Vilar, J. M. G. (2007). Synthetic cooperation in engineered yeast populations. *Proc. Natl. Acad. Sci. U.S.A.* 104:1877. doi: 10.1073/pnas.0610575104

Shreiner, A. B., Kao, J. Y., and Young, V. B. (2015). The gut microbiome in health and in disease. *Curr. Opin. Gastroenterol.* 31, 69–75.

Solé, R. V., Montañez, R., and Duran-Nebreda, S. (2015). Synthetic circuit designs for earth terraformation. *Biol. Direct.* 10:37. doi: 10.1186/s13062-015-0064-7

Stillman, N. R., Kovacevic, M., Balaz, I., and Hauert, S. (2020). In silico modelling of cancer nanomedicine, across scales and transport barriers. *Npj Comput. Mater.* (in press).

Stopka, A., Kokic, M., and Iber, D. (2019). Cell-based simulations of biased epithelial lung growth. *Phys. Biol.* 17:016006. doi: 10.1088/1478-3975/ab5613

Tang, T.-Y. D., Cecchi, D., Fracasso, G., Accardi, D., Coutable-Pennarun, A., Mansy, S. S., et al. (2018). Gene-mediated chemical communication in synthetic protocell communities. *ACS Synth. Biol.* 7, 339–346. doi: 10.1021/acssynbio.7b00306

Taroni, A. (2015). 90 years of the Ising model. *Nat. Phys.* 11, 997–997. doi: 10.1038/nphys3595

Toettcher, J. E., Gong, D., Lim, W. A., and Weiner, O. D. (2011). Light-based feedback for controlling intracellular signaling dynamics. *Nat. Methods* 8, 837–839. doi: 10.1038/nmeth.1700

Treloar, N. J., Fedorec, A. J. H., Ingalls, B., and Barnes, C. P. (2020). Deep reinforcement learning for the control of microbial co-cultures in bioreactors. *PLoS Comput. Biol.* 16:e1007783. doi: 10.1371/journal.pcbi.1007783

Velazquez, J. J., Su, E., Cahan, P., and Ebrahimkhani, M. R. (2018). Programming morphogenesis through systems and synthetic biology. *Trends Biotechnol.* 36, 415–429. doi: 10.1016/j.tibtech.2017.11.003

Waclaw, B., Bozic, I., Pittman, M. E., Hruban, R. H., Vogelstein, B., and Nowak, M. A. (2015). A spatial model predicts that dispersal and cell turnover limit intratumour heterogeneity. *Nature* 525, 261–264. doi: 10.1038/nature14971

Ward, D., Montes Olivas, S., Fletcher, A., Homer, M., and Marucci, L. (2020). Cross-talk between Hippo and Wnt signalling pathways in intestinal crypts: insights from an agent-based model. *Comput. Struct. Biotechnol. J.* 18, 230–240. doi: 10.1016/j.csbj.2019.12.015

West, G. B., Woodruff, W. H., and Brown, J. H. (2002). Allometric scaling of metabolic rate from molecules and mitochondria to cells and mammals. *Proc. Natl. Acad. Sci. U.S.A.* 99:2473. doi: 10.1073/pnas.012579799

Widder, S., Allen, R. J., Pfeiffer, T., Curtis, T. P., Wiuf, C., Sloan, W. T., et al. (2016). Challenges in microbial ecology: building predictive understanding of community function and dynamics. *ISME J.* 10, 2557–2568. doi: 10.1038/ismej.2016.45

Wu, S., Liu, J., Liu, C., Yang, A., and Qiao, J. (2020). Quorum sensing for population-level control of bacteria and potential therapeutic applications. *Cell. Mol. Life Sci.* 77, 1319–1343. doi: 10.1007/s00018-019-03326-8

Zadorin, A. S., Rondelez, Y., Gines, G., Dilhas, V., Urtel, G., Zambrano, A., et al. (2017). Synthesis and materialization of a reaction–diffusion French flag pattern. *Nat. Chem.* 9, 990–996. doi: 10.1038/nchem.2770

Zerfaß, C., Chen, J., and Soyer, O. S. (2018). Engineering microbial communities using thermodynamic principles and electrical interfaces. *Energy Biotechnol. Environ. Biotechnol.* 50, 121–127. doi: 10.1016/j.copbio.2017.12.004

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

REVIEW
published: 22 July 2020
doi: 10.3389/fbioe.2020.00834

From Microbial Communities to Distributed Computing Systems

Behzad D. Karkaria[1†], Neythen J. Treloar[1†], Chris P. Barnes[1,2] and Alex J. H. Fedorec[1]*

[1] Department of Cell and Developmental Biology, University College London, London, United Kingdom, [2] UCL Genetics Institute, University College London, London, United Kingdom

A distributed biological system can be defined as a system whose components are located in different subpopulations, which communicate and coordinate their actions through interpopulation messages and interactions. We see that distributed systems are pervasive in nature, performing computation across all scales, from microbial communities to a flock of birds. We often observe that information processing within communities exhibits a complexity far greater than any single organism. Synthetic biology is an area of research which aims to design and build synthetic biological machines from biological parts to perform a defined function, in a manner similar to the engineering disciplines. However, the field has reached a bottleneck in the complexity of the genetic networks that we can implement using monocultures, facing constraints from metabolic burden and genetic interference. This makes building distributed biological systems an attractive prospect for synthetic biology that would alleviate these constraints and allow us to expand the applications of our systems into areas including complex biosensing and diagnostic tools, bioprocess control and the monitoring of industrial processes. In this review we will discuss the fundamental limitations we face when engineering functionality with a monoculture, and the key areas where distributed systems can provide an advantage. We cite evidence from natural systems that support arguments in favor of distributed systems to overcome the limitations of monocultures. Following this we conduct a comprehensive overview of the synthetic communities that have been built to date, and the components that have been used. The potential computational capabilities of communities are discussed, along with some of the applications that these will be useful for. We discuss some of the challenges with building co-cultures, including the problem of competitive exclusion and maintenance of desired community composition. Finally, we assess computational frameworks currently available to aide in the design of microbial communities and identify areas where we lack the necessary tools.

Keywords: synthetic biology, microbial consortia, biological computing, multicellular systems, biotechnology

OPEN ACCESS

Edited by:
Boyan Yordanov,
Microsoft Research, United Kingdom

Reviewed by:
Jerome Bonnet,
Institut National de la Santé et de la Recherche Médicale (INSERM), France
Chris John Myers,
The University of Utah, United States
Karen Marie Polizzi,
Imperial College London, United Kingdom

***Correspondence:**
Alex J. H. Fedorec
alexander.fedorec.13@ucl.ac.uk

[†] These authors have contributed equally to this work

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *30 April 2020*
Accepted: *29 June 2020*
Published: *22 July 2020*

Citation:
Karkaria BD, Treloar NJ, Barnes CP and Fedorec AJH (2020) From Microbial Communities to Distributed Computing Systems. Front. Bioeng. Biotechnol. 8:834. doi: 10.3389/fbioe.2020.00834

WHAT DO WE MEAN BY COMPUTING WITH BIOLOGICAL SYSTEMS?

There may be as many definitions of computing as individuals willing to give one. In this review we will stick to one which is relatively general in order to allow us to draw analogy between electronic and biological computing implementations without becoming too restricted. As such, we define computing as the processing of information, to produce an output, in a manner that is encoded in a program. There are less ambiguous, yet still broad, definitions that have been used,

for example to determine when a physical system computes (Horsman et al., 2014). However, our layman's definition will suffice for this review. Although the dangers of analogizing have been well-documented (Thouless, 1953), even specifically in the field of synthetic biology (McLeod and Nerlich, 2018), we will proceed with caution.

The field of electronic computing has made great impact through the use, and evolution, of two core models: the Turing machine and the von Neumann architecture. The Turing machine defines a theoretical automaton which, according to a set of instructions, reads and writes symbols to an infinitely long tape (Turing, 1937). This model is used to demonstrate the limits of computability in what is known as the Church-Turing thesis. Although many other models of computing machines have been invented which may be faster or more efficient, none are capable of computing anything that a Turing machine cannot. The von Neumann architecture defines a "stored-program" model in which the instructions for performing computation are stored in the same way as the data on which the computation is being performed (von Neumann, 1993). This architecture includes a central processing unit (CPU) which communicates with a separate memory unit, an input and an output device. The CPU executes the instructions of the computer program and the memory stores data and instructions for the CPU. Although alternatives to both models have been explored, they remain the dominant paradigm for the design and programming of most electronic computers.

At least since Jacob and Monod (1961) famously described the *lac* operon in terms of a control system engaged in information processing, researchers have been exploring the ability of natural biological systems to compute. The engineering of *de novo* biological computation began with a demonstration of the use of DNA to solve an NP-complete Hamiltonian path problem (Adleman, 1994). Since then a large number of DNA molecular computing systems have been detailed: a molecular full-adder (Lederman et al., 2006), a small neural network (Qian et al., 2011), a non-deterministic universal Turing machine capable of solving non-deterministic polynomial (NP) time problems in polynomial time (Currin et al., 2017), all 16 two input logic gates (Siuti et al., 2013), a neural network capable of pattern recognition (Cherry and Qian, 2018), and even simple games (Macdonald et al., 2006; Pei et al., 2010). While DNA, and RNA, molecular computing is still actively being pursued, the other dominant paradigm since the advent of synthetic biology has been the use of gene regulatory networks (GRNs) within cells. Manzoni et al. (2016) provide an excellent introduction into the use of GRNs to produce Boolean logic operations; an approach which has provided some remarkable successes. However, an excellent recent perspective persuasively argues that synthetic biologists need to escape from the Boolean logic paradigm which has been so successful for electronic computation due to inherent differences between electronic circuits and biological systems (Grozinger et al., 2019).

The magnitude of the populations of cells that are used for most biotechnological applications is vast and, although our ability to engineer cells has greatly improved, the computational capabilities that we can implement in each cell is still relatively small. In computer science, these characteristics have been taken advantage of in large-scale distributed computer systems. However, it is only recently that synthetic biologists have started to move away from attempting to engineer monocultures of cells, all carrying out the same process. In this review we will introduce the current state-of-the-art in the engineering of microbial cells to compute. The limitations of the current approach of using monocultures are detailed and the concept of distributed computing is introduced as a potential solution. We review the tools available to produce distributed biological systems and suggest the current challenges to implementing such systems robustly.

ENGINEERING BACTERIA TO COMPUTE

The first synthetic biology papers engineered a toggle switch (Gardner et al., 2000), oscillator (Elowitz and Leibler, 2000) and autoregulation (Becskei and Serrano, 2000), which can be used as fundamental components in engineering a computer (Dalchau et al., 2018): memory, clock and noise filter. Since then, the tools necessary for engineering microbes for computation have been extensively developed over the last two decades of synthetic biology research. Though some of these tools have been developed explicitly for their use in cellular computing applications, many have been used to understand natural biological systems and to develop applications such as biotherapeutics (Ozdemir et al., 2018).

A biological switch is a bi-stable system that can be flipped between the two states. The first synthetic genetic toggle switch was built in *Escherichia coli* and was composed of two repressible promoters (Gardner et al., 2000). The product of each promoter repressed the other and chemical inducers could then be used to flip the switch between the two states. Similar switching behavior can also be achieved using transcriptional regulation (Kim et al., 2006). Multi-stable switches have been theorized (Leon et al., 2016) and implemented (Li et al., 2018) which would allow for greater than two state memory. The information storage capability of DNA has also been exploited to create cellular memory devices (Siuti et al., 2013), lasting for over 100 generations (Bonnet et al., 2012). Unlike a molecular toggle switch, DNA has the potential to encode complex sequences of data, allowing the encoding and decoding of a 5.27 megabit book (Church et al., 2012) and could extend cellular memory capabilities. However, DNA based memory is not currently switchable repeatably in the same manner as the transcriptional toggle switches.

A minimal sustained oscillator can be created with only a negative feedback loop and a time delay (Stricker et al., 2008; Hasegawa and Arita, 2013), but most biological oscillators are more complex. The repressilator (Elowitz and Leibler, 2000) was the first synthetic oscillator and consisted of a system of three cyclically inhibitory proteins. Oscillators are used in natural systems to coordinate the timing of events; the most ubiquitous example being the circadian clock, which keeps time with the day/night cycle and is found in even the most primitive organisms (Schippers and Nichols, 2014). A fast oscillator with tuneable periods as short as 13 min (Stricker et al., 2008) represents

a programmable timing device that could be used to time or synchronize cellular events with high precision, such as the release of a therapeutic dose (Danino et al., 2010). The robustness of the oscillations can be improved through the addition of autoregulation (Woods et al., 2016) or a "sponge" on one of the nodes (Potvin-Trottier et al., 2016).

As previously mentioned, transcriptional networks that produce Boolean logic gates have been extensively investigated. An AND gate that integrates the output of two promoters has been implemented in single cells (Anderson et al., 2007) and later more complex logical circuits were created by wiring together multiple layers of orthogonal AND gates (Moon et al., 2012). We now have libraries of orthogonal repressor-promoter NOT gates (Stanton et al., 2014), as well as the ability to produce *de novo* CRISPR-dCas9 gates (Zhang and Voigt, 2018), that can be wired together to make complex logical functions (Nielsen et al., 2016). These advances, along with tools to reduce DNA context effects (Davis et al., 2011; Lou et al., 2012; Mutalik et al., 2013) have enabled the construction of logic circuits with a great deal complexity in common lab strains of bacteria as well as strains relevant to microbiome engineering (Taketani et al., 2020). This level of circuit complexity is only achievable through the use of automated design tools, such as Cello (Nielsen et al., 2016), which match the empirical properties of genetic logic gates to ensure they will function together.

Biological processes in cells, based on the continuous concentration of metabolites and other molecules, are naturally analog. Analog computing is more efficient, in terms of the rate of ATP consumption and the number of protein molecule required, for doing addition with a genetic circuit at the ranges of precision that are metabolically feasible in single cells (Sarpeshkar, 2014). This is due to the mathematical dependence of precision on ATP consumption and number of protein molecules differing for analog and digital genetic circuits (Sarpeshkar, 2014). Additionally, it has been shown that building the equivalent circuit using analog logic can require orders of magnitude fewer genetic parts (Qian and Winfree, 2011; Daniel et al., 2013). Analog sensing, addition, and ratiometric and power law computations were implemented using only three transcription factors (Daniel et al., 2013). This was achieved by developing tuneable positive and negative logarithm circuits and connecting them through a common output to produce more complex circuits. Perceptrons, the building blocks of artificial neural networks, produce an output that is a function of the weighted sum of multiple inputs. They have been implemented using enzymes that transduce different inputs into a common output molecule, benzoate, and a synthetic actuator circuit that sensed benzoate (Pandi et al., 2019). This was used to build a cell based adder and cell free metabolic perceptrons in which enzyme concentrations acted as weights between nodes (Pandi et al., 2019).

LIMITATIONS OF MONOCULTURE ENGINEERING

Components of electrical circuits are, to a great degree, insulated from one another and the environment, with interactions enabled explicitly by wiring. Heterologously expressed genetic circuits lack insulation from one another within a cell. While efforts to create subcellular compartments in prokaryotes are ongoing (Giessen and Silver, 2016), these approaches will be difficult to generalize across different circuits and applications (Menon et al., 2017). Our construction of genetic circuits in a single strain is thus limited by fundamental and interconnected concerns: non-orthogonality, retroactivity, load, and burden (**Figure 1**).

The library of transcriptional regulators available for the construction of genetic circuits has vastly expanded in the last two decades, particularly in model organisms such as *E. coli*. However, as we cannot directly wire one component to another, we cannot reuse components without there being a confounding interaction. Even more frustratingly, several non-identical components share similarities that lead to non-orthogonality between those components, perturbing the intended functionality of the engineered circuit (**Figure 1A**). As the scale of genetic circuits grows, the number of opportunities for non-orthogonal interactions grows exponentially, making it difficult to scale complexity. Efforts to circumvent this include

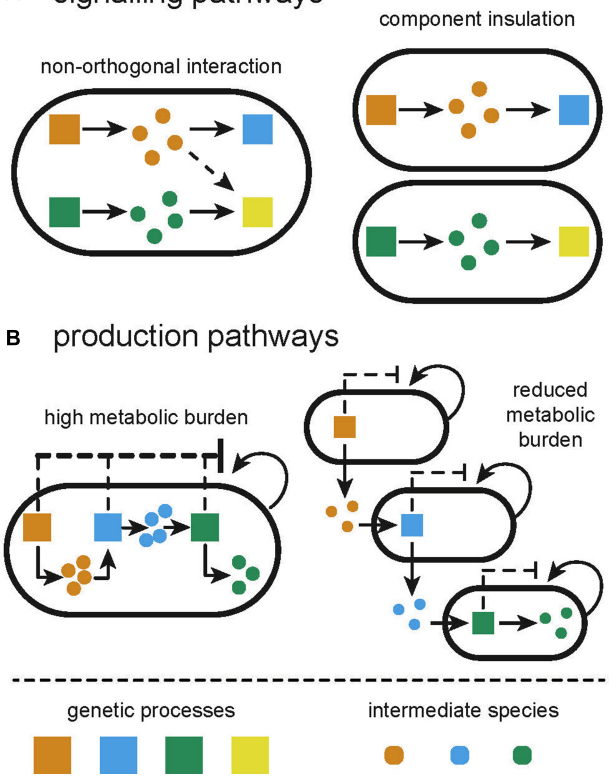


FIGURE 1 | Illustration of the limitations of monocultures and how they are overcome by constructing distributed systems from multiple cell types. **(A)** Signaling pathways in monocultures can often suffer from unintended and unexpected crosstalk between processes. Compartmentalizing independent functions in separate subpopulations will prevent crosstalk. **(B)** Applications in biosynthesis suffer from high metabolic burden due to the expression of multiple heterologous processes. Division of labor between multiple subpopulations can alleviate the metabolic burden.

"part-mining" to build libraries of orthogonal parts (Stanton et al., 2014) and computational design tools to incorporate known non-orthogonal interactions as part of the design process (Kylilis et al., 2018; Nguyen et al., 2019). Even the vast space of *de novo* parts enabled by CRISPR-dCas9 is limited by the number of sgRNAs that can be co-expressed before severely depleting the pool of dCas9 (Zhang and Voigt, 2018). The largest genetic circuit within a single cell, at the time of writing, consists of 55 genetic parts (Nielsen et al., 2016). In addition to such unwanted molecular interactions, sequence similarities between components can lead to mutation of genetic circuits due to homologous recombination. Libraries of parts, for example terminators, have been specifically designed that can be used together in order to circumvent this (Chen et al., 2013). Retroactivity describes a type of non-orthogonal interaction, whereby an upstream process is perturbed by a downstream species (Jayanthi and Del Vecchio, 2011). Retroactivity is common in signaling pathways with reactions that operate on different time scales, causing the accumulation of intermediate species that may interact with the upstream process (Jayanthi and Del Vecchio, 2011; Kim and Sauro, 2011; Pantoja-Hernández and Martínez-García, 2015).

The expression of genes draws from a pool of shared resources within the host. As such, the co-expression of two genes within a circuit can become coupled due to limited resource availability (Gyorgy et al., 2015). This has been compared to the load that is experienced in electrical circuits when components are placed in parallel (Carbonell-Ballestero et al., 2016). One is therefore limited in the number of components that can utilize the output from another component as their input. Since recombinant and host processes use the same resource pool, recombinant gene expression will also draw resources away from host processes causing a metabolic burden, exhibited as reduced growth rate (Glick, 1995; **Figure 1B**). The slower growth can encourage selection for cells which manage to lose or mutate their genetic circuit (Rugbjerg et al., 2018); strains not expressing the burdensome circuit have a competitive advantage and can outgrow the burdened population (Summers, 1991). Furthermore, metabolic burden can induce stress responses in the host, increasing mutation rates (Matic, 2013; Couto et al., 2018). Whole cell models, combining the impact of load and metabolic burden, show how changing resource availability in a host strain can produce different circuit behavior (Gorochowski et al., 2016; Boeing et al., 2018). Efforts to reduce load and metabolic burden include optimizing circuits for low copy plasmids or chromosomal integration (Lee et al., 2016), and using orthogonal ribosomes to allocate recombinant gene expression to different resource pools (Darlington et al., 2018; Boo et al., 2019). Expression of burdensome circuits can be regulated dynamically in response to population density (Gupta et al., 2017) or using promoters that are directly sensitive to burden (Ceroni et al., 2018). Mishra et al. (2014) developed a load driver for *Saccharomyces cerevisiae,* demonstrating consistent levels of expression regardless of load induced.

All of these limitations can be overcome by dividing the functionality of a circuit between subpopulations of cells, in what we will call a distributed biological system, rather than attempting to engineer a monoculture to achieve everything (**Figure 1**).

FROM SEQUENTIAL TO DISTRIBUTED COMPUTING

Before discussing distributed biological systems, it is sensible to provide a short introduction to distributed computing and how it relates to other approaches to computing. In simple terms, a computer program is a set of instructions for reading, operating on, and writing data. A sequential computer processes the instructions from programs, one after the other, until the program halts. Concurrency is the execution of many programs during the same period of time, but not necessarily at the same instant. This can be achieved on a single processing unit by interleaving the instructions from multiple programs. This produces the appearance of programs running in parallel and allows the computer to respond to input from devices such as a keyboard. Although parallel and distributed computing are inherently forms of concurrent computing (many programs being run during the same time period), single processor concurrency is not true parallelism as there is still only one instruction being processed at a time.

Parallelism is the execution of instructions on separate processing units, simultaneously. There are many forms of parallelism and many ways of categorizing them but the most common is Flynn's taxonomy (Flynn, 1972). This taxonomy, shown in **Figure 2**, uses the number of streams of instructions and data to create four categories: SISD, SIMD, MISD, and MIMD. Single-instruction single-data (SISD) corresponds to the sequential computer; one instruction is being carried out using one location in memory. In a single-instruction multiple-data (SIMD) architecture, the same operation is synchronously performed by different processor units on data from different locations in a shared memory. Graphics processors use this architecture to, for example, parallelize operations on pixels within an image. Multiple-instruction single-data (MISD) is an uncommon form of parallelism but has been employed in safety critical systems as a redundancy methodology i.e., agreement must be reached by multiple systems, exposed to the same input, for an operation to be accepted.

Multiple-instruction multiple-data (MIMD) is a form of parallelism that is now ubiquitous in modern personal computers. Here we choose to further subdivide MIMD systems to discriminate between single-program multiple-data (SPMD) and multiple-program multiple-data (MPMD). The former is a commonly used parallel programming paradigm used to speed up the runtime of a program by allowing instructions, that do not depend on results from one another, to run simultaneously on separate processing units. The limits of the speedup that can be achieved are given by Amdahl's (fixed problem size) (Amdahl, 1967) and Gustafson's (problem size scales with number of processors) (Gustafson, 1988) laws. It is often hard to achieve significant speedup as the requirement for independence excludes many steps within common algorithms.

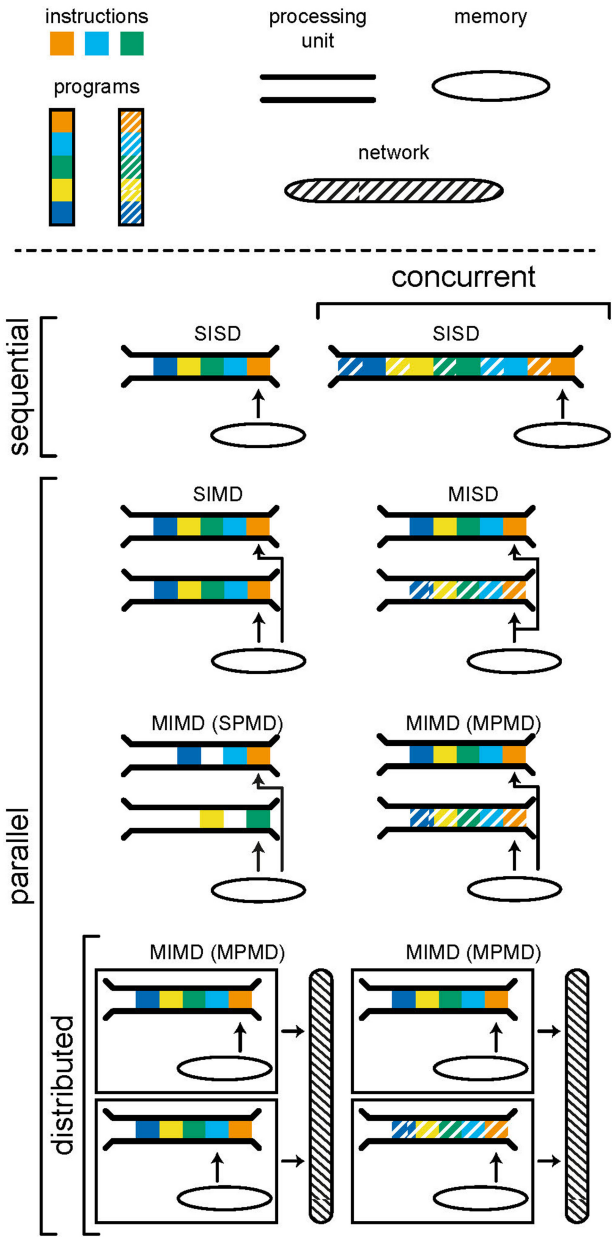


FIGURE 2 | Models of computing categorized using Flynn's taxonomy. A program consists of a number of instructions which are executed by a processing unit. A computer, or computational unit, consists of one (sequential) or more (parallel) processing units which can access data and instructions from memory. Concurrency is achieved by interweaving instructions from different programs in order to produce the appearance of parallelism. Distributed models include separate computational units which communicate by passing messages through a network. The individual computational units in a distributed system may be sequential or parallel.

MPMD is the category within which distributed systems lie. Here, different programs are run on separate processing units, accessing their own data. Distributed systems are a special case in which each processor does not have access to a shared memory and instead programs must communicate with one another through message passing. This tends to have a far higher latency (the time it takes for information to be transferred) but also higher bandwidth (the amount of information that can be transferred at once) than accessing local memory and, as such, message passing should be limited to infrequent but large transfers of data. When a distributed system is used for a common goal, there is often a control computer which assigns tasks to computers within the network and receives and synthesizes resulting data, as is common in high performance computing. Alternatively, computers within the network may have their own compulsion and the network merely allows for the sharing of resources. It is important to note that each individual computer within a distributed system can be operating in any of the categories of Flynn's taxonomy; each computer may run the program(s) it is tasked to run sequentially or in parallel.

Models developed for describing concurrency have become the dominant models of distributed systems. Petri nets use graphs of "transitions" and "places," analogous to instructions and memory, connected by "arcs," to describe dynamic systems of discrete events (Petri, 1966). If the state of the places connected to a transition meet the defined requirements, the transition fires and the states of the connected places will change. Petri nets have been extensively used to model discrete chemical and biological processes (Wilkinson, 2018). The actor model consists of "actors" with their own private state (Hewitt et al., 1973). They are able to communicate only through addressed message passing and can act, concurrently, based on the messages received by sending messages, creating new actors and queuing behaviors. Finally, process calculi are a collection of algebras for modeling concurrent systems using "channels" to communicate between processes. Several variants exist that enable reasoning about, for example, systems with mobility (ambient calculus; Cardelli and Gordon, 1998), systems with changing network configuration (pi-calculus; Milner et al., 1992) and probabilistic systems (PEPA; Hillston, 1996).

Challenges specific to distributed systems relate to communication and coordination. Two foundational concepts that should be discussed here, as they have strong parallels with biological systems, are common knowledge and faulty agents. The former is detailed in an important paper in the field of distributed systems (Halpern and Moses, 1990). Individual computers within a distributed system act solely on their own local information which is learnt from their own processes and receiving messages from other computers. However, some applications require the agreement or simultaneous action of multiple computers which can only be achieved through "common knowledge," globally known information. Halpern and Moses demonstrate that common knowledge is unattainable but detail weaker forms, such as time limited common knowledge, which allow some actions to be performed (Halpern and Moses, 1990). The problem of faulty agents is related as it concerns reaching agreements via communication of information between computers. In this scenario some of the computers in the network are faulty or malicious and, as such, the messages that they pass are unreliable. It is provably possible to reach agreement if less than one third of the network is faulty, as long as each computer knows the sender of each message it receives (Lamport

et al., 1982). However, this solution requires synchronization which is not possible without common knowledge and in an asynchronous system consensus is theoretically impossible with even one faulty computer (Fischer et al., 1985), though pragmatic solutions exist (Chandra and Toueg, 1996).

There have been many attempts to draw analogies between electronic computers and biological systems as computers. The main features of a distributed system are concurrency of components, lack of a global clock, and independent failure of components (Attiya and Welch, 2004), all of which apply naturally to biological communities. From the above description of computational systems, we believe it is reasonable to consider an individual cell as a computational unit. More detailed analogies could be made, for example, between fetching an instruction and the transcription process, or performing an operation and enzymatic reactions. However, these analogies often differ depending on the abstractions that one is working on within the cell. Cells are capable of parallel processing; they are able to execute multiple tasks simultaneously. Synthetic biology to date has predominantly been undertaken using monocultures in well mixed liquids with the assumption that all cells are performing the same operation in the same environment. However, we know that heterogeneity between cells and across the environment make these systems much more analogous to distributed systems in which cells are asynchronously running the same program, exposed to different environments, alongside numerous other programs running in parallel. Further, the necessity to distribute genetic circuits across heterogenous communities of engineered cells in order to tackle the limitations of monoculture computing compels us to think of synthetic biology through the prism of distributed systems.

DISTRIBUTED SYSTEMS IN NATURE

Several naturally occurring biological phenomena involving cellular communities and multicellular organisms can be considered naturally occurring distributed systems. Individual cells are able to process information intracellularly and share and receive information extracellularly through, for example, the secretion of molecules.

Bet Hedging

A solution to the problem of changing environments often encountered by natural microbial communities is bet hedging. This is a strategy in which a certain percentage of a population adopt a sub-optimal state for the current environment in anticipation that the environment can change (**Figure 3A**). In this way the long term fitness of the community is increased by reducing the current fitness of a subset of the community. This can be entirely stochastic (Wolf et al., 2005) or biased by sensors that pick up environmental signals (Kussell, 2005). A game theoretic analysis found that switching between different losing strategies produces a winning strategy when environmental transitions cannot be sensed (a Parrondo paradox) (Wolf et al., 2005). Further, the optimal switching rates are a function of environmental properties and that diversification is favorable upon entering new environments with noisy information. It was separately shown that stochastic switching can be favored over sensing when the environment changes infrequently and that the optimal switching rates are again dependent on the properties of the environment (Kussell, 2005). Bet hedging has been demonstrated to be even more favorable when colonizing new environments, supporting the view that expanding into novel environments supports diversification (Villa Martín et al., 2019). This research shows that bacterial colonies leverage the capacity for phenotypic heterogeneity to produce a community that is optimized, according to the principles of game theory, for survival or expansion in uncertain environments. This has analogs in various forms of search and optimization algorithm, in which multiple, simple heuristics or algorithms can be explored in parallel to provide a solution (Huberman et al., 1997; Deng et al., 2012).

Development of Multicellular Organisms

The process of development, by which a single cell becomes a morphologically complex organism composed of well-organized, heterogeneous tissue has been shown to be largely orchestrated by signaling using diffusible molecules called morphogens (**Figure 3B**). A theoretical model of morphogenesis was first presented by Alan Turing (Turing, 1952). This model is based on systems of multiple morphogens that react with each other and diffuse through tissue. Simulation results showed that the reaction diffusion model could correctly predict the spacing of angelfish stripe patterns (Kondo and Asai, 1995). Later work concluded that there are universal mechanisms of specifying cell spatial information, based on fields and polarities (Wolpert, 1969). A field is a group of cells that have their position specified with respect to the same set of points and polarity is the direction in which spatial information is specified. Francis Crick proposed that the fields might be produced by sources and sinks of diffusible molecules (Deuchar, 1970). This model has since been shown to be accurate for the Fgf8 morphogen in zebrafish embryos (Yu et al., 2009). A further proposed explanation of a field is that it constitutes a group of cells that are oscillating synchronously and are tightly coupled (Newman and Bhat, 2009). This could be the mechanism behind clusters of cells in the insect wing disc that progress through the cell cycle together and could also help explain how some developmental fields work over longer distances than would be possible by diffusion (Giribet, 2009). The epigenetic landscape (Waddington, 1957) for a simple regulatory network consisting of two genes has been quantified and found to behave as a potential function, with basins of attraction at the differentiated states (Wang et al., 2011). The idea of a fitness landscape has also been applied in areas such as cell signaling (Sekine et al., 2011), cell death (Zinovyev et al., 2013), and pattern formation in *Drosophila* (Lepzelter and Wang, 2008). Recent attempts to quantify spatial information during development include a demonstration that the expression level of just four gap genes can be used to specify a cell's position with 1% uncertainty in the *Drosophila* embryo (Dubuis et al., 2013). The developmental process has been compared to mathematics (Apter and Wolpert, 1965) in which a set of basal rules is used to derive a complex structure. In this way development

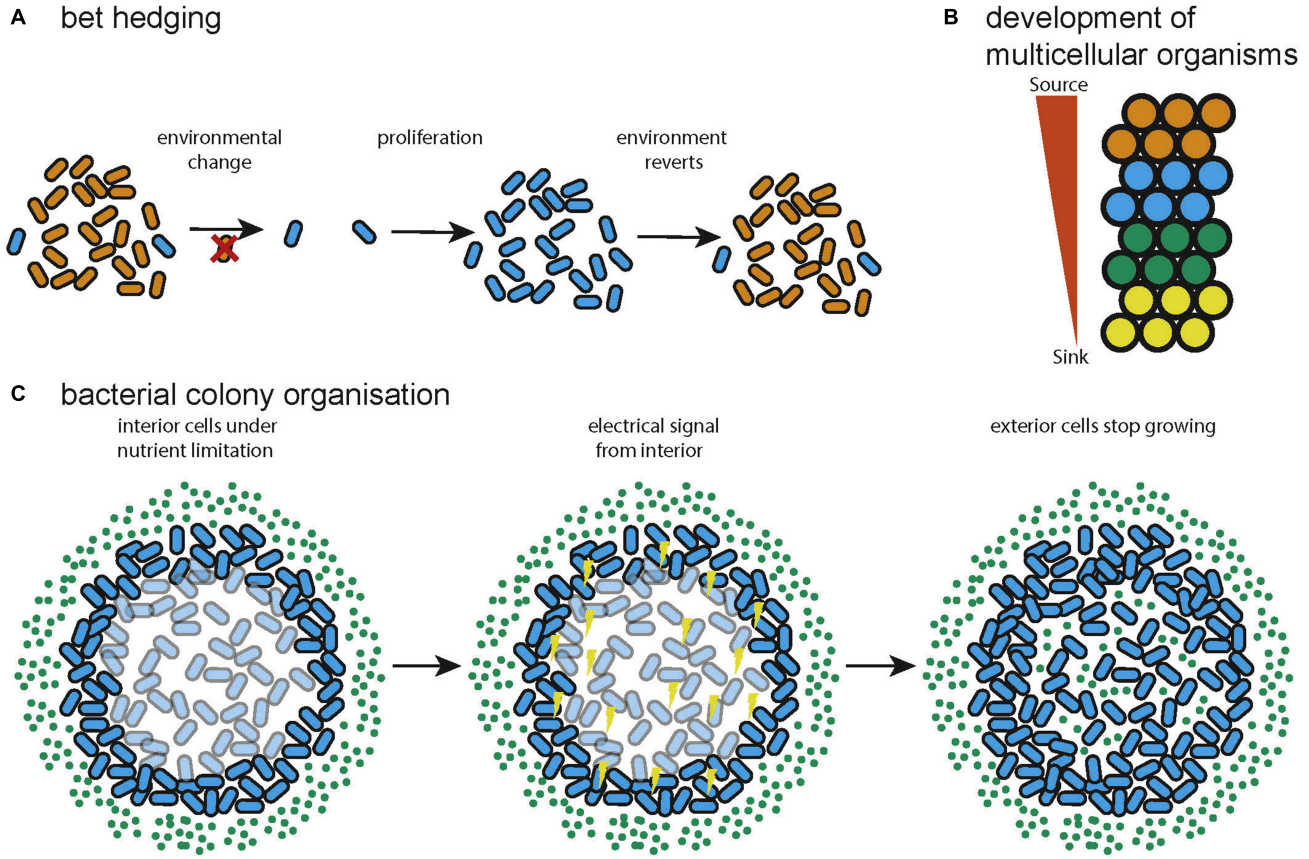


FIGURE 3 | Examples of distributed systems in nature. **(A)** Bet hedging in natural bacterial communities, variation in the phenotype of the cells confers resistance to environmental change to the community (adapted from Wolf et al., 2005). **(B)** During multicellular development spatial information can be specified by gradients of diffusible molecules (adapted from Yu et al., 2009). **(C)** Bacteria can use electrical signaling to organize metabolism so that both interior and exterior cells can grow (adapted from Martinez-Corral et al., 2019).

can be seen as the efficient compression of the spatial and cell type information required to generate a complex organism from a single cell.

Bacterial Colony Organization

Microbiomes are diverse communities of organisms that exhibit a group metabolism (Gill et al., 2006), resistance to pathogenic invasion (Stein et al., 2013; Buffie et al., 2015) and temporal stability of community function through dynamic adaptation of community members (Coyte et al., 2015). Bacteria have developed multiple methods of exchanging information including diffusible quorum sensing molecules (Nealson and Hastings, 1979), exchanging DNA via conjugation (Tatum and Lederberg, 1947), and even electrical communication (Prindle et al., 2015; Martinez-Corral et al., 2019). This allows the assembly and maintenance of spatial structure in a colony (Jacob et al., 2004; Ben-Jacob and Levine, 2006) and the spatial coordination of metabolism so that nutrients are shared across a community (**Figure 3C**; Prindle et al., 2015; Martinez-Corral et al., 2019). Additionally, using the ability of individual bacteria to sense environmental inputs and respond accordingly, bacterial colonies can adapt their spatial configuration to a changing environment, reacting to food availability or optimizing foraging. The bacteria *Paenibacillus vortex* forms highly modular colonies (Ben-Jacob et al., 1998; Ben-Jacob, 2003; Jacob et al., 2004; Ben-Jacob and Levine, 2006) in which circular modules of bacteria move around a common center. *P. vortex* can also form snake like swarms which can sense and collectively respond to input signals, for example swarming to collect multiple sources of extracellular material (Ingham and Jacob, 2008). This has also inspired an optimization algorithm called Bacterial Foraging Optimization (BFO) (Passino, 2002), a distributed optimization algorithm that mimics the foraging behavior of a colony of bacteria. BFO can be described as a variant of particle swarm optimization (Kennedy and Eberhart, 1995) that incorporates selection by using aspects of genetic algorithms (Holland, 1992). BFO has been found to be effective on real world problems such as signal estimation (Mishra, 2005) and controller optimization (Mishra and Bhende, 2007), in both cases it was found to outperform a conventional genetic algorithm in terms of convergence time or solution accuracy. Microbes can also interact through the exchange of metabolites. In this manner a bacterial community can exhibit an optimized group metabolism enabling the community to survive with minimal resources and persist in environments

inhospitable to the individual microbes (Schink, 2002; Morris et al., 2013; Lau et al., 2016). Mathematical modeling suggests that syntrophy can often emerge spontaneously between pairs of microbial metabolisms (Libby et al., 2019) and much work shows that syntrophy leads to the loss of functional independence as genes are lost to minimize the energy usage of the community (Morris et al., 2012; Hillesland et al., 2014; D'Souza and Kost, 2016; McNally and Borenstein, 2018). Syntrophy commonly occurs within bacterial communities, for example during methanogenesis (Zhu et al., 2020), and the metabolic reactions within the human gut (Ruaud et al., 2020).

DIFFERENCES BETWEEN BIOLOGICAL AND SILICON SYSTEMS

There are a few key differences between natural and man-made distributed systems that deserve highlighting. The first is the main method of communication; in a computer, components are connected by electrical wires and individual computers can communicate through wired networks which allow specific message passing. Even in wireless networks in which messages are broadcast, enough information can be attached to a message so that it is only readable by the target computer. This means that nodes in a distributed system can send messages specifically and communication networks can be set up to include arbitrary groupings of nodes. Although systems exist for the passage of messages specifically between cells (Goñi-Moreno et al., 2013), due to its ubiquity in bacteria and the relative low level of complexity quorum sensing is the dominant method of communication engineered into synthetic bacterial consortia. When communicating via quorum sensing, bacteria secrete the message in the form of a diffusible molecule. A secreted molecule sent by a cell will reach any cell within its vicinity and the requirement to read the "message" is only expression of the associated, or closely related, sensor, meaning that this is a form of broadcast communication. The rigidity of the connections between a set of computers in, for example, a local area network mean that the network can be classified as a solid network, meaning that the connectivity of the network does not change with time. This is in contrast to a community of cells, where agents can move relative to each other and agents communicate with other agents in their local area. This means that connectivity will change with time, and the community can be classified as a liquid network (Solé et al., 2019). This distinction has important implications for message passing and communication within a microbial community. For example, the "wiring problem" occurs when more than one communication channel is required within a bacterial community. Later, we discuss the current communication tools available for synthetic biologists and detail their limitations. Microbial communities are also composed of reproducing biological organisms, meaning that they are subject to selective competition and potential disruptions via mutations. This also allows natural communities to adapt to changing environments but is a fundamental challenge in synthetic biology, as will be discussed below. However, the merits of liquid networks have been investigated (Langton, 1986; Miramontes et al., 1993; Solé and Miramontes, 1995; Solé and Delgado, 1996; Vining et al., 2019) and it has been shown that liquid networks are capable of reaching a global consensus (Vining et al., 2019) and universal computation (Solé and Delgado, 1996).

A second key difference is that the great majority of electronic computers use digital memory and logic. Analog systems are often emulated on digital computers, which introduces inefficiencies in terms of power consumption and simulation time (Guo et al., 2016). Microbes are not limited to digital computation and often use analog computations to their advantage, for example the continuous responses of environmental sensors (Mannan et al., 2017) or the addition of the concentration of quorum molecules from multiple sources (Long et al., 2009). This in turn relates to how the different systems treat noise. In a digital computer variability in the output from a component is considered undesirable and, as such, error checking and correcting mechanisms are built into every level of a computer (Johnson, 1984; González et al., 1997). As detailed in the previous section natural communities, however, often harness noise in both gene expression and the genetic makeup of the community (Kussell, 2005; Wolf et al., 2005; Villa Martín et al., 2019).

DISTRIBUTED SYSTEMS IN SYNTHETIC BIOLOGY

The challenges, described previously, of non-orthogonality, load, and burden in synthetic biological systems have been confronted by the expansion of genetic parts libraries (Chen et al., 2013; Mutalik et al., 2013; Stanton et al., 2014). However, more and better parts will only push our problems further into the future. The ever-increasing capabilities of computers has enabled, and perhaps been driven by, the development of ever more demanding software. The same will happen with synthetic biology; the complexity of the systems we design will always push the limits of the parts that are available to us.

Using the principles developed over several decades of work on distributed computing and insights from research into natural biological distributed systems offers an alternative, and complementary, approach to expanding parts libraries. Distributing a system between subpopulations of cells means that we can reduce the number of parallel tasks that we are asking host cells to perform, reducing load and burden, and enabling the reuse of parts in different subpopulations without orthogonality issues.

Available Tools for Building Distributed Synthetic Biological Systems

Liquid and Solid State Environments

Distributed synthetic biological systems can be assembled as liquid or solid cultures. The choice of which will be dictated by the intended application, with each choice possessing important advantages and disadvantages.

In a well-mixed liquid culture, microbial cells exist as independent entities that are free swimming. All subpopulations

share approximately the same environment, offering a constant intermediary for the exchange of resources and information.

Bioreactors and microfluidic devices allow different scales of control over liquid culture environments, the choice of which plays an important role in the behavior of the populations. Over the past several years a number of low-cost bioreactors have been developed (Takahashi et al., 2015; Hoffmann et al., 2017; Steel et al., 2019). Turbidostats are a class of continuous bioreactor that maintain the culture at a constant optical density (OD) by varying the dilution rate. A turbidostat can maintain the culture in the desired growth phase indefinitely (Takahashi et al., 2015; Hoffmann et al., 2017). This is of particular interest for implementing distributed systems since gene expression profiles often differ between phases of growth (Klumpp et al., 2009). Some of these bioreactor devices can be configured to measure the output of several fluorescent proteins simultaneously and control multiple inputs dynamically (Steel et al., 2019). Dilution rate has been cited several times as a critical controllable parameter; the rate of removal of molecules from the environment can produce very different population dynamics (Balagaddé et al., 2008; Weiße et al., 2015; Yurtsev et al., 2016; Fedorec et al., 2019). As such, possessing the correct tools is important for building distributed systems in liquid culture.

Microfluidic devices have been developed that enable batch, chemostat and turbidostat cultures (Lee et al., 2011; Ullman et al., 2013). These have been used for a range of applications, such as high-throughput gene expression analysis (Lee et al., 2011; Ullman et al., 2013), elucidating the relationship between population density and antibiotic effectiveness (Karslake et al., 2016), the evolution of antimicrobial resistance (Toprak et al., 2012), and screening for fitness under different environmental conditions (Wong et al., 2018). Such devices are suited to assessing community cultures and have been applied in the microbial ecology field to understand multi-faceted interactions (Kehe et al., 2019). Microfluidic traps can be used to monitor cells in a fixed position and enable the establishment of local microenvironments while still having a regular turnover of cells and nutrients (Bennett and Hasty, 2009). Microbial traps capture some properties of solid state cultures. In some cases trap-like structures are essential for generating a critical cell density and ensuring short diffusion distances (Chen et al., 2015). Microfluidic traps can also be used to investigate the spatiotemporal dynamics of consortia and how strain interaction and signaling efficacy is affected by trap size (Alnahhas et al., 2019). A further microfluidic device has been used to investigate quorum sensing over different lengths. The effect of distance on information transmission, the robustness of a distributed genetic oscillator and mutualistic interaction between two strains was investigated (Gupta et al., 2020).

Liquid cultures provide the closest analog to a shared memory model of computing in which all processing units (the cells) have direct access to the same data (the environmental state). However, the common assumption that liquid cultures are homogenous does not stand up to scrutiny (van 't Riet and van der Lans, 2011). Accounting for the latency in a communication network and spatial distribution of species are important characteristics to include. For example, changing flow rates in a microfluidic device can turn synchronized population oscillations into spatiotemporal traveling waves because dilution occurs non-uniformly in space (Danino et al., 2010). This suggests that, rather than using a model of shared memory that is implicit in most models of bacterial liquid cultures may be insufficient under some circumstances.

In solid state cultures, microbes will often assemble into a biofilm. Biofilms are a mass of microorganisms which adhere to a self-produced extracellular matrix (ECM) (Flemming et al., 2016). The ECM density allows for the establishment of local concentration gradients (Flemming et al., 2016) which in turn allows the formation of local niches (Poltak and Cooper, 2011). Biofilm formation itself is a form of computation through communication, invoking a pattern of gene expression to drive a developmental process (Davies et al., 1998; Sauer et al., 2002; Liu et al., 2017; Abisado et al., 2018), similar to how morphogen gradients that define cell fate are a well-characterized form of computation in mammalian cells (Christian, 2012). Members of a biofilm often experience direct cell-to-cell contact with one another, required for horizontal gene transfer through conjugation (Flemming et al., 2016; Madsen et al., 2018). Microbial ecology studies show that the community metabolic output of a biofilm is positively associated with ecological diversity (Boles et al., 2004; Poltak and Cooper, 2011). Since biofilms are often naturally diverse systems, they possess attractive characteristics for building spatially distributed systems. Studies have demonstrated control over biofilm formation in a variety of ways. Optogenetically induced gene expression systems can be used to produce defined patterns of biofilm formation (Huang et al., 2018). Quorum sensing and antimicrobial peptides can be used to generate tuneable bandpass patterns (Kong et al., 2017) or control the dispersal and colonization of biofilms in multiple subpopulations (Hong et al., 2012).

Explicit distribution of subpopulations in 3D structures may prove to be an important tool for building distributed systems in solid states. 3D-printing offers a manufacturing platform for rapid prototyping from CAD designs to three-dimensional structures (Savini and Savini, 2015). The more recent falling cost of desktop 3D-printers have made this technology an attractive option for bioengineering, replacing extruded plastics with bioinks. These are made from biocompatible materials such as hydrogels, gelatin or alginate and are designed to cross-link immediately after or during bioprinting (Gungor-Ozkerim et al., 2018). They are seeded with living cells which can be printed directly into the desired 3D conformation (Connell et al., 2013; Schaffner et al., 2017; Huang et al., 2018; Qian et al., 2019). Structures can be designed to increase mass transfer, leading to improvements in product yield (Qian et al., 2019) and distinct populations can be layered on top of one another (Lehner et al., 2017; Schmieden et al., 2018). Bacteria can be used to functionalize these materials. For example, hydrogels mixed with *Pseudomonas putida* conferred the degradation of phenol (bioremediation functionality); while improved mechanical robustness can be harnessed by mixing hydrogels with cellulose producer *Acetobacter xylinum*,

suitable for biocompatible medical applications (Schaffner et al., 2017). Connell et al. (2013) demonstrated generation of "core-shell" geometries, where an internal core population can be protected from external environmental conditions by being encompassed by a distinct shell population. Such cross-species protection interactions can be observed in the oral microbiota (Marsh, 2005).

Modeling Approaches

The field of microbial ecology frequently uses genome scale metabolic models to infer the interactions between community members and can serve as an important guide for building large scale synthetic systems (Biggs et al., 2015). It has become common practice to build metabolic models of individual community members that can then be combined to make quantitative predictions about the metabolic dependencies and interactions. This approach has been applied to the prediction of metabolic interactions between species in the gut microbiome (Shoaie et al., 2015). Similarly, genome-scale metabolic models have been used to aide in the design of large scale communities by predicting metabolites that can be released by the producer without detriment to fitness, and conditions that encourage the establishment of stable communities (Pacheco et al., 2019). Thommes et al. (2019) used genome scale metabolic models of *E. coli* to compute feasible division of labor strategies that could arise from an initial monoculture through loss of function in genes, giving insight into possible avenues for engineering community formation. Angulo et al. (2019) demonstrated a mathematical method for identifying "driver species" in an ecological network. External control of the driver species allows the user to manipulate the state of the entire network. Approaches such as these could be a steppingstone between ecological communities and building entirely synthetic networks.

Agent-based models are a class of computational model that simulate a system of autonomous agents and their interactions. Agent-based models are effective for modeling systems with discrete elements and are useful for representing heterogenous environments and spatial distribution of species (Gorochowski, 2016). This approach has been used extensively to model formation and interactions in biofilms (Kreft et al., 1998; Lardon et al., 2011). Gro is a high-level framework for defining and simulating bacterial colony growth (Jang et al., 2012). Gro has more recently been extended to include nutrient uptake and cell-cell signaling, enabling the simulation of spatial patterning in 2D (Gutiérrez et al., 2017). Agent-based modeling frameworks DiSCUS and BactoSIM have been used to simulate conjugation processes in biofilms and how this effects the population as a whole (García and Rodríguez-Patón, 2015; Goñi-Moreno and Amos, 2015); an important form of information propagation bacterial systems. BSim 2.0 is a flexible modeling framework that can be used to simulate microbial community systems in microfluidic devices (Matyjaszkiewicz et al., 2017). The software can simulate signal expression, diffusion and response, and has been used to identify optimal microfluidic chamber design for a particular community behavior (Matyjaszkiewicz et al., 2017).

Implemented Synthetic Biological Distributed Systems

Modular Logical Circuits

One of the key engineering principles that synthetic biology strives to adhere to is modularity so that biological components can be recombined and interchanged to build new systems rather than needing to design full systems from scratch. A successful example within the context of synthetic biological distributed systems is the decomposition of a complex logical function into multiple subunits, each engineered within a different population of cells that communicate with each other (**Figure 4A**). This mirrors a common approach in electronics where two universal logic gates, for example NOR and NAND, are wired together to produce any logical function. In this manner all 16 two input logic gates have been created using bacterial colonies on agar plates, containing genetically engineered NOR gates, and communicating via diffusible molecules (Tamsir et al., 2011). A similar approach consisted of a community of yeast cells that carried out the functions AND, NIMPLIES, NOT, and IDENTITY (Regot et al., 2011). These are chemically wired together using diffusible communication molecules to produce complex functions. The output was also distributed across multiple cell types, helping to reduce wiring requirements and enabling the construction of all the two input logic gates, multiplexer and 1-bit adder with carry (Regot et al., 2011). Mathematical work into the optimal design of computational communities implementing distributed genetic logic gates given realistic constraints on the number of logic gates possible per cell and the number of orthogonal quorum molecules has been done (Al-Radhawi et al., 2020). It was found that under the assumption that any cell is limited to a maximum of seven logic gates the use of a community composed of two cell types increased the number of logic gates by 7.58-fold over the capabilities of a monoculture. Another automated design framework for the construction of user specified logical functions using DNA recombinase NOT and IDENTITY gates distributed over multiple cell types enables the design of a consortium of bacteria to perform the desired digital function (Guiziou et al., 2018). This framework was then used to build consortia capable of four input digital logic (Guiziou et al., 2019). The standard mathematical proof that any Boolean function can be decomposed into a double summation of IDENTITY and NOT logics was used to build multicellular circuits encoding the IDENTITY and NOT logic into cells and then performing sums by mixing cell cultures together (Macia et al., 2016). In this manner arbitrary logic functions can be built. A different approach using antibiotic sensitivity has been used to construct a three-bit full adder and full subtractor using *E. coli* cells with a calculator like display (Millacura et al., 2019). Combinatorial resistance was used to distinguish between different combinations of three antibiotics, then a visual output was distributed across cell types arranged in a spatial display.

Memory

A key component for computation is memory. Quorum sensing has been combined with a genetic toggle switch, resulting in a population level toggle switch (Kobayashi et al., 2004).

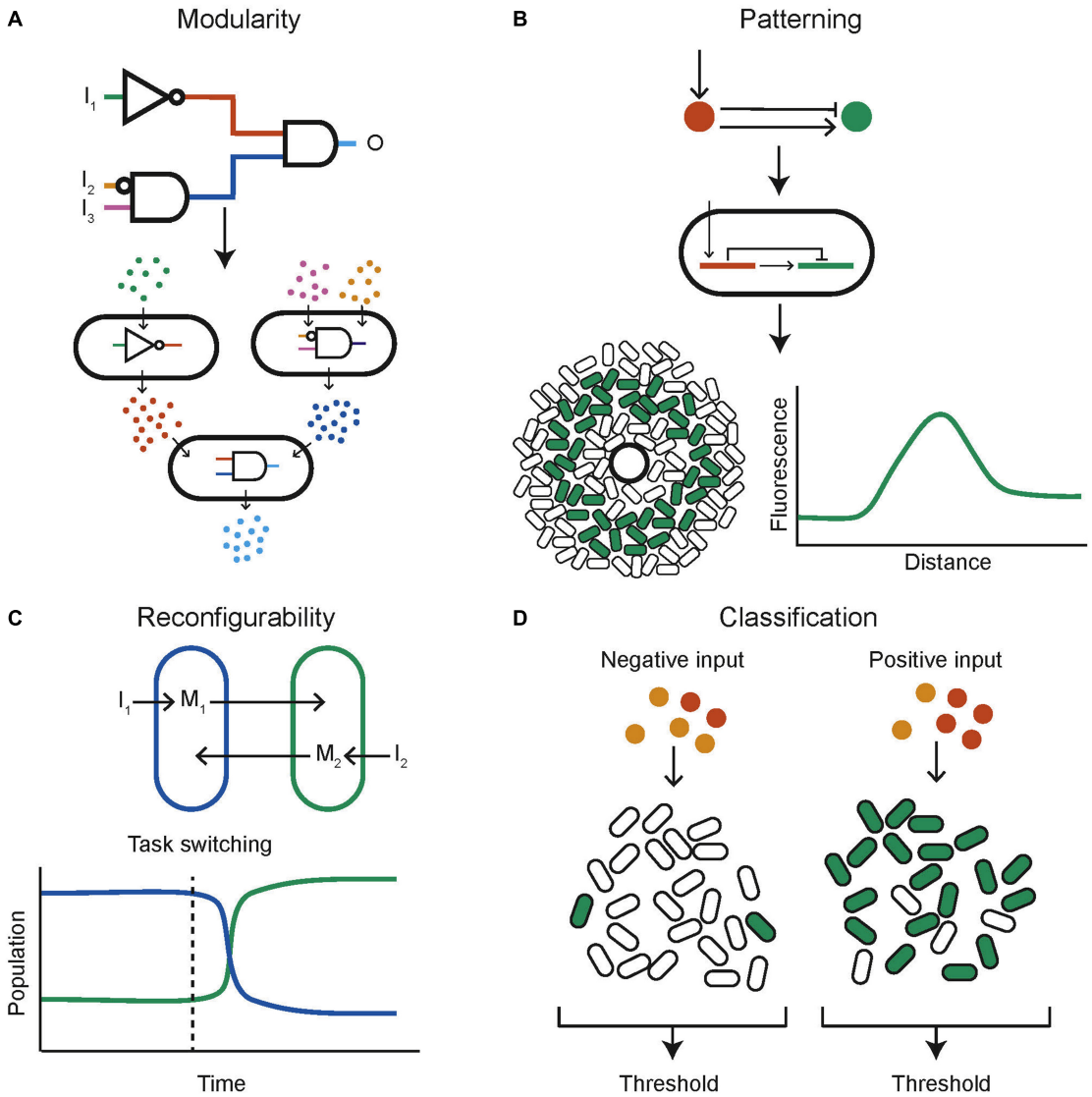


FIGURE 4 | Example capabilities of computational communities. **(A)** A complex circuit can be split into modules, distributed across different populations of cells (adapted from Regot et al., 2011). **(B)** Computational methods can be used to find networks capable of stripe formation, these can be programmed into cells using genetic circuits, the expressed phenotype of each cell depends on its position relative to a source of signaling molecule (adapted from Schaerli et al., 2014). **(C)** Reconfigurability could be a key capability of biological computing. Here the composition of a bacterial community can be controlled through inducers I1 and I2. This capability could be used to task switch in a computational bacterial community (adapted from Kerner et al., 2012). **(D)** Bacterial communities are naturally applicable to complex functions such as ensemble classification (adapted from Kanakov et al., 2015).

A synthetic community composed of *E. coli* strains has been used to record the order, duration and timing of chemical events (Hsiao et al., 2016). Here the stochasticity of the intercellular processes was harnessed to do the encoding of memories at the population level. This facilitated functionality not possible at the level of individual cells, including recording the order and time difference between two events and the start time and pulse width of an inducer signal. A bistable switch was built across two distinct cell types, controllable by two different yeast pheromones, that switched the community between two states (Urrios et al., 2016). The simulation of a design for a flip flop memory device distributed over four populations of cells show that its function is robust to changes in parameters and that circuit behavior can be tuned by changing experimental conditions (Sardanyés et al., 2015). This design leveraged the modularity possible with a microbial community, the flip flop logical circuit was broken down into four modules that were distributed across the four cell types and the modules were wired together using diffusible molecules. Another computational investigation showed how a co-culture of two bacterial strains could be used to do associative learning, with both short- and long-term memory (Macia et al., 2017). The microbial community responds to an input (A) but not a second input (B) unless both A and B have been simultaneously present in the

past. This results in a computational system that can respond differently depending on its history. Here the modularity of a co-culture was exploited again to prevent cross talk and simplify the genetic constructs required by distributing different logical components into different populations.

Edge Detection

A genetic light sensor and communication with diffusible signals was used to create a lawn of *E. coli* capable of edge detection (Tabor et al., 2009), an important algorithm in image recognition and artificial intelligence. An image is applied to the lawn by placing an image mask in front of a light source. Cells produce a quorum sensing molecule when not exposed to light and fluoresce when exposed to both light and the quorum signal; a combination which is only present at a light-dark interface.

Reconfigurable "Hardware"

Unlike electronic computers, biological systems are able to change their "hardware" depending on the task at hand by, for example dynamically controlling the constituents of a community (**Figure 4C**). Two independent auxotrophic *E. coli* populations have been designed so that their growth is tuneable by inducing production of amino acids (Kerner et al., 2012). Using a community of microbes that inhabit slightly different temperature niches, a temperature cycling scheme is able to dynamically tune the community (Lin et al., 2020). Methods of intrinsic community composition control can be built into cells genetically. This has been done using self-inhibition using quorum molecule signaling (Dinh et al., 2020). One strain produces an N-Acyl homoserine lactone (AHL) quorum molecule which leads to a reduction in its own growth rate when at a high concentration. This was used to control co-culture composition as the two strains of cells grew together and resulted in a 60% increase in productivity. Simulation results also show that a population of cells containing a reconfigurable logic gate that can be switched between NOR and NAND behavior (Goñi-Moreno and Amos, 2012). Furthermore, a rock-paper-scissors system of three populations of *E. coli* that cyclically inhibit one another, combined with population dependant synchronized lysis, shows the capability to cycle the community composition through the three strains (Liao et al., 2019). This was built with the intention of plasmid stability, but by using three functionally different strains a community could be built that can be cycled between different functions as required.

Classification

Classifiers aim to identify which category an observation belongs to. Biological classifiers have been built to identify cancer cells using miRNA (Xie et al., 2011; Mohammadi et al., 2017). A key concept in machine learning is the use of ensemble methods. These combine the output of many individual weak classifiers, which perform at least slightly better than random choice, and produce an overall output with much greater accuracy. This methodology can naturally be applied to a community of cells, where each cell contains a genetically encoded weak classifier and the overall community output is computed by combining the individual outputs of all cells (**Figure 4D**). This approach has been investigated *in silico*. For each data point in a training data set a heterogenous population of cells containing weak classifiers vote on the answer (Kanakov et al., 2015). The community learns as cells are stochastically pruned from the population; cells that voted incorrectly are removed with a higher probability. A multi-input classifier composed of a community of cells containing either a linear or a bell-shaped classifier was simulated and found to be able to represent practically arbitrary shapes in the input space (Kanakov et al., 2015). Other numerical results on a similar population of cells showed that complex classification problems could be tackled (Didovyk et al., 2015). In both papers, soft training, in which cells are removed with a certain probability according to their decision, outperforms hard training, in which incorrect cells are always removed and correct cells are always retained.

Noise Reduction

Noise in biological systems can arise due to a number of intracellular and environmental reasons. Although noise seems to be important to the functioning of many biological systems (Rao et al., 2002), engineered systems are required to be predictable and therefore resilient to noise. Mechanisms have been developed to reduce gene expression noise within cells. Buffer systems have been built using miRNA to degrade mRNA transcripts in a controlled manner, reducing gene expression variability at the cost of a reduced maximal expression (Strovas et al., 2014). A genetic integral feedback controller with the potential to maintain cellular system variables at desired levels despite noisy dynamics was shown to be able to control growth rate (Aoki et al., 2019). Mundt et al. (2018) dampened noise in gene expression by tuning transcription rates and the degradation rate of mRNA. Instead of implementing a complex intracellular mechanism to reduce noise, computational communities have the potential to repeat a computation over multitudes of cells and integrate the results by reaching a global consensus, vastly improving the robustness of the computation to noise inside any single cell. This is particularly important in analog computing as the continuous states of an analog computer are susceptible to small perturbations (Sarpeshkar, 2014). The global consensus problem is a fundamental problem in distributed computing (Wang et al., 2014), where multiple independent agents converge to a global consensus that is robust to failure or noise of individual agents. Modeling work on a community of agents, resembling a microbial community, that are capable of movement and local communication shows that the community is capable of solving the global consensus problem (Vining et al., 2019) and is an indication that this could be implemented in a bacterial community.

Patterning

Both multicellular organisms and communities of unicellular organisms have the ability to cooperate to produce spatial structures that allow them to better perform complex functions. The prime example of this phenomena is development in multicellular organism, in which cells containing identical DNA differentiate and organize themselves spatially to assemble a complex organism. Harnessing this capability could mean the

realization of biological computers that can self-assemble and reproduce in a manner that is not currently possible with silicon systems. The first step in this direction was taken by engineering *E. coli* "receiver cells" which respond to a quorum molecule with a band detect activation (Basu et al., 2005). Sources of the quorum molecule could then be used to produce different patterns of fluorescence in a lawn of *E. coli*. This approach was complemented by the development of quorum molecule producing "sender" cells (Basu et al., 2004). Work has also been undertaken using senders and receivers to produce 3D patterning of mammalian cells (Carvalho et al., 2014). It is possible that sender and receiver cells could be combined to produce dynamic pattern formation in response to environmental changes. The value of using computational modeling to investigate pattern formation and design spatially structured synthetic communities has been shown (**Figure 4B**; Schaerli et al., 2014). Here the space of two and three-node, stripe forming networks was investigated computationally, and used to inform wet laboratory experiments. Further computational investigation using the modeling platform GRO (Jang et al., 2012; Gutiérrez et al., 2017) acts as a proof of concept for the design of bacterial colonies capable of self-assembling into spatial structures including L and T shapes (Pascalie et al., 2016). It has also been shown that synthetic communities engineered to grow with a ring shaped pattern show scale invariance, similar to natural systems (Cao et al., 2016). An artificial symmetry breaking mechanism was combined with domain specific cellular regulation resulting in artificial patterning and cell differentiation reminiscent of a simple developmental process (Nuñez et al., 2017). Interactions between motile and non-motile bacteria when grown together in a biofilm have been shown to trigger the emergence of complex patterns over time (Xiong et al., 2020).

CHALLENGES (AND POTENTIAL SOLUTIONS) IN DESIGNING AND IMPLEMENTING DISTRIBUTED SYNTHETIC BIOLOGICAL SYSTEMS

Although several steps have been taken down the path toward distributed synthetic biological systems, some hurdles stand in the way of the paradigm becoming ubiquitous in the field.

Building Stable Communities

In distributed computing the execution of tasks is dependent upon limited resources such as available memory or processors. Tasks are allocated resources by central schedulers upon request, aiming to distribute resources in a "fair" and "efficient" manner while accounting for task priority (**Figure 5A**; Haupt, 1989). Similarly, distributed biological systems in liquid cultures are constrained by limited resources including carbon sources and essential amino acids (Jacob and Monod, 1961). Microbes tend to maximize growth, consuming the resources in a system without request. Biological systems lack a central scheduler to allocate resources fairly between subpopulations, multiple subpopulations sharing an environment therefore compete for limited resources, a single subpopulation with the highest fitness will drive the others to extinction, this is a principle known as competitive exclusion (Butler and Wolkowicz, 1985). Evidence from natural microbial systems and ecological studies shows us stability can arise through interactions between subpopulations. These interactions alter the resource demand of a subpopulation by changing its population density or metabolic activity (**Figure 5B**). Both cooperative and competitive interactions are important for stabilizing communities (Czárán

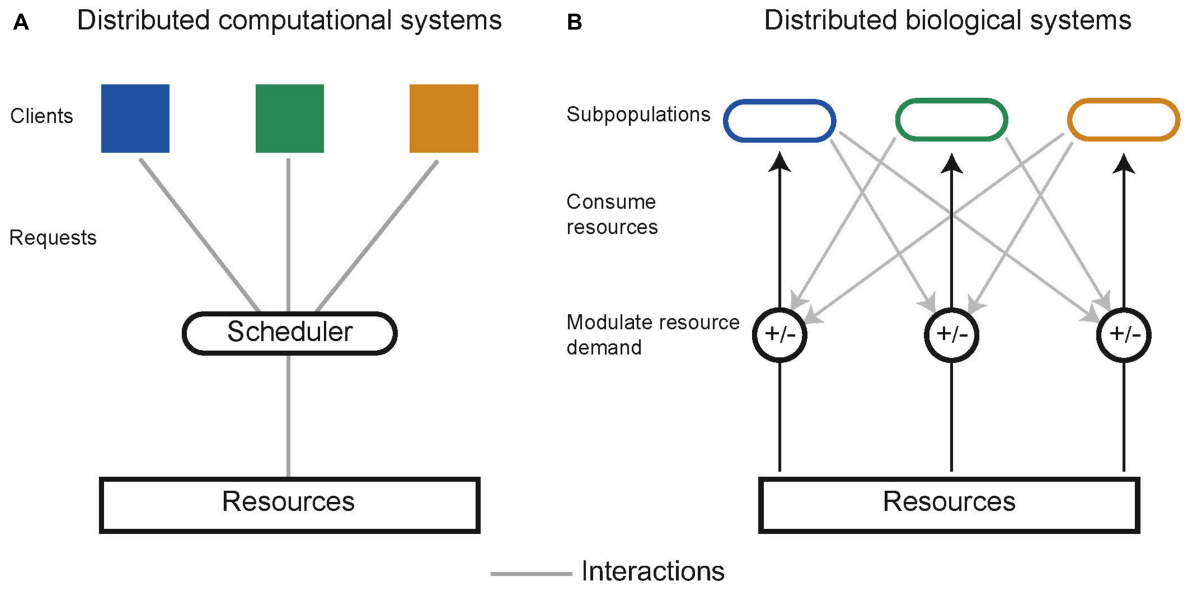
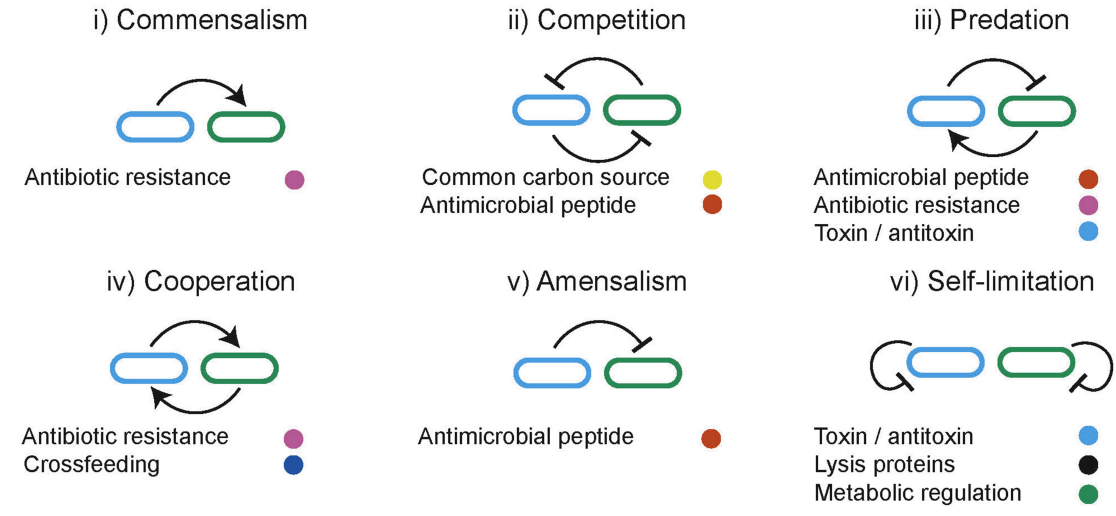


FIGURE 5 | (A) Schematic of resource allocation in distributed computational systems. Tasks communicate with a central scheduler which in turn allocates resources to tasks. **(B)** Resource allocation in distributed biological systems is decentralized. Subpopulations communicate and interact with one another to modulate the demand for resources, which can optimally allocate resources and prevent competitive exclusion.



Article	i)	ii)	iii)	iv)	v)	vi)
Balagadde et al. 2008		●	●			
Kong et al. 2018	●	● ●	● ●	●	●	
Fedorec et al. 2019		●			●	
Scott et al. 2017		●				●
F. Liu et al. 2019		●	● ●			
Stephens et al. 2019		●				●
Dinh, Chen and Prather 2020		●				●
Ziesack et al. 2019		●		●		
Zhang and Reed 2014		●		●		
Zhou et al. 2015				●		

FIGURE 6 | Illustration of ecological interactions that can be used to dynamically manipulate resource allocation within co-cultures. Table summarizes the ecological interactions engineered in discussed studies where the colored dots refer to the methods used to implement the interaction.

et al., 2002; Hibbing et al., 2010; Freilich et al., 2011; Foster and Bell, 2012; Zelezniak et al., 2015; May et al., 2019). Using these principles, groups have attempted to engineer interactions as a means to ensure coexistence within synthetic microbial communities. Engineered pair-wise interactions are analogous with ecological interactions, **Figure 6** summarizes studies discussed in this section, highlighting the ecological analogs that have been demonstrated synthetically, and the tools used to implement them.

Predator-prey interactions are pervasive in nature and are well-known for producing coexistence over prolonged periods. A predator has detrimental effects on the prey, while the predator is dependent upon the prey for survival. Predator-prey interactions are prevalent in natural environments and are predicted to produce limit cycle behavior indefinitely (Volterra, 1926). Planktonic predator-prey communities have been used to demonstrate long term persistence under experimental conditions and show robustness to stochastic events (Blasius et al., 2020). In synthetic biology, predator-prey interactions can be engineered between subpopulations to enable the persistence of a community that would otherwise undergo competitive exclusion. Balagaddé et al. (2008) demonstrated the use of quorum sensing (QS) coupled with toxin/antitoxin systems to implement predator-prey-like interactions. Liu et al. (2019) used modulation of a shared environment to create predator-prey dynamics. Media containing the antibiotic chloramphenicol (CM) kills the predator strain which is dependent upon the prey strain to degrade CM. In turn, the predator strain expresses IcnA, killing the prey. By providing CM exogenously, the authors created a tuneable environmental parameter that is directly involved in the social interaction.

The expression and secretion of antimicrobial peptides (AMPs) can be used to engineer amensal effects on sensitive subpopulations within a community. The signaling and AMP properties of nisin have been used with a second AMP to produce a modular system for building predatory, cooperative and competitive interactions in *Lactococcus lactis* (Kong et al., 2018). AMP microcin-V has been used with QS regulation to stabilize a two species community by engineering a single strain to have an amensal effect on another faster growing strain (Fedorec et al., 2019). Co-existence can also be achieved without engineering interactions between subpopulations. Using two strains with orthogonal QS controlled expression of lysis proteins, Scott et al. (2017) ensured that neither strain could grow beyond a threshold, thereby preventing competitive exclusion occurring through self-limitation. This effectively behaves as a block on the maximal resource occupation by any single subpopulation.

Controlling the flux of metabolites essential for growth through different pathways has been demonstrated in a monoculture using QS. The expression of a burdensome heterologous circuit was regulated, switching between "growth mode" and "production mode" in response to population density (Gupta et al., 2017). It has also been demonstrated that control over the growth rates of one strain, through modulating expression of the *ptsH* sugar transport gene, can be used to control the composition of co-cultures (Stephens et al., 2019). A similar approach was used to distribute a naringenin production pathway between two strains (Dinh et al., 2020). By using QS to self-regulate the growth of a high growth rate subpopulation combined with a low growth rate population the authors were able to generate a stable co-culture and significantly improve production yields. These examples prevent overutilization of a resource by a single strain by modulating growth directly.

Metabolic interdependencies are pervasive in microbial communities and are an important interaction that can be used to produce stable co-existence (Zelezniak et al., 2015). Interdependencies decouple the growth of a subpopulation from the limited environmental resource. Instead resources must be made available by another subpopulation in the system. Previously discussed modeling frameworks can be used to inform cross-feeding strategies and identify conditions that encourage establishment of cooperative communities (Pacheco et al., 2019). A sustainable multi-species system was generated by engineering amino acid auxotrophies and overproduction in *E. coli, Salmonella typhimurium, Bacteroides fragilis*, and *Bacteroides thetaiotaomicron* (Ziesack et al., 2019), forcing dependencies between community members. Synthetic metabolic interdependent co-cultures have been shown to undergo significant adaptation over long term co-cultures resulting in improved growth rates (Zhang and Reed, 2014). An *E. coli* – *S. cerevisiae* stabilized co-culture has been demonstrated on xylose based feed stock (Zhou et al., 2015). *E. coli* metabolizes xylose producing acetate, which is in turn used by *S. cerevisiae*. Since acetate is an inhibitor of *E. coli* growth, it is dependent on *S. cerevisiae* to remove it from the environment.

These ecological interactions manipulate the resource consumption of each subpopulations by regulating population densities and metabolic activity, providing opportunities for autonomously regulated systems. This contrasts with the centralized resource allocation commonly seen in computing. A hybrid of these approaches has been achieved through external regulation of the environment to maintain coexistence of competing. Reinforcement learning was used to train an agent that controls the supply of essential nutrients to two competing auxotrophs in a chemostat, in principle demonstrating the use of a centralized controller to regulate a biological system (Treloar et al., 2020).

Orthogonal and Directed Communication

Quorum sensing (QS) systems are a key set of tools that enable us to engineer communications between and within subpopulations of a community. QS systems consist of one or more proteins that produce small, freely diffusible molecules. These quorum molecules bind to regulatory proteins that can activate or repress gene expression at specific promoters (Miller and Bassler, 2001). QS can be used to regulate the expression of genes in a population, but because cells broadcast to all other cells in their vicinity, each communication channel must utilize a different quorum molecule. However, in practice there are a limited number of QS systems available and even distinct QS systems may not be totally orthogonal (Grant et al., 2016). Kylilis et al. (2018) performed a comprehensive characterization of the crosstalk between several QS systems in conjunction with computational tools to identify conditions in which channels can be used simultaneously. Moreover, these tools can be used to account for and incorporate crosstalk into system design. Studies have also reduced crosstalk through rational sequence mutation (Grant et al., 2016; Scott and Hasty, 2016). Quorum quenching refers to the enzymatic degradation of quorum molecules allowing controllable degradation of QS molecules in a system. The AiiA quorum quenching enzyme and LuxI quorum molecule synthase have been used to produce oscillations in a bacterial population (Danino et al., 2010) and to introduce a negative feedback layer in a two strain oscillating system (Chen et al., 2015).

While QS is the dominant choice for engineering communication in synthetic biology, alternative channels are being developed. The $\gamma\gamma$-butyrolactone system (derived from *Streptomyces coelicolor*) has been demonstrated E. coli to implement orthogonal signaling that can be used alongside QS (Biarnes-Carrera et al., 2018). Other signaling channels exist between different species of bacteria (Hughes and Sperandio, 2008), however, the synthetic biology field has yet to embrace these channels to the same degree as QS for controlling. Signal response mechanisms have also been observed between the host and bacteria of the human gut through polyamine compounds, highlighting the clear potential for host-community interfacing (Lopes and Sourjik, 2018).

A potential limitation of quorum sensing based approaches is that communication is non-specific and global. Cells communicate through broadcast signaling which, in contrast to the targeted information transfer afforded by electrical wires, means that each communication molecule in a bacterial community must be different in order to address different subpopulations. This acts as a constraint on the possible complexity of a distributed computation for a given number of quorum sensing molecules. In electrical engineering, circuits are only marginally constrained by the number of wires and are often optimized to minimize the number of logic gates. An analogous approach has been carried out by using an evolutionary algorithm to optimize a distributed bacterial community to reduce the number of wires (Macia and Sole, 2014). In optimized electronic circuits NOR and NAND gates are widely used. Interestingly, when optimizing for the communication constraints within a microbial community using quorum sensing, a high number of non-standard logic gates (NIMPLIES, NOT, and AND) are selected, highlighting the differences between electrical and biological computing. The optimal design of computational communities will require new tools, such as an algorithm to distribute genetic NOR gates

among cell populations communicating via diffusible molecules (Al-Radhawi et al., 2020).

Other communication channels could be exploited to overcome the wiring problem. For example, the transfer of DNA between bacterial cells. The packaging and transfer of DNA messages using bacteriophage has been demonstrated in *E. coli* (Ortiz and Endy, 2012). Although this is still a broadcast approach, as in wireless networking, the amount of information that one can encode may allow selective reading of the message, for example using non-native RNA polymerases or state dependent expression. Alternatively, direct message passing has been achieved by bacterial conjugation (Goñi-Moreno et al., 2013). The sharing of conjugative plasmids has been used to design, *in silico*, a community of distributed NOR gates wired together for a population level XOR gate (Goñi-Moreno et al., 2013). Finally, electrical signaling is another potential method of communication that could allow specific message passing at a much higher speed than conjugation. Natural bacterial communities can communicate using ion channel based electrical waves similar to neurons (Prindle et al., 2015; Martinez-Corral et al., 2019) and networks of fibrous cables are used as electrical communication channels (Meysman et al., 2019). It will be exciting to see how synthetic biology can harness these behaviors over the coming years.

CONCLUSION

Distributed systems are ubiquitous in modern computing, from the Internet to scientific high-performance computing. Thinking about biological systems through this lens will offer unique opportunities in the development of biological computing. A great deal of effort has been put into developing *de novo* biological systems that compute and some magnificent advances have been made. However, we are, and will remain, fundamentally limited in the systems we can build if we stick to the prevailing paradigm of engineering a single strain to do everything. The prevalence of genetically and phenotypically diverse distributed systems in nature is clear, and in this review we have highlighted some examples that we believe to be particularly relevant in the pursuit of engineering biological computation. While the prospective rewards of distributed systems cannot be overlooked, challenges in the establishment of robust and controllable distributed systems are significant but not insurmountable.

The majority of engineered biological communities demonstrated to date have focused on the establishment of co-existing populations. Building these methodologies and experimental frameworks will allow us to take the next step in focusing on exploiting communities as distributed systems. The demonstration of the advantages held by distributed systems in functionality and productivity over a monoculture will be paramount for advancing the field. The fundamental differences between microbial communities and computer networks (competition, communication and naturally analog processes) highlight opportunities for the development and advancement of the theory. These differences also present some of the greatest opportunities for functionality that is hard to achieve in digital hardware, including adaptability, self-assembly and analog information processing (Grozinger et al., 2019). Many of the competitive advantages communities have in nature are due to the ability to adapt to noisy, diverse and changing environments.

Although success has been found in overcoming these limitations and implementing familiar digital computations, focus should also be on exploiting these capabilities to build useful biological computers. Evidence indicates that the optimal organization of a bacterial computer differs from that of a digital computer (Macia and Sole, 2014). This means that new methodologies will have to be developed, extending our current capabilities of automatic circuit design in single cells (Nielsen et al., 2016) to computational communities. To realize the advantages of biological computing we will have to move away from replicating feats of electrical engineering. We envisage the biological computers will find their application niche in interfacing with biological systems. Immediately attractive applications lie in disease diagnosis through biosensing and reactive treatment through *in situ* production of biological material (Slomovic et al., 2015; Courbet et al., 2016). An open challenge to the field lies in converting the immense progress demonstrated in laboratory environments into real-world applications, validating with demonstrable improvements.

AUTHOR CONTRIBUTIONS

BK, NT, and AF wrote the first draft of the manuscript. All authors contributed to the conception of this review, contributed to manuscript revision, and read and approved the submitted version.

FUNDING

NT, AF, and CB received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant No. 770835). BK was funded through the BBSRC LIDo Doctoral Training Partnership.

REFERENCES

Abisado, R. G., Benomar, S., Klaus, J. R., Dandekar, A. A., and Chandler, J. R. (2018). Bacterial quorum sensing and microbial community interactions. *mBio* 9:e02331-17. doi: 10.1128/mBio.02331-17

Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266, 1021–1024. doi: 10.1126/science.7973651

Alnahhas, R. N., Winkle, J. J., Hirning, A. J., Karamched, B., Ott, W., Josiæ, K., et al. (2019). Spatiotemporal dynamics of synthetic microbial consortia in microfluidic devices. *ACS Synth. Biol.* 8, 2051–2058. doi: 10.1021/acssynbio.9b00146

Al-Radhawi, M. A., Tran, A. P., Ernst, E. A., Chen, T., Voigt, C. A., and Sontag, E. D. (2020). Distributed implementation of Boolean functions by transcriptional synthetic circuits. *bioRxiv*[Preprint]. doi: 10.1101/2020.04.21.053231

Amdahl, G. M. (1967). "Validity of the single processor approach to achieving large scale computing capabilities," in *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, (New York, NY: Association for Computing Machinery), 483–485. doi: 10.1145/1465482.1465560

Anderson, J. C., Voigt, C. A., and Arkin, A. P. (2007). Environmental signal integration by a modular AND gate. *Mol. Syst. Biol.* 3:133. doi: 10.1038/msb4100173

Angulo, M. T., Moog, C. H., and Liu, Y.-Y. (2019). A theoretical framework for controlling complex microbial communities. *Nat. Commun.* 10:1045. doi: 10.1038/s41467-019-08890-y

Aoki, S. K., Lillacci, G., Gupta, A., Baumschlager, A., Schweingruber, D., and Khammash, M. (2019). A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* 570, 533–537. doi: 10.1038/s41586-019-1321-1

Apter, M., and Wolpert, L. (1965). Cybernetics and development I. Information theory. *J. Theor. Biol.* 8, 244–257. doi: 10.1016/0022-5193(65)90075-5

Attiya, H., and Welch, J. (2004). *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. Hoboken, NJ: John Wiley & Sons, Inc, doi: 10.1002/0471478210

Balagaddé, F. K., Song, H., Ozaki, J., Collins, C. H., Barnet, M., Arnold, F. H., et al. (2008). A synthetic *Escherichia coli* predator–prey ecosystem. *Mol. Syst. Biol.* 4:187. doi: 10.1038/msb.2008.24

Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., and Weiss, R. (2005). A synthetic multicellular system for programmed pattern formation. *Nature* 434, 1130–1134. doi: 10.1038/nature03461

Basu, S., Mehreja, R., Thiberge, S., Chen, M.-T., and Weiss, R. (2004). Spatiotemporal control of gene expression with pulse-generating networks. *Proc. Natl. Acad. Sci. U.S.A.* 101, 6355–6360. doi: 10.1073/pnas.0307571101

Becskei, A., and Serrano, L. (2000). Engineering stability in gene networks by autoregulation. *Nature* 405, 590–593. doi: 10.1038/35014651

Ben-Jacob, E. (2003). Bacterial self–organization: co–enhancement of complexification and adaptability in a dynamic environment. *Philos. Trans. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* 361, 1283–1312. doi: 10.1098/rsta.2003.1199

Ben-Jacob, E., Cohen, I., and Gutnick, D. L. (1998). Cooperative organization of bacterial colonies: from genotype to morphotype. *Annu. Rev. Microbiol.* 52, 779–806. doi: 10.1146/annurev.micro.52.1.779

Ben-Jacob, E., and Levine, H. (2006). Self-engineering capabilities of bacteria. *J. R. Soc. Interface* 3, 197–214. doi: 10.1098/rsif.2005.0089

Bennett, M. R., and Hasty, J. (2009). Microfluidic devices for measuring gene network dynamics in single cells. *Nat. Rev. Genet.* 10, 628–638. doi: 10.1038/nrg2625

Biarnes-Carrera, M., Lee, C.-K., Nihira, T., Breitling, R., and Takano, E. (2018). Orthogonal regulatory circuits for *Escherichia coli* based on the γ-butyrolactone system of *Streptomyces coelicolor*. *ACS Synth. Biol.* 7, 1043–1055. doi: 10.1021/acssynbio.7b00425

Biggs, M. B., Medlock, G. L., Kolling, G. L., and Papin, J. A. (2015). Metabolic network modeling of microbial communities. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 7, 317–334. doi: 10.1002/wsbm.1308

Blasius, B., Rudolf, L., Weithoff, G., Gaedke, U., and Fussmann, G. F. (2020). Long-term cyclic persistence in an experimental predator–prey system. *Nature* 577, 226–230. doi: 10.1038/s41586-019-1857-0

Boeing, P., Leon, M., Nesbeth, D. N., Finkelstein, A., and Barnes, C. P. (2018). Towards an aspect-oriented design and modelling framework for synthetic biology. *Processes* 6:167. doi: 10.3390/pr6090167

Boles, B. R., Thoendel, M., and Singh, P. K. (2004). Self-generated diversity produces "insurance effects" in biofilm communities. *Proc. Natl. Acad. Sci. U.S.A.* 101, 16630–16635. doi: 10.1073/pnas.0407460101

Bonnet, J., Subsoontorn, P., and Endy, D. (2012). Rewritable digital data storage in live cells via engineered control of recombination directionality. *Proc. Natl. Acad. Sci. U.S.A.* 109, 8884–8889. doi: 10.1073/pnas.1202344109

Boo, A., Ellis, T., and Stan, G. B. (2019). Host-aware synthetic biology. *Curr. Opin. Syst. Biol.* 14, 66–72. doi: 10.1016/j.coisb.2019.03.001

Buffie, C. G., Bucci, V., Stein, R. R., McKenney, P. T., Ling, L., Gobourne, A., et al. (2015). Precision microbiome reconstitution restores bile acid mediated resistance to Clostridium difficile. *Nature* 517, 205–208. doi: 10.1038/nature13828

Butler, G. J., and Wolkowicz, G. S. K. (1985). A mathematical model of the chemostat with a general class of functions describing nutrient uptake. *SIAM J. Appl. Math.* 45, 138–151. doi: 10.1137/0145006

Cao, Y., Ryser, M. D., Payne, S., Li, B., Rao, C. V., and You, L. (2016). Collective space-sensing coordinates pattern scaling in engineered bacteria. *Cell* 165, 620–630. doi: 10.1016/j.cell.2016.03.006

Carbonell-Ballestero, M., Garcia-Ramallo, E., Montañez, R., Rodriguez-Caso, C., and Macía, J. (2016). Dealing with the genetic load in bacterial synthetic biology circuits: convergences with the Ohm's law. *Nucleic Acids Res.* 44, 496–507. doi: 10.1093/nar/gkv1280

Cardelli, L., and Gordon, A. D. (1998). "Mobile ambients," in *Proceedings of the First International Conference on Foundations of Software Science and Computation Structure*, (Berlin: Springer-Verlag), 140–155.

Carvalho, A., Menendez, D. B., Senthivel, V. R., Zimmermann, T., Diambra, L., and Isalan, M. (2014). Genetically encoded sender–receiver system in 3D mammalian cell culture. *ACS Synth. Biol.* 3, 264–272. doi: 10.1021/sb400053b

Ceroni, F., Boo, A., Furini, S., Gorochowski, T. E., Borkowski, O., Ladak, Y. N., et al. (2018). Burden-driven feedback control of gene expression. *Nat. Methods* 15, 387–393. doi: 10.1038/nmeth.4635

Chandra, T. D., and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *J. ACM* 43, 225–267. doi: 10.1145/226643.226647

Chen, Y., Kim, J. K., Hirning, A. J., Josi, K., and Bennett, M. R. (2015). Emergent genetic oscillations in a synthetic microbial consortium. *Science* 349, 986–989. doi: 10.1126/science.aaa3794

Chen, Y.-J., Liu, P., Nielsen, A. A. K. K., Brophy, J. A. N. N., Clancy, K., Peterson, T., et al. (2013). Characterization of 582 natural and synthetic terminators and quantification of their design constraints. *Nat. Methods* 10, 659–664. doi: 10.1038/nmeth.2515

Cherry, K. M., and Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* 559, 370–376. doi: 10.1038/s41586-018-0289-6

Christian, J. L. (2012). Morphogen gradients in development: from form to function. *Wiley Interdiscip. Rev. Dev. Biol.* 1, 3–15. doi: 10.1002/wdev.2

Church, G. M., Gao, Y., and Kosuri, S. (2012). Next-generation digital information storage in DNA. *Science* 337, 1628–1628. doi: 10.1126/science.1226355

Connell, J. L., Ritschdorff, E. T., Whiteley, M., and Shear, J. B. (2013). 3D printing of microscopic bacterial communities. *Proc. Natl. Acad. Sci. U.S.A.* 110, 18380–18385. doi: 10.1073/pnas.1309729110

Courbet, A., Renard, E., and Molina, F. (2016). Bringing next−generation diagnostics to the clinic through synthetic biology. *EMBO Mol. Med.* 8, 987–991. doi: 10.15252/emmm.201606541

Couto, J. M., McGarrity, A., Russell, J., and Sloan, W. T. (2018). The effect of metabolic stress on genome stability of a synthetic biology chassis *Escherichia coli* K12 strain. *Microb. Cell Fact.* 17:8. doi: 10.1186/s12934-018-0858-2

Coyte, K. Z., Schluter, J., and Foster, K. R. (2015). The ecology of the microbiome: networks, competition, and stability. *Science* 350, 663–666. doi: 10.1126/science.aad2602

Currin, A., Korovin, K., Ababi, M., Roper, K., Kell, D. B., Day, P. J., et al. (2017). Computing exponentially faster: implementing a non-deterministic universal Turing machine using DNA. *J. R. Soc. Interface* 14:20160990. doi: 10.1098/rsif.2016.0990

Czárán, T. L., Hoekstra, R. F., and Pagie, L. (2002). Chemical warfare between microbes promotes biodiversity. *Proc. Natl. Acad. Sci.* 99, 786–790. doi: 10.1073/pnas.012399899

Dalchau, N., Szép, G., Hernansaiz-Ballesteros, R., Barnes, C. P., Cardelli, L., Phillips, A., et al. (2018). Computing with biological switches and clocks. *Nat. Comput.* 17, 761–779. doi: 10.1007/s11047-018-9686-x

Daniel, R., Rubens, J. R., Sarpeshkar, R., and Lu, T. K. (2013). Synthetic analog computation in living cells. *Nature* 497, 619–623. doi: 10.1038/nature12148

Danino, T., Mondragón-Palomino, O., Tsimring, L., and Hasty, J. (2010). A synchronized quorum of genetic clocks. *Nature* 463, 326–330. doi: 10.1038/nature08753

Darlington, A. P. S. S., Kim, J., Jiménez, J. I., and Bates, D. G. (2018). Dynamic allocation of orthogonal ribosomes facilitates uncoupling of co-expressed genes. *Nat. Commun.* 9:695. doi: 10.1038/s41467-018-02898-6

Davies, D. G., Parsek, M. R., Pearson, J. P., Iglewski, B. H., Costerton, J. W., and Greenberg, E. P. (1998). The involvement of cell-to-cell signals in the development of a bacterial biofilm. *Science* 280, 295–298. doi: 10.1126/science.280.5361.295

Davis, J. H., Rubin, A. J., and Sauer, R. T. (2011). Design, construction and characterization of a set of insulated bacterial promoters. *Nucleic Acids Res.* 39, 1131–1141. doi: 10.1093/nar/gkq810

Deng, J., Krause, J., Berg, A. C., and Fei-Fei, L. (2012). "Hedging your bets: optimizing accuracy-specificity trade-offs in large scale visual recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Providence, RI: IEEE), 3450–3457. doi: 10.1109/CVPR.2012.6248086

Deuchar, E. M. (1970). Diffusion in embryogenesis. *Nature* 225, 671–671. doi: 10.1038/225671b0

Didovyk, A., Kanakov, O. I., Ivanchenko, M. V., Hasty, J., Huerta, R., and Tsimring, L. (2015). Distributed classifier based on genetically engineered bacterial cell cultures. *ACS Synth. Biol.* 4, 72–82. doi: 10.1021/sb500235p

Dinh, C. V., Chen, X., and Prather, K. L. J. (2020). Development of a quorum-sensing based circuit for control of coculture population composition in a naringenin production system. *ACS Synth. Biol.* 9, 590–597. doi: 10.1021/acssynbio.9b00451

D'Souza, G., and Kost, C. (2016). Experimental evolution of metabolic dependency in bacteria. *PLoS Genet.* 12:e1006364. doi: 10.1371/journal.pgen.1006364

Dubuis, J. O., Tkacik, G., Wieschaus, E. F., Gregor, T., and Bialek, W. (2013). Positional information, in bits. *Proc. Natl. Acad. Sci. U.S.A.* 110, 16301–16308. doi: 10.1073/pnas.1315642110

Elowitz, M. B., and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335–338. doi: 10.1038/35002125

Fedorec, A. J. H., Karkaria, B. D., Sulu, M., and Barnes, C. P. (2019). Killing in response to competition stabilises synthetic microbial consortia. *bioRxiv* [Preprint]. doi: 10.1101/2019.12.23.887331

Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *J. ACM* 32, 374–382. doi: 10.1145/3149.214121

Flemming, H. C., Wingender, J., Szewzyk, U., Steinberg, P., Rice, S. A., and Kjelleberg, S. (2016). Biofilms: an emergent form of bacterial life. *Nat. Rev. Microbiol.* 14, 563–575. doi: 10.1038/nrmicro.2016.94

Flynn, M. J. (1972). Some computer organizations and their effectiveness. *IEEE Trans. Comput.* 21, 948–960. doi: 10.1109/TC.1972.5009071

Foster, K. R., and Bell T. (2012). Competition, not cooperation, dominates interactions among culturable microbial species. *Curr. Biol.* 22, 1845–1850. doi: 10.1016/j.cub.2012.08.005

Freilich, S., Zarecki R., Eilam, O., Segal, E. S., Henry, C. S., Kupiec M., et al. (2011). Competitive and cooperative metabolic interactions in bacterial communities. *Nat. Commun.* 2:589. doi: 10.1038/ncomms1597

García, A. P., and Rodríguez-Patón, A. (2015). "BactoSim – an individual-based simulation environment for bacterial conjugation," in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection*, eds Y. Demazeau, K. S. Decker, J. Bajo Pérez, and F. de la Prieta (Cham: Springer International Publishing), 275–279. doi: 10.1007/978-3-319-18944-4_26

Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403, 339–342. doi: 10.1038/35002131

Giessen, T. W., and Silver, P. A. (2016). Encapsulation as a strategy for the design of biological compartmentalization. *J. Mol. Biol.* 428, 916–927. doi: 10.1016/j.jmb.2015.09.009

Gill, S. R., Pop, M., DeBoy, R. T., Eckburg, P. B., Turnbaugh, P. J., Samuel, B. S., et al. (2006). Metagenomic analysis of the human distal gut microbiome. *Science* 312, 1355–1359. doi: 10.1126/science.1124234

Giribet, G. (2009). Perspectives in animal phylogeny and evolution. *Syst. Biol.* 58, 159–160. doi: 10.1093/sysbio/syp002

Glick, B. R. (1995). Metabolic load and heterologous gene expression. *Biotechnol. Adv.* 13, 247–261. doi: 10.1016/0734-9750(95)00004-A

Goñi-Moreno, A., and Amos, M. (2012). A reconfigurable NAND/NOR genetic logic gate. *BMC Syst. Biol.* 6:126. doi: 10.1186/1752-0509-6-126

Goñi-Moreno, A., and Amos, M. (2015). "DiSCUS: a simulation platform for conjugation computing," in *Unconventional Computation and Natural Computation*, eds C. S. Calude and M. J. Dinneen (Cham: Springer International Publishing), 181–191. doi: 10.1007/978-3-319-21819-9_13

Goñi-Moreno, A., Amos, M., and de la Cruz, F. (2013). Multicellular computing using conjugation for wiring. *PLoS One* 8:e65986. doi: 10.1371/journal.pone.0065986

González, O., Shrikumar, H., Stankovic, J. A., and Ramamritham, K. (1997). "Adaptive fault tolerance and graceful degradation under dynamic hard real-time scheduling," in *Proceedings of the Real-Time Systems Symposium*, (San Francisco, CA: IEEE), 79–89.

Gorochowski, T. E. (2016). Agent-based modelling in synthetic biology. *Essays Biochem.* 60, 325–336. doi: 10.1042/EBC20160037

Gorochowski, T. E., Avcilar-Kucukgoze, I., Bovenberg, R. A. L., Roubos, J. A., and Ignatova, Z. (2016). A minimal model of ribosome allocation dynamics captures trade-offs in expression between endogenous and synthetic genes. *ACS Synth. Biol.* 5, 710–720. doi: 10.1021/acssynbio.6b00040

Grant, P. K., Dalchau, N., Brown, J. R., Federici, F., Rudge, T. J., Yordanov, B., et al. (2016). Orthogonal intercellular signaling for programmed spatial behavior. *Mol. Syst. Biol.* 12:849. doi: 10.15252/msb.20156590

Grozinger, L., Amos, M., Gorochowski, T. E., Carbonell, P., Oyarzún, D. A., Stoof, R., et al. (2019). Pathways to cellular supremacy in biocomputing. *Nat. Commun.* 10:5250. doi: 10.1038/s41467-019-13232-z

Guiziou, S., Mayonove, P., and Bonnet, J. (2019). Hierarchical composition of reliable recombinase logic devices. *Nat. Commun.* 10:456. doi: 10.1038/s41467-019-08391-y

Guiziou, S., Ulliana, F., Moreau, V., Leclere, M., and Bonnet, J. (2018). An automated design framework for multicellular recombinase logic. *ACS Synth. Biol.* 7, 1406–1412. doi: 10.1021/acssynbio.8b00016

Gungor-Ozkerim, P. S., Inci, I., Zhang, Y. S., Khademhosseini, A., and Dokmeci, M. R. (2018). Bioinks for 3D bioprinting: an overview. *Biomater. Sci.* 6, 915–946. doi: 10.1039/C7BM00765E

Guo, N., Huang, Y., Mai, T., Patil, S., Cao, C., Seok, M., et al. (2016). Energy-efficient hybrid analog/digital approximate computation in continuous time. *IEEE J. Solid State Circuits* 51, 1514–1524. doi: 10.1109/JSSC.2016.2543729

Gupta, A., Reizman, I. M. B., Reisch, C. R., and Prather, K. L. J. (2017). Dynamic regulation of metabolic flux in engineered bacteria using a pathway-independent quorum-sensing circuit. *Nat. Biotechnol.* 35, 273–279. doi: 10.1038/nbt.3796

Gupta, S., Ross, T. D., Gomez, M. M., Grant, J. L., Romero, P. A., and Venturelli, O. S. (2020). Investigating the dynamics of microbial consortia in spatially structured environments. *Nat. Commun.* 11:2418. doi: 10.1038/s41467-020-16200-0

Gustafson, J. L. (1988). Reevaluating Amdahl's law. *Commun. ACM* 31, 532–533. doi: 10.1145/42411.42415

Gutiérrez, M., Gregorio-Godoy, P., Pérez Del Pulgar, G., Munoz, L. E., Sáez, S., and Rodríguez-Patón, A. (2017). A new improved and extended version of the multicell bacterial simulator gro. *ACS Synth. Biol.* 6, 1496–1508. doi: 10.1021/acssynbio.7b00003

Gyorgy, A., Jiménez, J. I., Yazbek, J., Huang, H. H., Chung, H., Weiss, R., et al. (2015). Isocost lines describe the cellular economy of genetic circuits. *Biophys. J.* 109, 639–646. doi: 10.1016/j.bpj.2015.06.034

Halpern, J. Y., and Moses, Y. (1990). Knowledge and common knowledge in a distributed environment. *J. ACM* 37, 549–587. doi: 10.1145/79147.79161

Hasegawa, Y., and Arita, M. (2013). Enhanced entrainability of genetic oscillators by period mismatch. *J. R. Soc. Interface* 10:20121020. doi: 10.1098/rsif.2012.1020

Haupt, R. (1989). A survey of priority rule-based scheduling. *OR Spektrum* 11, 3–16. doi: 10.1007/BF01721162

Hewitt, C., Bishop, P., and Steiger, R. (1973). "A universal modular ACTOR formalism for artificial intelligence," in *Proceeding IJCAI'73 Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, (New York, NY: Association for Computing Machinery), 235–245. doi: 10.1145/359545.359563

Hibbing, M. E., Fuqua, C., Parsek, M. R., and Peterson, S. B. (2010). Bacterial competition: surviving and thriving in the microbial jungle. *Nat. Rev. Microbiol.* 8, 15–25. doi: 10.1038/nrmicro2259

Hillesland, K. L., Lim, S., Flowers, J. J., Turkarslan, S., Pinel, N., Zane, G. M., et al. (2014). Erosion of functional independence early in the evolution of a microbial mutualism. *Proc. Natl. Acad. Sci. U.S.A.* 111, 14822–14827. doi: 10.1073/pnas.1407986111

Hillston, J. (1996). *A Compositional Approach to Performance Modelling*. New York, NY: Cambridge University Press.

Hoffmann, S. A., Wohltat, C., Müller, K. M., and Arndt, K. M. (2017). A user-friendly, low-cost turbidostat with versatile growth rate estimation based on an extended Kalman filter. *PLoS One* 12:e0181923. doi: 10.1371/journal.pone.0181923

Holland, J. H. (1992). Genetic algorithms. *Sci. Am.* 267, 66–73.

Hong, S. H., Hegde, M., Kim, J., Wang, X., Jayaraman, A., and Wood, T. K. (2012). Synthetic quorum-sensing circuit to control consortial biofilm formation and dispersal in a microfluidic device. *Nat. Commun.* 3, 613. doi: 10.1038/ncomms1616

Horsman, C., Stepney, S., Wagner, R. C., and Kendon, V. (2014). When does a physical system compute? *Proc. R. Soc. A Math. Phys. Eng. Sci.* 470:20140182. doi: 10.1098/rspa.2014.0182

Hsiao, V., Hori, Y., Rothemund, P. W., and Murray, R. M. (2016). A population−based temporal logic gate for timing and recording chemical events. *Mol. Syst. Biol.* 12:869. doi: 10.15252/msb.20156663

Huang, Y., Xia, A., Yang, G., and Jin, F. (2018). Bioprinting living biofilms through optogenetic manipulation. *ACS Synth. Biol.* 7, 1195–1200. doi: 10.1021/acssynbio.8b00003

Huberman, B. A., Lukose, R. M., and Hogg, T. (1997). An economics approach to hard computational problems. *Science* 275, 51–54. doi: 10.1126/science.275.5296.51

Hughes, D. T., and Sperandio, V. (2008). Inter-kingdom signalling: communication between bacteria and their hosts. *Nat. Rev. Microbiol.* 6, 111–120. doi: 10.1038/nrmicro1836

Ingham, C. J., and Jacob, E. (2008). Swarming and complex pattern formation in Paenibacillus vortex studied by imaging and tracking cells. *BMC Microbiol.* 8:36. doi: 10.1186/1471-2180-8-36

Jacob, E. B., Becker, I., Shapira, Y., and Levine, H. (2004). Bacterial linguistic communication and social intelligence. *Trends Microbiol.* 12, 366–372. doi: 10.1016/j.tim.2004.06.006

Jacob, F., and Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.* 3, 318–356. doi: 10.1016/S0022-2836(61)80072-7

Jang, S. S., Oishi, K. T., Egbert, R. G., and Klavins, E. (2012). Specification and simulation of synthetic multicelled behaviors. *ACS Synth. Biol.* 1, 365–374. doi: 10.1021/sb300034m

Jayanthi, S., and Del Vecchio, D. (2011). Retroactivity attenuation in bio-molecular systems based on timescale separation. *IEEE Trans. Automat. Contr.* 56, 748–761. doi: 10.1109/TAC.2010.2069631

Johnson, B. (1984). Fault-tolerant microprocessor-based systems. *IEEE Micro* 4, 6–21. doi: 10.1109/MM.1984.291277

Kanakov, O., Kotelnikov, R., Alsaedi, A., Tsimring, L., Huerta, R., Zaikin, A., et al. (2015). Multi-input distributed classifiers for synthetic genetic circuits. *PLoS One* 10:e0125144. doi: 10.1371/journal.pone.0125144

Karslake, J., Maltas, J., Brumm, P., and Wood, K. B. (2016). Population density modulates drug inhibition and gives rise to potential bistability of treatment outcomes for bacterial infections. *PLoS Comput. Biol.* 12:e1005098. doi: 10.1371/journal.pcbi.1005098

Kehe, J., Kulesa, A., Ortiz, A., Ackerman, C. M., Thakku, S. G., Sellers, D., et al. (2019). Massively parallel screening of synthetic microbial communities. *Proc. Natl. Acad. Sci. U.S.A.* 116, 12804–12809. doi: 10.1073/pnas.1900102116

Kennedy, J., and Eberhart, R. (1995). "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, (Perth, WA: IEEE), 1942–1948.

Kerner, A., Park, J., Williams, A., and Lin, X. N. (2012). A programmable *Escherichia coli* consortium via tunable symbiosis. *PLoS One* 7:e34032. doi: 10.1371/journal.pone.0034032

Kim, J., White, K. S., and Winfree, E. (2006). Construction of an in vitro bistable circuit from synthetic transcriptional switches. *Mol. Syst. Biol.* 2:68. doi: 10.1038/msb4100099

Kim, K. H., and Sauro, H. M. (2011). Measuring retroactivity from noise in gene regulatory networks. *Biophys. J.* 100, 1167–1177. doi: 10.1016/j.bpj.2010.12.3737

Klumpp, S., Zhang, Z., and Hwa, T. (2009). Growth rate-dependent global effects on gene expression in bacteria. *Cell* 139, 1366–1375. doi: 10.1016/j.cell.2009.12.001

Kobayashi, H., Kaern, M., Araki, M., Chung, K., Gardner, T. S., Cantor, C. R., et al. (2004). Programmable cells: interfacing natural and engineered gene networks. *Proc. Natl. Acad. Sci. U.S.A.* 101, 8414–8419. doi: 10.1073/pnas.0402940101

Kondo, S., and Asai, R. (1995). A reaction–diffusion wave on the skin of the marine angelfish Pomacanthus. *Nature* 376, 765–768. doi: 10.1038/376765a0

Kong, W., Blanchard, A. E., Liao, C., and Lu, T. (2017). Engineering robust and tunable spatial structures with synthetic gene circuits. *Nucleic Acids Res.* 45, 1005–1014. doi: 10.1093/nar/gkw1045

Kong, W., Meldgin, D. R., Collins, J. J., and Lu, T. (2018). Designing microbial consortia with defined social interactions. *Nat. Chem. Biol.* 14, 821–829. doi: 10.1038/s41589-018-0091-7

Kreft, J. U., Booth, G., and Wimpenny, J. W. T. (1998). BacSim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology* 144, 3275–3287. doi: 10.1099/00221287-144-12-3275

Kussell, E. (2005). Phenotypic diversity, population growth, and information in fluctuating environments. *Science* 309, 2075–2078. doi: 10.1126/science.1114383

Kylilis, N., Tuza, Z. A., Stan, G.-B., and Polizzi, K. M. (2018). Tools for engineering coordinated system behaviour in synthetic microbial consortia. *Nat. Commun.* 9:2677. doi: 10.1038/s41467-018-05046-2

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Trans. Program. Lang. Syst.* 4, 382–401. doi: 10.1145/357172.357176

Langton, C. G. (1986). Studying artificial life with cellular automata. *Phys. D Nonlinear Phenom.* 22, 120–149. doi: 10.1016/0167-2789(86)90237-X

Lardon, L. A., Merkey, B. V., Martins, S., Dötsch, A., Picioreanu, C., Kreft, J. U., et al. (2011). iDynoMiCS: next-generation individual-based modelling of biofilms. *Environ. Microbiol.* 13, 2416–2434. doi: 10.1111/j.1462-2920.2011.02414.x

Lau, M. C. Y., Kieft, T. L., Kuloyo, O., Linage-Alvarez, B., van Heerden, E., Lindsay, M. R., et al. (2016). An oligotrophic deep-subsurface community dependent on syntrophy is dominated by sulfur-driven autotrophic denitrifiers. *Proc. Natl. Acad. Sci. U.S.A.* 113, E7927–E7936. doi: 10.1073/pnas.1612244113

Lederman, H., Macdonald, J., Stefanovic, D., and Stojanovic, M. N. (2006). Deoxyribozyme-based three-input logic gates and construction of a molecular full adder. *Biochemistry* 45, 1194–1199. doi: 10.1021/bi051871u

Lee, J. W., Gyorgy, A., Cameron, D. E., Pyenson, N., Choi, K. R., Way, J. C., et al. (2016). Creating single-copy genetic circuits. *Mol. Cell* 63, 329–336. doi: 10.1016/j.molcel.2016.06.006

Lee, K. S., Boccazzi, P., Sinskey, A. J., and Ram, R. J. (2011). Microfluidic chemostat and turbidostat with flow rate, oxygen, and temperature control for dynamic continuous culture. *Lab. Chip.* 11:1730. doi: 10.1039/c1lc20019d

Lehner, B. A. E., Schmieden, D. T., and Meyer, A. S. (2017). A straightforward approach for 3D bacterial printing. *ACS Synth. Biol.* 6, 1124–1130. doi: 10.1021/acssynbio.6b00395

Leon, M., Woods, M. L., Fedorec, A. J. H., and Barnes, C. P. (2016). A computational method for the investigation of multistable systems and its application to genetic switches. *BMC Syst. Biol.* 10:130. doi: 10.1186/s12918-016-0375-z

Lepzelter, D., and Wang, J. (2008). Exact probabilistic solution of spatial-dependent stochastics and associated spatial potential landscape for the bicoid protein. *Phys. Rev. E* 77:041917. doi: 10.1103/PhysRevE.77.041917

Li, T., Dong, Y., Zhang, X., Ji, X., Luo, C., Lou, C., et al. (2018). Engineering of a genetic circuit with regulatable multistability. *Integr. Biol.* 10, 474–482. doi: 10.1039/c8ib00030a

Liao, M. J., Din, M. O., Tsimring, L., and Hasty, J. (2019). Rock-paper-scissors: engineered population dynamics increase genetic stability. *Science* 365, 1045–1049. doi: 10.1126/science.aaw0542

Libby, E., Hébert-Dufresne, L., Hosseini, S.-R., and Wagner, A. (2019). Syntrophy emerges spontaneously in complex metabolic systems. *PLoS Comput. Biol.* 15:e1007169. doi: 10.1371/journal.pcbi.1007169

Lin, X. N., Krieger, A. G., Zhang, J., and Lin, X. N. (2020). Temperature regulation as a tool to program synthetic microbial community composition. *bioRxiv*[Preprint]. doi: 10.1101/2020.02.14.944090

Liu, F., Mao J., Lu, T., and Hua, Q. (2019). Synthetic, Context-dependent microbial consortium of predator and prey. *ACS Synth. Biol.* 8, 1713–1722. doi: 10.1021/acssynbio.9b00110

Liu, J., Martinez-Corral, R., Prindle, A., Lee, D. D., Larkin, J., Gabalda-Sagarra, M., et al. (2017). Coupling between distant biofilms and emergence of nutrient time-sharing. *Science* 356, 638–642. doi: 10.1126/science.aah4204

Long, T., Tu, K. C., Wang, Y., Mehta, P., Ong, N. P., Bassler, B. L., et al. (2009). Quantifying the integration of quorum-sensing signals with single-cell resolution. *PLoS Biol.* 7:e1000068. doi: 10.1371/journal.pbio.1000068

Lopes, J. G., and Sourjik, V. (2018). Chemotaxis of *Escherichia coli* to major hormones and polyamines present in human gut. *ISME J.* 12, 2736–2747. doi: 10.1038/s41396-018-0227-5

Lou, C., Stanton, B., Chen, Y.-J., Munsky, B., and Voigt, C. A. (2012). Ribozyme-based insulator parts buffer synthetic circuits from genetic context. *Nat. Biotechnol.* 30, 1137–1142. doi: 10.1038/nbt.2401

Macdonald, J., Li, Y., Sutovic, M., Lederman, H., Pendri, K., Lu, W., et al. (2006). Medium scale integration of molecular logic gates in an automaton. *Nano Lett.* 6, 2598–2603. doi: 10.1021/nl0620684

Macia, J., Manzoni, R., Conde, N., Urrios, A., de Nadal, E., Solé, R., et al. (2016). Implementation of complex biological logic circuits using spatially distributed multicellular consortia. *PLoS Comput. Biol.* 12:e1004685. doi: 10.1371/journal.pcbi.1004685

Macia, J., and Sole, R. (2014). How to make a synthetic multicellular computer. *PLoS One* 9:e81248. doi: 10.1371/journal.pone.0081248

Macia, J., Vidiella, B., and Solé, R. V. (2017). Synthetic associative learning in engineered multicellular consortia. *J. R. Soc. Interface* 14:20170158. doi: 10.1098/rsif.2017.0158

Madsen, J. S., Sørensen, S. J., and Burmølle, M. (2018). Bacterial social interactions and the emergence of community-intrinsic properties. *Curr. Opin. Microbiol.* 42, 104–109. doi: 10.1016/j.mib.2017.11.018

Mannan, A. A., Liu, D., Zhang, F., and Oyarzún, D. A. (2017). Fundamental design principles for transcription-factor-based metabolite biosensors. *ACS Synth. Biol.* 6, 1851–1859. doi: 10.1021/acssynbio.7b00172

Manzoni, R., Urrios, A., Velazquez-Garcia, S., de Nadal, E., and Posas, F. (2016). Synthetic biology: insights into biological computation. *Integr. Biol.* 8, 518–532. doi: 10.1039/C5IB00274E

Marsh, P. D. (2005). Dental plaque: biological significance of a biofilm and community life-style. *J. Clin. Periodontol.* 32, 7–15. doi: 10.1111/j.1600-051X.2005.00790.x

Martinez-Corral, R., Liu, J., Prindle, A., Süel, G. M., and Garcia-Ojalvo, J. (2019). Metabolic basis of brain-like electrical signalling in bacterial communities. *Philos. Trans. R. Soc. B Biol. Sci.* 374:20180382. doi: 10.1098/rstb.2018.0382

Matic, I. (2013). "Stress-induced mutagenesis in bacteria," in *Stress-Induced Mutagenesis*, ed. D. Mittelman (New York: Springer), 1–19. doi: 10.1007/978-1-4614-6280-4_1

Matyjaszkiewicz, A., Fiore, G., Annunziata, F., Grierson, C. S., Savery, N. J., Marucci, L., et al. (2017). BSim 2.0: an advanced agent-based cell simulator. *ACS Synth. Biol.* 6, 1969–1972. doi: 10.1021/acssynbio.7b00121

May, A., Narayanan, S., Alcock, J, Varsani, A., Maley, C., and Aktipis, A. (2019). Kombucha: a novel model system for cooperation and conflict in a complex multi-species microbial ecosystem. *PeerJ* 7:e7565. doi: 10.7717/peerj.7565

McLeod, C., and Nerlich, B. (2018). Synthetic biology: how the use of metaphors impacts on science, policy and responsible research [Special issue]. *Life Sci. Soc. Policy* 13:13.

McNally, C. P., and Borenstein, E. (2018). Metabolic model-based analysis of the emergence of bacterial cross-feeding via extensive gene loss. *BMC Syst. Biol.* 12:69. doi: 10.1186/s12918-018-0588-4

Menon, G., Okeke, C., and Krishnan, J. (2017). Modelling compartmentalization towards elucidation and engineering of spatial organization in biochemical pathways. *Sci. Rep.* 7:12057. doi: 10.1038/s41598-017-11081-8

Meysman, F. J. R., Cornelissen, R., Trashin, S., Bonné, R., Martinez, S. H., van der Veen, J., et al. (2019). A highly conductive fibre network enables centimetre-scale electron transport in multicellular cable bacteria. *Nat. Commun.* 10:4120. doi: 10.1038/s41467-019-12115-7

Millacura, F. A., Largey, B., and French, C. E. (2019). ParAlleL: a novel population-based approach to biological logic gates. *Front. Bioeng. Biotechnol.* 7:46. doi: 10.3389/fbioe.2019.00046

Miller, M. B., and Bassler, B. L. (2001). Quorum sensing in bacteria. *Annu. Rev. Microbiol.* 55, 165–199. doi: 10.1146/annurev.micro.55.1.165

Milner, R., Parrow, J., and Walker, D. (1992). A calculus of mobile processes. I. *Inf. Comput.* 100, 1–40. doi: 10.1016/0890-5401(92)90008-4

Miramontes, O., Solé, R. V., and Goodwin, B. C. (1993). Collective behaviour of random-activated mobile cellular automata. *Phys. D Nonlinear Phenom.* 63, 145–160. doi: 10.1016/0167-2789(93)90152-Q

Mishra, D., Rivera, P. M., Lin, A., Del Vecchio, D., and Weiss, R. (2014). A load driver device for engineering modularity in biological networks. *Nat. Biotechnol.* 32, 1268–1275. doi: 10.1038/nbt.3044

Mishra, S. (2005). A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Trans. Evol. Comput.* 9, 61–73. doi: 10.1109/TEVC.2004.840144

Mishra, S., and Bhende, C. N. (2007). Bacterial foraging technique-based optimized active power filter for load compensation. *IEEE Trans. Power Deliv.* 22, 457–465. doi: 10.1109/TPWRD.2006.876651

Mohammadi, P., Beerenwinkel, N., and Benenson, Y. (2017). Automated design of synthetic cell classifier circuits using a two-step optimization strategy. *Cell Syst.* 4, 207–218.e14. doi: 10.1016/j.cels.2017.01.003

Moon, T. S., Lou, C., Tamsir, A., Stanton, B. C., and Voigt, C. A. (2012). Genetic programs constructed from layered logic gates in single cells. *Nature* 491, 249–253. doi: 10.1038/nature11516

Morris, B. E. L., Henneberger, R., Huber, H., and Moissl-Eichinger, C. (2013). Microbial syntrophy: interaction for the common good. *FEMS Microbiol. Rev.* 37, 384–406. doi: 10.1111/1574-6976.12019

Morris, J. J., Lenski, R. E., and Zinser, E. R. (2012). The black queen hypothesis: evolution of dependencies through adaptive gene loss. *mBio* 3:e00036-12. doi: 10.1128/mBio.00036-12

Mundt, M., Anders, A., Murray, S. M., and Sourjik, V. (2018). A system for gene expression noise control in yeast. *ACS Synth. Biol.* 7, 2618–2626. doi: 10.1021/acssynbio.8b00279

Mutalik, V. K., Guimaraes, J. C., Cambray, G., Lam, C., Christoffersen, M. J., Mai, Q.-A., et al. (2013). Precise and reliable gene expression via standard transcription and translation initiation elements. *Nat. Methods* 10, 354–360. doi: 10.1038/nmeth.2404

Nealson, K. H., and Hastings, J. W. (1979). Bacterial bioluminescence: its control and ecological significance. *Microbiol. Rev.* 43, 496–518. doi: 10.1128/MMBR.43.4.496-518.1979

Newman, S. A., and Bhat, R. (2009). Dynamical patterning modules: a "pattern language" for development and evolution of multicellular form. *Int. J. Dev. Biol.* 53, 693–705. doi: 10.1387/ijdb.072481sn

Nguyen, T., Jones, T. S., Fontanarrosa, P., Mante, J. V., Zundel, Z., Densmore, D., et al. (2019). Design of asynchronous genetic circuits. *Proc. IEEE* 107, 1356–1368. doi: 10.1109/JPROC.2019.2916057

Nielsen, A. A. K. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., et al. (2016). Genetic circuit design automation. *Science* 352:aac7341. doi: 10.1126/science.aac7341

Nuñez, I. N., Matute, T. F., Del Valle, I. D., Kan, A., Choksi, A., Endy, D., et al. (2017). Artificial symmetry-breaking for morphogenetic engineering bacterial colonies. *ACS Synth. Biol.* 6, 256–265. doi: 10.1021/acssynbio.6b00149

Ortiz, M. E., and Endy, D. (2012). Engineered cell-cell communication via DNA messaging. *J. Biol. Eng.* 6:16. doi: 10.1186/1754-1611-6-16

Ozdemir, T., Fedorec, A. J. H., Danino, T., and Barnes, C. P. (2018). Synthetic biology and engineered live biotherapeutics: toward increasing system complexity. *Cell Syst.* 7, 5–16. doi: 10.1016/j.cels.2018.06.008

Pacheco, A. R., Moel, M., and Segrè, D. (2019). Costless metabolic secretions as drivers of interspecies interactions in microbial ecosystems. *Nat. Commun.* 10:103. doi: 10.1038/s41467-018-07946-9

Pandi, A., Koch, M., Voyvodic, P. L., Soudier, P., Bonnet, J., Kushwaha, M., et al. (2019). Metabolic perceptrons for neural computing in biological systems. *Nat. Commun.* 10:3880. doi: 10.1038/s41467-019-11889-0

Pantoja-Hernández, L., and Martínez-García, J. C. (2015). Retroactivity in the context of modularly structured biomolecular systems. *Front. Bioeng. Biotechnol.* 3:85. doi: 10.3389/fbioe.2015.00085

Pascalie, J., Potier, M., Kowaliw, T., Giavitto, J. L., Michel, O., Spicher, A., et al. (2016). Developmental design of synthetic bacterial architectures by morphogenetic engineering. *ACS Synth. Biol.* 5, 842–861. doi: 10.1021/acssynbio.5b00246

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst.* 22, 52–67. doi: 10.1109/MCS.2002.1004010

Pei, R., Matamoros, E., Liu, M., Stefanovic, D., and Stojanovic, M. N. (2010). Training a molecular automaton to play a game. *Nat. Nanotechnol.* 5, 773–777. doi: 10.1038/nnano.2010.194

Petri, C. A. (1966). *Communication with Automata*. New York, NY: Rome Air Development Center.

Poltak, S. R., and Cooper, V. S. (2011). Ecological succession in long-term experimentally evolved biofilms produces synergistic communities. *ISME J.* 5, 369–378. doi: 10.1038/ismej.2010.136

Potvin-Trottier, L., Lord, N. D., Vinnicombe, G., and Paulsson, J. (2016). Synchronous long-term oscillations in a synthetic gene circuit. *Nature* 538, 514–517. doi: 10.1038/nature19841

Prindle, A., Liu, J., Asally, M., Ly, S., Garcia-Ojalvo, J., and Süel, G. M. (2015). Ion channels enable electrical communication in bacterial communities. *Nature* 527, 59–63. doi: 10.1038/nature15709

Qian, F., Zhu, C., Knipe, J. M., Ruelas, S., Stolaroff, J. K., DeOtte, J. R., et al. (2019). Direct writing of tunable living inks for bioprocess intensification. *Nano Lett.* 19, 5829–5835. doi: 10.1021/acs.nanolett.9b00066

Qian, L., and Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *Science* 332, 1196–1201. doi: 10.1126/science.1200520

Qian, L., Winfree, E., and Bruck, J. (2011). Neural network computation with DNA strand displacement cascades. *Nature* 475, 368–372. doi: 10.1038/nature10262

Rao, C. V., Wolf, D. M., and Arkin, A. P. (2002). Control, exploitation and tolerance of intracellular noise. *Nature* 420, 231–237. doi: 10.1038/nature01258

Regot, S., Macia, J., Conde, N., Furukawa, K., Kjellén, J., Peeters, T., et al. (2011). Distributed biological computation with multicellular engineered networks. *Nature* 469, 207–211. doi: 10.1038/nature09679

Ruaud, A., Esquivel-Elizondo, S., de la Cuesta-Zuluaga, J., Waters, J. L., Angenent, L. T., Youngblut, N. D., et al. (2020). Syntrophy via interspecies H 2 transfer between christensenella and methanobrevibacter underlies their global cooccurrence in the human gut. *mBio* 11:e03235-19. doi: 10.1128/mBio.03235-19

Rugbjerg, P., Myling-Petersen, N., Porse, A., Sarup-Lytzen, K., and Sommer, M. O. A. (2018). Diverse genetic error modes constrain large-scale bio-based production. *Nat. Commun.* 9:787. doi: 10.1038/s41467-018-03232-w

Sardanyés, J., Bonforti, A., Conde, N., Solé, R., and Macia, J. (2015). Computational implementation of a tunable multicellular memory circuit for engineered eukaryotic consortia. *Front. Physiol.* 6:281. doi: 10.3389/fphys.2015.00281

Sarpeshkar, R. (2014). Analog synthetic biology. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 372:20130110. doi: 10.1098/rsta.2013.0110

Sauer, K., Camper, A. K., Ehrlich, G. D., Costerton, J. W., and Davies, D. G. (2002). *Pseudomonas aeruginosa* displays multiple phenotypes during development as a biofilm. *J. Bacteriol.* 184, 1140–1154. doi: 10.1128/jb.184.4.1140-1154.2002

Savini, A., and Savini, G. G. (2015). "A short history of 3D printing, a technological revolution just started," in *Proceedings of 2015 ICOHTEC/IEEE International History of High-Technologies and Their Socio-Cultural Contexts Conference (HISTELCON)*, (Tel-Aviv: IEEE), 1–8. doi: 10.1109/HISTELCON.2015.7307314

Schaerli, Y., Munteanu, A., Gili, M., Cotterell, J., Sharpe, J., and Isalan, M. (2014). A unified design space of synthetic stripe-forming networks. *Nat. Commun.* 5:4905. doi: 10.1038/ncomms5905

Schaffner, M., Rühs, P. A., Coulter, F., Kilcher, S., and Studart, A. R. (2017). 3D printing of bacteria into functional complex materials. *Sci. Adv.* 3:eaao6804. doi: 10.1126/sciadv.aao6804

Schink, B. (2002). Synergistic interactions in the microbial world. *Antonie Van Leeuwenhoek* 81, 257–261. doi: 10.1023/A:1020579004534

Schippers, K. J., and Nichols, S. A. (2014). Deep, dark secrets of melatonin in animal evolution. *Cell* 159, 9–10. doi: 10.1016/j.cell.2014.09.004

Schmieden, D. T., Basalo Vázquez, S. J., Sangüesa, H., van der Does, M., Idema, T., and Meyer, A. S. (2018). Printing of patterned, engineered *E. coli* biofilms with a low-cost 3D printer. *ACS Synth. Biol.* 7, 1328–1337. doi: 10.1021/acssynbio.7b00424

Scott, S. R., Din, M. O., Bittihn, P., Xiong, L., Tsimring, L. S., and Hasty, J. (2017). A stabilized microbial ecosystem of self-limiting bacteria using synthetic quorum-regulated lysis. *Nat. Microbiol.* 2:17083. doi: 10.1038/nmicrobiol.2017.83

Scott, S. R., and Hasty, J. (2016). Quorum sensing communication modules for microbial consortia. *ACS Synth. Biol.* 5, 969–977. doi: 10.1021/acssynbio.5b00286

Sekine, R., Yamamura, M., Ayukawa, S., Ishimatsu, K., Akama, S., Takinoue, M., et al. (2011). Tunable synthetic phenotypic diversification on Waddington's landscape through autonomous signaling. *Proc. Natl. Acad. Sci. U.S.A.* 108, 17969–17973. doi: 10.1073/pnas.1105901108

Shoaie, S., Ghaffari, P., Kovatcheva-Datchary, P., Mardinoglu, A., Sen, P., Pujos-Guillot, E., et al. (2015). Quantifying diet-induced metabolic changes of the human gut microbiome. *Cell Metab.* 22, 320–331. doi: 10.1016/j.cmet.2015.07.001

Siuti, P., Yazbek, J., and Lu, T. K. (2013). Synthetic circuits integrating logic and memory in living cells. *Nat. Biotechnol.* 31, 448–452. doi: 10.1038/nbt.2510

Slomovic, S., Pardee, K., and Collins, J. J. (2015). Synthetic biology devices for in vitro and in vivo diagnostics. *Proc. Natl. Acad. Sci. U.S.A.* 112, 14429–14435. doi: 10.1073/pnas.1508521112

Solé, R., Moses, M., and Forrest, S. (2019). Liquid brains, solid brains. *Philos. Trans. R. Soc. B Biol. Sci.* 374:20190040. doi: 10.1098/rstb.2019.0040

Solé, R. V., and Delgado, J. (1996). Universal computation in fluid neural networks. *Complexity* 2, 49–56. doi: 10.1002/(sici)1099-0526(199611/12)2:2<49::aid-cplx13>3.0.co;2-t

Solé, R. V., and Miramontes, O. (1995). Information at the edge of chaos in fluid neural networks. *Phys. D Nonlinear Phenom.* 80, 171–180. doi: 10.1016/0167-2789(95)90075-6

Stanton, B. C., Nielsen, A. A. K. K., Tamsir, A., Clancy, K., Peterson, T., and Voigt, C. A. (2014). Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nat. Chem. Biol.* 10, 99–105. doi: 10.1038/nchembio.1411

Steel, H., Habgood, R., and Papachristodoulou, A. (2019). Chi.Bio: an open-source automated experimental platform for biological science research. *bioRxiv*[Preprint]. doi: 10.1101/796516

Stein, R. R., Bucci, V., Toussaint, N. C., Buffie, C. G., Rätsch, G., Pamer, E. G., et al. (2013). Ecological modeling from time-series inference: insight into dynamics and stability of intestinal microbiota. *PLoS Comput. Biol.* 9:e1003388. doi: 10.1371/journal.pcbi.1003388

Stephens, K., Pozo, M., Tsao, C.-Y., Hauk, P., and Bentley, W. E. (2019). Bacterial co-culture with cell signaling translator and growth controller modules for autonomously regulated culture composition. *Nat. Commun.* 10:4129. doi: 10.1038/s41467-019-12027-6

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S., and Hasty, J. (2008). A fast, robust and tunable synthetic gene oscillator. *Nature* 456, 516–519. doi: 10.1038/nature07389

Strovas, T. J., Rosenberg, A. B., Kuypers, B. E., Muscat, R. A., and Seelig, G. (2014). MicroRNA-based single-gene circuits buffer protein synthesis rates against perturbations. *ACS Synth. Biol.* 3, 324–331. doi: 10.1021/sb4001867

Summers, D. (1991). The kinetics of plasmid loss. *Trends Biotechnol.* 9, 273–278. doi: 10.1016/0167-7799(91)90089-Z

Tabor, J. J., Salis, H. M., Simpson, Z. B., Chevalier, A. A., Levskaya, A., Marcotte, E. M., et al. (2009). A synthetic genetic edge detection program. *Cell* 137, 1272–1281. doi: 10.1016/j.cell.2009.04.048

Takahashi, C. N., Miller, A. W., Ekness, F., Dunham, M. J., and Klavins, E. (2015). A low cost, customizable turbidostat for use in synthetic circuit characterization. *ACS Synth. Biol.* 4, 32–38. doi: 10.1021/sb500165g

Taketani, M., Zhang, J., Zhang, S., Triassi, A. J., Huang, Y.-J., Griffith, L. G., et al. (2020). Genetic circuit design automation for the gut resident species *Bacteroides* thetaiotaomicron. *Nat. Biotechnol.* doi: 10.1038/s41587-020-0468-5 [Epub ahead of print].

Tamsir, A., Tabor, J. J., and Voigt, C. A. (2011). Robust multicellular computing using genetically encoded NOR gates and chemical 'wires.'. *Nature* 469, 212–215. doi: 10.1038/nature09565

Tatum, E. L., and Lederberg, J. (1947). Gene recombination in the bacterium *Escherichia coli*. *J. Bacteriol.* 53, 673–684. doi: 10.1128/JB.53.6.673-684.1947

Thommes, M., Wang, T., Zhao, Q., Paschalidis, I. C., and Segrè, D. (2019). Designing metabolic division of labor in microbial communities. *mSystems* 4:e00263-18. doi: 10.1128/msystems.00263-18

Thouless, R. H. (1953). "Pitfalls in analogy," in *Straight and Crooked Thinking*, ed. R. H. Thouless (London: Pan Books Ltd).

Toprak, E., Veres, A., Michel, J.-B., Chait, R., Hartl, D. L., and Kishony, R. (2012). Evolutionary paths to antibiotic resistance under dynamically sustained drug selection. *Nat. Genet.* 44, 101–105. doi: 10.1038/ng.1034

Treloar, N. J., Fedorec, A. J. H., Ingalls, B., and Barnes, C. P. (2020). Deep reinforcement learning for the control of microbial co-cultures in bioreactors. *PLoS Comput. Biol.* 16:e1007783. doi: 10.1371/journal.pcbi.1007783

Turing, A. M. (1937). On computable numbers, with an application to the entscheidungsproblem. *Proc. Lond. Math. Soc.* 42, 230–265. doi: 10.1112/plms/s2-42.1.230

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 237, 37–72. doi: 10.1098/rstb.1952.0012

Ullman, G., Wallden, M., Marklund, E. G., Mahmutovic, A., Razinkov, I., and Elf, J. (2013). High-throughput gene expression analysis at the level of single proteins using a microfluidic turbidostat and automated cell tracking. *Philos. Trans. R. Soc. B Biol. Sci.* 368:20120025. doi: 10.1098/rstb.2012.0025

Urrios, A., Macia, J., Manzoni, R., Conde, N., Bonforti, A., de Nadal, E., et al. (2016). A synthetic multicellular memory device. *ACS Synth. Biol.* 5, 862–873. doi: 10.1021/acssynbio.5b00252

van 't Riet, K., and van der Lans, R. G. J. M. (2011). "Mixing in bioreactor vessels," in *Comprehensive Biotechnology*, ed. M. Moo-Young (Burlington: Academic Press), 63–80. doi: 10.1016/B978-0-08-088504-9.00083-0

Villa Martín, P., Muñoz, M. A., and Pigolotti, S. (2019). Bet-hedging strategies in expanding populations. *PLoS Comput. Biol.* 15:e1006529. doi: 10.1371/journal.pcbi.1006529

Vining, W. F., Esponda, F., Moses, M. E., and Forrest, S. (2019). How does mobility help distributed systems compute? *Philos. Trans. R. Soc. B Biol. Sci.* 374:20180375. doi: 10.1098/rstb.2018.0375

Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature* 118, 558–560. doi: 10.1038/118558a0

von Neumann, J. (1993). First draft of a report on the EDVAC. *IEEE Ann. Hist. Comput.* 15, 27–75. doi: 10.1109/85.238389

Waddington, C. H. (1957). *The Strategy of the Genes*. London: Routledge, doi: 10.4324/9781315765471

Wang, J., Zhang, K., Xu, L., and Wang, E. (2011). Quantifying the Waddington landscape and biological paths for development and differentiation. *Proc. Natl. Acad. Sci. U.S.A.* 108, 8257–8262. doi: 10.1073/pnas.1017017108

Wang, Q., Gao, H., Alsaadi, F., and Hayat, T. (2014). An overview of consensus problems in constrained multi-agent coordination. *Syst. Sci. Control Eng.* 2, 275–284. doi: 10.1080/21642583.2014.897658

Weiße, A. Y., Oyarzún, D. A., Danos, V., and Swain, P. S. (2015). Mechanistic links between cellular trade-offs, gene expression, and growth. *Proc. Natl. Acad. Sci. U.S.A.* 112, E1038–E1047. doi: 10.1073/pnas.1416533112

Wilkinson, D. J. (2018). *Stochastic Modelling for Systems Biology*, 3rd Edn. Milton: Chapman and Hall, doi: 10.1201/9781420010664

Wolf, D. M., Vazirani, V. V., and Arkin, A. P. (2005). Diversity in times of adversity: probabilistic strategies in microbial survival games. *J. Theor. Biol.* 234, 227–253. doi: 10.1016/j.jtbi.2004.11.020

Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.* 25, 1–47. doi: 10.1016/S0022-5193(69)80016-0

Wong, B. G., Mancuso, C. P., Kiriakov, S., Bashor, C. J., and Khalil, A. S. (2018). Precise, automated control of conditions for high-throughput growth of yeast and bacteria with eVOLVER. *Nat. Biotechnol.* 36, 614–623. doi: 10.1038/nbt.4151

Woods, M. L., Leon, M., Perez-Carrasco, R., and Barnes, C. P. (2016). A statistical approach reveals designs for the most robust stochastic gene oscillators. *ACS Synth. Biol.* 5, 459–470. doi: 10.1021/acssynbio.5b00179

Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R., and Benenson, Y. (2011). Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science* 333, 1307–1311. doi: 10.1126/science.1205527

Xiong, L., Cao, Y., Cooper, R., Rappel, W.-J., Hasty, J., and Tsimring, L. (2020). Flower-like patterns in multi-species bacterial colonies. *eLife* 9, 1–27. doi: 10.7554/eLife.48885

Yu, S. R., Burkhardt, M., Nowak, M., Ries, J., Petrášek, Z., Scholpp, S., et al. (2009). Fgf8 morphogen gradient forms by a source-sink mechanism with freely diffusing molecules. *Nature* 461, 533–536. doi: 10.1038/nature08391

Yurtsev, E. A., Conwill, A., and Gore, J. (2016). Oscillatory dynamics in a bacterial cross-protection mutualism. *Proc. Natl. Acad. Sci. U.S.A.* 113, 6236–6241. doi: 10.1073/pnas.1523317113

Zelezniak, A., Andrejev, S., Ponomarova, O., Mende, D. R., Bork, P., and Patil, K. R. (2015). Metabolic dependencies drive species co-occurrence in diverse microbial communities. *Proc. Natl. Acad. Sci. U.S.A.* 112, 6449–6454. doi: 10.1073/pnas.1421834112

Zhang, S., and Voigt, C. A. (2018). Engineered dCas9 with reduced toxicity in bacteria: implications for genetic circuit design. *Nucleic Acids Res.* 46, 11115–11125. doi: 10.1093/nar/gky884

Zhang, X., and Reed, J. L. (2014). Adaptive evolution of synthetic cooperating communities improves growth performance. *PLoS One* 9:e108297. doi: 10.1371/journal.pone.0108297

Zhou, K., Qiao, K., Edgar, S., and Stephanopoulos, G. (2015). Distributing a metabolic pathway among a microbial consortium enhances production of natural products. *Nat. Biotechnol.* 33, 377–383. doi: 10.1038/nbt.3095

Zhu, X., Campanaro, S., Treu, L., Seshadri, R., Ivanova, N., Kougias, P. G., et al. (2020). Metabolic dependencies govern microbial syntrophies during methanogenesis in an anaerobic digestion ecosystem. *Microbiome* 8:22. doi: 10.1186/s40168-019-0780-9

Ziesack, M., Gibson, T., Oliver, J. K. W., Shumaker, A. M., Hsu, B. B., Riglar, D. T., et al. (2019). Engineered interspecies amino acid cross-feeding increases population evenness in a synthetic bacterial consortium. *mSystems* 4:e00352-19. doi: 10.1128/mSystems.00352-19

Zinovyev, A., Calzone, L., Fourquet, S., and Barillot, E. (2013). "How cell decides between life and death: mathematical modeling of epigenetic landscapes of cellular fates," in *Pattern Formation in Morphogenesis. Springer Proceedings in Mathematics*, Vol. 15, eds V. Capasso, M. Gromov, A. Harel-Bellan, N. Morozova, and L. Pritchard (Berlin: Springer), 191–204. doi: 10.1007/978-3-642-20164-6_16

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

PERSPECTIVE
published: 24 July 2020
doi: 10.3389/fbioe.2020.00788

Cell-Free Systems: A Proving Ground for Rational Biodesign

*Nadanai Laohakunakorn**

School of Biological Sciences, Institute of Quantitative Biology, Biochemistry, and Biotechnology, University of Edinburgh, Edinburgh, United Kingdom

Cell-free gene expression systems present an alternative approach to synthetic biology, where biological gene expression is harnessed inside non-living, *in vitro* biochemical reactions. Taking advantage of a plethora of recent experimental innovations, they easily overcome certain challenges for computer-aided biological design. For instance, their open nature renders all their components directly accessible, greatly facilitating model construction and validation. At the same time, these systems present their own unique difficulties, such as limited reaction lifetimes and lack of homeostasis. In this Perspective, I propose that cell-free systems are an ideal proving ground to test rational biodesign strategies, as demonstrated by a small but growing number of examples of model-guided, forward engineered cell-free biosystems. It is likely that advances gained from this approach will contribute to our efforts to more reliably and systematically engineer both cell-free as well as living cellular systems for useful applications.

Keywords: cell-free synthetic biology, cell-free protein synthesis, *in vitro* transcription translation, model-guided design, rational design

OPEN ACCESS

Edited by:
Thomas Edward Gorochowski, University of Bristol, United Kingdom

Reviewed by:
Manish Kushwaha, INRA UMR1319 Microbiologie de l'Alimentation au Service de la Santé, France
Borkowski Olivier, Institut Pasteur, France

***Correspondence:**
Nadanai Laohakunakorn
nadanai.laohakunakorn@ed.ac.uk

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *30 April 2020*
Accepted: *22 June 2020*
Published: *24 July 2020*

Citation:
Laohakunakorn N (2020) Cell-Free Systems: A Proving Ground for Rational Biodesign. Front. Bioeng. Biotechnol. 8:788. doi: 10.3389/fbioe.2020.00788

1. INTRODUCTION

A basic aim of synthetic biology is to design and construct biological systems which perform a given function. An extension of this, inspired by common engineering practice, is to additionally demand that the systems perform robustly, predictably, and with quantitative precision. Some practitioners of synthetic biology explicitly adopt the conventional engineering approach of rational design, where a system is constructed predictively (Endy, 2005; Heinemann and Panke, 2006). In contrast to non-biological engineering, synthetic biological systems are also open to the possibility of evolutionary design (Arnold, 1998), where function is obtained through directed evolutionary screens. It is still an open question as to whether or not a purely rational engineering approach can ultimately be successfully applied to engineer complex biomolecular systems (Davies, 2019).

A fully rational approach adopts all conventional engineering principles, such as standardization and quantitative characterization of parts, mathematical models to describe their behavior, and abstraction which allows hierarchical assembly of parts into modules, subsystems, and systems (Endy, 2005; Arkin, 2008; Canton et al., 2008). For any system of non-trivial complexity, this approach relies on computational methods to enable predictive design (MacDonald et al., 2011).

To a large extent, strict adherence to this approach has not yet been widely successful in synthetic biology, with a few notable exceptions (e.g., Nielsen et al., 2016). Typically, biodesign involves multiple iterations through a so-called "design-build-test-learn" (DBTL) cycle. While eventually a functioning system is produced, the path to get there is not directly through predictive design, but rather informed trial-and-error. The current necessity of DBTL cycles is due to partly to the fact that the complexity associated with biomolecular systems eludes simplifying black-box approximations common in other physical scenarios. Additionally, interrogating biosystems with controlled inputs

and perturbations is difficult, and system parameters can be context-dependent and varying. However, with the emergence of high-throughput automation and biofoundries (Hillson et al., 2019), the promise is that DBTL efforts will ultimately enable fully predictive, rational biodesign.

Cell-free systems (Garenne and Noireaux, 2019) can contribute in several ways to improve the design process of synthetic biological systems, which span scales from the molecular (genetic regulatory elements, proteins, enzymes), to the systemic (gene regulatory and metabolic networks), and all the way to the extracellular levels (synthetic cells, communication, self-assembly). First, they can accelerate DBTL cycles through rapid prototyping (Chappell et al., 2013; Niederholtmeyer et al., 2015; Takahashi et al., 2015). Second, they can be used efficiently for *in vitro* directed evolution (Contreras-Llano and Tan, 2018). In this Perspective I would like to focus on a third contribution, and suggest that they offer an ideal proving ground to test the approach of rational computer-aided biodesign as applied to biomolecular systems (**Figure 1**). In particular, they present features which overcome some of the difficulties associated with engineering living cells, and so can be used to more easily develop and calibrate mechanistic models, as well as generate sufficient data for machine learning approaches.

To understand their strengths and weaknesses in the context of synthetic biology, it is first important to consider the differences between cell-free and living cellular systems.

2. BIOPHYSICAL DIFFERENCES BETWEEN CELL-FREE AND CELLULAR SYSTEMS

Today, cell-free technology generally refers to cell-free protein synthesis (CFPS), which rests on the foundational processes of *in vitro* transcription and translation (Silverman et al., 2019a; Laohakunakorn et al., 2020). Strictly speaking, CFPS belongs to the much broader field of *in vitro* reconstitution, which consists of recapitulating biological processes outside of the living cell. This involves combining relevant enzymes (either purified or extracted in crude cellular lysate) with a reaction mixture containing substrates, cofactors, and specific ionic and pH conditions. Constructing such a reaction isolates specific biological processes, and has historically served as a key approach to elucidate molecular biological mechanisms, including deciphering the genetic code itself (Nirenberg and Matthaei, 1961; Zubay, 1973). While this article will focus predominantly on bacterial cell-free systems due to their current widespread use, cell-free systems have also been successfully prepared from a number of prokaryotic and eukaryotic organisms (Perez et al., 2016).

In addition to developing fundamental understanding, this approach also enables technological applications: examples of these are the use of CFPS to carry out *in vitro* biomanufacturing, where the production of exogenous protein is advantageously decoupled from cellular growth (Karim and Jewett, 2018; Gregorio et al., 2019); and biosensing, where robust, lyophilized cell-free gene circuits can be activated and used to detect environmental contaminants and pathogens directly in the field (Pardee et al., 2014).

One predominant viewpoint of cell-free systems is that they are cellular mimics. The crude cellular lysate is a representation of the cellular cytosol, and contains, in addition to transcription and translation, a number of intact and functional core metabolic pathways (Kim and Swartz, 2001; Kim and Kim, 2009). Thus, cell-free systems have been successfully used as a prototyping platform for synthetic biology, a so-called "cellular breadboard"

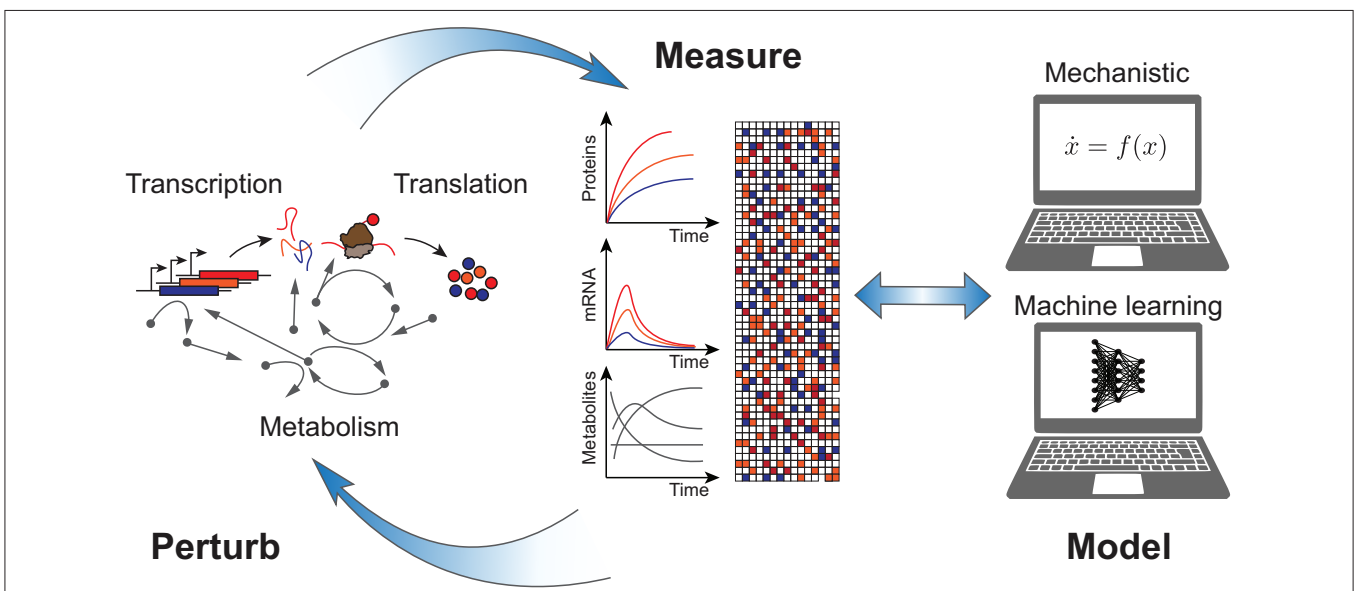


FIGURE 1 | Rational computer-aided biodesign with cell-free systems: cell-free gene expression systems perform *in vitro* transcription, translation, and metabolism in reactions which are open and transparent to measurement and perturbation. Data sets generated are kinetically resolved and high-throughput, facilitating development of both mechanistic as well as machine learning models.

where synthetic gene circuit designs can be quickly implemented, validated, and ported back into a living cell (Siegal-Gaskins et al., 2014; Garamella et al., 2016). The success of this approach relies on a basic similarity between the cell-free and cellular environments, an assumption that has been verified in a number of notable examples (Chappell et al., 2013; Niederholtmeyer et al., 2015; Borkowski et al., 2018; Halleran and Murray, 2018; Hu et al., 2018).

On the other hand, cell-free systems do contain fundamental differences from cells. In addition to being non-living, there are a number of key biophysical differences. Below I highlight these, and consider their consequences in the context of the implementation of a generic synthetic gene regulatory network (GRN). In some cases, strategies for making the system more "lifelike" by bottom-up construction are briefly discussed.

1. **Accessible system:** Without a barrier between the reaction and the environment, the cell-free reaction is transparent to observation and perturbation, allowing the reaction conditions to be adjusted at will. This property can be leveraged, for example, to change redox environments to promote disulphide bond formation (Matsuda et al., 2013). The kinetic progress of reactions can be followed using fluorescence from proteins and mRNA, as well as real-time metabolomic profiling, which has allowed the internal metabolism of cell-free systems to be dissected at high resolution (Bujara et al., 2011; Vilkhovoy et al., 2019). For GRN design, parameters, such as dissociation and kinetic constants between a transcription factor and promoter may be measured *in situ* (Geertz et al., 2012; Swank et al., 2019), and perturbations applied to the reaction composition to facilitate parameter identification and model selection (Hu et al., 2015; Moore et al., 2018). These key properties of controllable inputs, perturbations, and consistency between conditions where parameter measurement and system operation take place directly address challenges faced in engineering living cells. Crucially, this enables a close coupling of cell-free experiments and computational models.
2. **Dilute, well-mixed reaction environment:** The lack of compartmentalization is also related to a number of other physical effects, including a loss of stochasticity, slower enzymatic rates, a reduced level of macromolecular crowding, a loss of spatial organization, and a loss of membrane-associated processes [although lysates can contain inverted membrane vesicles which permit oxidative phosphorylation (Jewett et al., 2008)]. A useful consequence of such a simplified reaction environment is that the system can be described with deterministic kinetics; a practical side-effect is that exogenous protein aggregation is minimized, which facilitates bioproduction. In order to recreate more lifelike reaction environments, there is much ongoing effort to encapsulate cell-free reactions in a variety of compartments including liposomes, polymersomes, and droplets, as well as introducing crowding and organization into cell-free systems (Laohakunakorn et al., 2020).
3. **Relaxation to equilibrium:** Living cells are maintained in a homeostatic, non-equilibrium steady state by a constant flux of energy and metabolites through the system, while cell-free reactions relax to biochemical equilibrium as the reaction proceeds. This sets a limit on the lifetime of the cell-free reaction. The lifetime may be extended by engineering a more homeostatic metabolic system [for instance, rationally-designed *in vitro* metabolic systems can operate autonomously for days (Korman et al., 2017)], but ultimately to maintain cell-free systems in a steady state, an energy and metabolite flux must be set up between the system and environment. This can be achieved using continuous flow or continuous exchange reactors (Spirin et al., 1988; Niederholtmeyer et al., 2013; Karzbrun et al., 2014), or by compartmentalizing and coupling the reaction to transport processes (Noireaux and Libchaber, 2004). A consequence of limited lifetime is that after a few hours, any synthetic cell-free gene circuit ceases to be functional. Thus, recent efforts have focused on extending reaction lifetimes (Caschera and Noireaux, 2015) as well as accelerating the computational and output steps in the circuit (Alam et al., 2019).
4. **No regulation:** While living cells are actively regulated at multiple levels of organization, from molecular- to network-scale, cell-free systems contain no active regulatory mechanisms. This simplifies the identification and measurement of host-chassis interactions, allowing resource allocation within the cell-free system to be elucidated in detail (Siegal-Gaskins et al., 2014; Gyorgy and Murray, 2016; Borkowski et al., 2018; Halter et al., 2018). On the other hand, cell-free systems lose the robustness conferred by homeostasis (Lewis et al., 2014). They are thus sensitive to effects which would otherwise be regulated, for example partial degradation products (Kim and Winfree, 2011), stochasticity in gene expression (Karig et al., 2013), and variable partitioning of reactants during system encapsulation (Altamura et al., 2018). This property may thus be an impediment to predictive design.
5. **No self-regeneration:** Self-regeneration is a defining hallmark of life (Luisi et al., 2006), and cell-free systems do not regenerate their components, implying that the lifetime of cell-free reactions is also limited by enzyme stability (Stögbauer et al., 2012), in addition to resource depletion and metabolic arrest. The possibility of programming regeneration directly in the cell-free system leads to the tantalizing prospect of a cell-free system capable of maintaining its components, which could form the basis of an engine to power artificial cells (Schwille et al., 2018).
6. **No replication:** In addition to not regenerating their components, cell-free systems also do not replicate their genetic material. This has been considered an opportunity for bottom-up reconstruction, from early demonstrations of *in vitro* replication of plasmids and viral DNA in prokaryotic and eukaryotic lysates (Diaz and Staudenbauer, 1982; Li and Kelly, 1984; Stillman and Gluzman, 1985) to more recent studies involving phi29 DNA polymerase (Sakatani et al., 2015; van Nies et al., 2018), which have culminated in the replication of up to 116 kb of DNA in the PURE system (Libicher et al., 2020). Lack of replication implies genetic stability of introduced DNA, unlike living cells which can

mutate away exogenous gene circuit function. Steady-state *in vitro* replication of nucleic acids would form a necessary subsystem of self-replicating artificial cells as well as enable *in vitro* evolutionary studies (Meyer et al., 2012).

These properties have influenced the approaches used in cell-free engineering. In particular, the accessibility of the reaction environment has made cell-free systems particularly suited for rational biodesign strategies, as will be discussed next.

3. RATIONAL BIODESIGN STRATEGIES FOR CELL-FREE SYNTHETIC BIOLOGY

3.1. Model-Guided Design

The most ambitious approach to rational biodesign uses a quantitative and predictive model to guide the design process, adopting workflows from well-established fields, such as electrical and aerospace engineering. For synthetic biology, the largest obstacles to this involve unknown, uncharacterized, or changing interactions among biomolecular components, and the difficulty of accessing and perturbing system components. In general, we can envisage two broad approaches which aim to mitigate this knowledge gap in cell-free systems: a "bottom-up" approach, where purified, reconstituted systems are constructed one component at a time, allowing interactions to be taken into account as they arise; and a "top-down" approach, where crude cellular lysates are interrogated and potentially modified to remove unwanted interactions, exposing the minimal system beneath. These approaches mirror the bottom-up and top-down approaches to the construction of artificial cells, with the final result being a minimal system that is maximally understood.

Recently, efforts have been made to combine the development of reconstituted cell-free systems with mathematical modeling (Mavelli et al., 2015; Matsuura et al., 2017, 2018; Carrara et al., 2018; Doerr et al., 2019). Reconstituted systems are composed of purified cellular enzymes and an energy solution, mixed together in a known composition, and are available commercially [e.g., commonly-used variants based on the PURE system (Shimizu et al., 2001)]. Compared to lysates, reconstituted systems are dramatically simplified. In principle, since the exact system composition is known, a model incorporating all predicted interactions can be written down. In practice, it is infeasible to calibrate such fine-grained models to experimental measurements, although properties, such as robustness of the system can be investigated *in silico* (Matsuura et al., 2017). Current coarse-grained models are generally not sufficient to globally capture all observed experimental effects (Doerr et al., 2019). The overarching aim is therefore to search for computational models of appropriate granularity which can describe all experimental observations, and yet remain feasible for calibration. The success of this is likely to be borne out through approaches which combine automation and high-throughput measurements with improved cost-efficient methods for preparing recombinant systems (Lavickova and Maerkl, 2019).

The top-down, systems-level approach aims to develop mechanistic understanding by interrogating lysates, which contain significantly more unknowns. The 'black-box' of lysates has slowly been opened over the last two decades, motivated by a desire to improve productivity and lifetime of the system (Silverman et al., 2019b). Using a combination of strain engineering and data from biochemical and metabolic analyses, it is now possible to rationally redirect metabolic flux and energy usage. Energy regeneration schemes of increasing complexity have been developed in order to improve lysate reaction lifetime and yield, proceeding initially from single-step (Zubay, 1973; Kigawa et al., 1999) to multi-step pathways which regenerate ATP using enzymes present within the extract (Kim and Swartz, 1999, 2001; Jewett and Swartz, 2004; Sitaraman et al., 2004; Calhoun and Swartz, 2005; Jewett et al., 2008; Caschera and Noireaux, 2015). While the complexity of lysates is considerable, in contrast to cellular systems biology, cell-free systems are amenable to essentially unconstrained perturbation, which greatly facilitates model testing and validation. This has been demonstrated by a number of modeling studies of increasing sophistication (Karzbrun et al., 2011; Stögbauer et al., 2012; Tuza et al., 2015; Gyorgy and Murray, 2016; Nieß et al., 2017; Marshall and Noireaux, 2019), as well as notable examples of model-guided forward engineering of genetic circuits (Hu et al., 2015, 2018; Agrawal et al., 2019; Lehr et al., 2019; Westbrook et al., 2019). Recent development of integrated gene expression and metabolic models have elucidated the factors limiting CFPS (Wayman et al., 2015; Vilkhovoy et al., 2018, 2019; Horvath et al., 2020), suggesting that combined computational and experimental metabolomic studies are poised to contribute significantly to our understanding of CFPS in lysates.

3.2. Control Theoretic Approaches for Robust Operation

Even if all interactions could be measured, many are unlikely to remain constant with time. Additionally, cell-free reactions operate dynamically, in an equally dynamic, fluctuating environment. It is clear, then, that knowledge of interactions is not generally sufficient to ensure robust performance.

Control engineering attempts to maintain the performance of a dynamic system within certain specified bounds, while parts of the system are subject to uncontrolled disturbances. A specific example of this is reference tracking using feedback control, where an output, such as the gene expression level follows a reference signal despite the presence of perturbations. Achieving this requires the system to sense the reference as well as its output, which involves redirecting the output back into the system in a feedback loop. Another example is buffering outputs against upstream variability using feed-forward regulators which balance each other's activity. Feedback and feedforward loops are ubiquitous in natural biological systems, making it a natural extension to develop synthetic biology within a control theoretic framework (Vecchio et al., 2016; Del Vecchio et al., 2018; Hsiao et al., 2018; Baetica et al., 2019).

Feedback control has been successfully implemented in a number of *in vivo* examples, for instance to regulate exogenous gene expression in response to burden (Ceroni et al., 2018), or to control growth rate using robust perfect adaptation (Aoki

et al., 2019). Feedforward architectures have also been used to control variations in the amount of DNA template present in cells (Bleris et al., 2011). In the context of control circuits, a significant advantage of cell-free over cellular systems is that they are free from biological noise, and operate in the deterministic regime. Despite these benefits, surprisingly little work has been carried out on cell-free control; recent examples, including a computational study (Agrawal et al., 2018) and an experimental demonstration of a feedback integral controller based on molecular sequestration (Agrawal et al., 2019), as well as feedforward loop circuits (Guo and Murray, 2019), suggest that such approaches are starting to become more widespread within the cell-free community.

3.3. Active Learning

While a mechanistic or phenomenological model may lead to transparent understanding of the system, they are not the only models offering sufficient predictive capabilities for rational design. Statistical or non-parametric models, developed from data using machine learning approaches, can be equally or more strongly predictive, albeit with the well-known challenges of interpretability (Doshi-Velez and Kim, 2017).

An example problem is to determine the composition of a cell-free reaction to maximize its protein productivity. In the absence of a predictive model fully connecting all its components to protein output, Caschera et al. (2018) used an evolutionary design of experiments approach to iteratively train an ensemble neural network model to optimize conditions for cell-free protein synthesis. More recently Borkowski et al. (2020) trained a similar model on ∼4,000 reactions, improving yields by 34 times. Importantly, they discovered a training dataset of only 20 compositions which was informative enough to allow the model to generalize its predictions to different lysates and conditions. These examples demonstrate that the throughput afforded by cell-free systems is sufficient for informing data-driven modeling approaches.

Data-driven cell-free techniques could also potentially be applied to other long-standing questions in systems biology, for instance, determining the mapping of sequence to phenotype for a genetic element (Cuperus et al., 2017; Sample et al., 2019). Cell-free implementations of massively parallel reporter assays, perhaps using droplet microfluidic technology to maintain genotype-phenotype linkage, may yield datasets of sufficient quality and size to contribute to this problem.

4. CONCLUSIONS

Cell-free systems are ideally suited for rational engineering approaches: their open reactions facilitate construction and validation of mechanistic and phenomenological models, and their throughput allows them to generate sufficient data to train machine learning models. Developing these models, and designing experiments to calibrate and validate them, are general strategies which can be tested on the cell-free platform, but eventually also applied to the more challenging problem of engineering living systems. In this sense, cell-free systems can be thought of as a proving ground for rational design strategies.

Cell-free systems do however present unique challenges for predictive design. As discussed above, a lack of homeostasis can imply ultrasensitivity of cell-free reactions to various effects. It is also well-known that strong batch-to-batch variation of lysates can limit the predictability of results to within-batch repeats, constraining the usefulness of the approach; however recent efforts have been made to identify and control these effects (Cole et al., 2019; Silverman et al., 2019b). And finally, while examples were given of successful transfer of cell-free designs back into cellular hosts, the generality of this approach has so far remained unclear. These are all avenues for future research within the field.

In this Perspective, I have deliberately left out a discussion of directed evolution, a complementary and powerful strategy for biodesign. Cell-free systems have also been extensively deployed for *in vitro* evolution (Contreras-Llano and Tan, 2018), maintaining genotype-phenotype coupling through the use of display technologies and compartmentalization.

Reliable engineering of synthetic biological systems remains a great challenge, and it is likely that a number of complementary efforts including rational as well as evolutionary design, and cellular and cell-free systems, will be required to eventually achieve this grand goal.

AUTHOR CONTRIBUTIONS

NL conceived and wrote the article.

FUNDING

NL was supported by a Chancellor's Fellowship from the University of Edinburgh.

REFERENCES

Agrawal, D. K., Marshall, R., Noireaux, V., and Sontag, E. D. (2019). *In vitro* implementation of robust gene regulation in a synthetic biomolecular integral controller. *Nat. Commun.* 10, 1–12. doi: 10.1038/s41467-019-13626-z

Agrawal, D. K., Tang, X., Westbrook, A., Marshall, R., Maxwell, C. S., Lucks, J., et al. (2018). Mathematical modeling of RNA-based architectures for closed loop control of gene expression. *ACS Synth. Biol.* 7, 1219–1228. doi: 10.1021/acssynbio.8b00040

Alam, K. K., Jung, J. K., Verosloff, M. S., Clauer, P. R., Lee, J. W., Capdevila, D. A., et al. (2019). Rapid, low-cost detection of water contaminants using regulated *in vitro* transcription. *biorXiv*. doi: 10.1101/619296

Altamura, E., Carrara, P., D'Angelo, F., Mavelli, F., and Stano, P. (2018). Extrinsic stochastic factors (solute partition) in gene expression inside lipid vesicles and lipid-stabilized water-in-oil droplets: a review. *Synth. Biol.* 3, 1–16. doi: 10.1093/synbio/ysy011

Aoki, S. K., Lillacci, G., Gupta, A., Baumschlager, A., Schweingruber, D., and Khammash, M. (2019). A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* 570, 533–537. doi: 10.1038/s41586-019-1321-1

Arkin, A. (2008). Setting the standard in synthetic biology. *Nat. Biotechnol.* 26, 771–774. doi: 10.1038/nbt0708-771

Arnold, F. H. (1998). Design by directed evolution. *Acc. Chem. Res.* 31, 125–131. doi: 10.1021/ar960017f

Baetica, A.-A., Westbrook, A., and El-Samad, H. (2019). Control theoretical concepts for synthetic and systems biology. *Curr. Opin. Syst. Biol.* 14, 50–57. doi: 10.1016/j.coisb.2019.02.010

Bleris, L., Xie, Z., Glass, D., Adadey, A., Sontag, E., and Benenson, Y. (2011). Synthetic incoherent feedforward circuits show adaptation to the amount of their genetic template. *Mol. Syst. Biol.* 7, 1–12. doi: 10.1038/msb.2011.49

Borkowski, O., Bricio, C., Murgiano, M., Rothschild-Mancinelli, B., Stan, G.-B., and Ellis, T. (2018). Cell-free prediction of protein expression costs for growing cells. *Nat. Commun.* 9, 1–11. doi: 10.1038/s41467-018-03970-x

Borkowski, O., Koch, M., Zettor, A., Pandi, A., Batista, A. C., Soudier, P., et al. (2020). Large scale active-learning-guided exploration for *in vitro* protein production optimization. *Nat. Commun.* 11, 1–8. doi: 10.1038/s41467-020-15798-5

Bujara, M., Schümperli, M., Pellaux, R., Heinemann, M., and Panke, S. (2011). Optimization of a blueprint for *in vitro* glycolysis by metabolic real-time analysis. *Nat. Chem. Biol.* 7, 271–277. doi: 10.1038/nchembio.541

Calhoun, K. A., and Swartz, J. R. (2005). Energizing cell-free protein synthesis with glucose metabolism. *Biotechnol. Bioeng.* 90, 606–613. doi: 10.1002/bit.20449

Canton, B., Labno, A., and Endy, D. (2008). Refinement and standardization of synthetic biological parts and devices. *Nat. Biotechnol.* 26, 787–793. doi: 10.1038/nbt1413

Carrara, P., Altamura, E., D'Angelo, F., Mavelli, F., and Stano, P. (2018). Measurement and numerical modeling of cell-free protein synthesis: combinatorial block-variants of the PURE system. *Data* 3, 1–12. doi: 10.3390/data3040041

Caschera, F., Karim, A. S., Gazzola, G., D'Aquino, A. E., Packard, N. H., and Jewett, M. C. (2018). High-throughput optimization cycle of a cell-free ribosome assembly and protein synthesis system. *ACS Synth. Biol.* 7, 2841–2853. doi: 10.1021/acssynbio.8b00276

Caschera, F., and Noireaux, V. (2015). Preparation of amino acid mixtures for cell-free expression systems. *Biotechniques* 58, 40–42. doi: 10.2144/000114249

Ceroni, F., Boo, A., Furini, S., Gorochowski, T. E., Borkowski, O., Ladak, Y. N., et al. (2018). Burden-driven feedback control of gene expression. *Nat. Methods* 15, 387–393. doi: 10.1038/nmeth.4635

Chappell, J., Jensen, K., and Freemont, P. S. (2013). Validation of an entirely *in vitro* approach for rapid prototyping of DNA regulatory elements for synthetic biology. *Nucleic Acids Res.* 41, 3471–3481. doi: 10.1093/nar/gkt052

Cole, S. D., Beabout, K., Turner, K. B., Smith, Z. K., Funk, V. L., Harbaugh, S. V., et al. (2019). Quantification of interlaboratory cell-free protein synthesis variability. *ACS Synth. Biol.* 8, 2080–2091. doi: 10.1021/acssynbio.9b00178

Contreras-Llano, L. E., and Tan, C. (2018). High-throughput screening of biomolecules using cell-free gene expression systems. *Synth. Biol.* 3, 1–13. doi: 10.1093/synbio/ysy012

Cuperus, J. T., Groves, B., Kuchina, A., Rosenberg, A. B., Jojic, N., Fields, S., et al. (2017). Deep learning of the regulatory grammar of yeast 5' untranslated regions from 500,000 random sequences. *Genome Res.* 27, 2015–2024. doi: 10.1101/gr.224964.117

Davies, J. A. (2019). Real-world synthetic biology: is it founded on an engineering approach, and should it be? *Life* 9, 1–15. doi: 10.3390/life9010006

Del Vecchio, D., Qian, Y., Murray, R. M., and Sontag, E. D. (2018). Future systems and control research in synthetic biology. *Annu. Rev. Control* 45, 5–17. doi: 10.1016/j.arcontrol.2018.04.007

Diaz, R., and Staudenbauer, W. L. (1982). Replication of the broad host range plasmid RSF1010 in cell-free extracts of *Escherichia coli* and *Pseudomonas aeruginosa*. *Nucleic Acids Res.* 10, 4687–4702. doi: 10.1093/nar/10.15.4687

Doerr, A., de Reus, E., van Nies, P., van der Haar, M., Wei, K., Kattan, J., et al. (2019). Modelling cell-free RNA and protein synthesis with minimal systems. *Phys. Biol.* 16:025001. doi: 10.1088/1478-3975/aaf33d

Doshi-Velez, F., and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv [Preprint] arXiv:*1702.08608.

Endy, D. (2005). Foundations for engineering biology. *Nature* 438, 449–453. doi: 10.1038/nature04342

Garamella, J., Marshall, R., Rustad, M., and Noireaux, V. (2016). The all *E. coli* TX-TL toolbox 2.0: a platform for cell-free synthetic biology. *ACS Synth. Biol.* 5, 344–355. doi: 10.1021/acssynbio.5b00296

Garenne, D., and Noireaux, V. (2019). Cell-free transcription–translation: engineering biology from the nanometer to the millimeter scale. *Curr. Opin. Biotechnol.* 58, 19–27. doi: 10.1016/j.copbio.2018.10.007

Geertz, M., Shore, D., and Maerkl, S. J. (2012). Massively parallel measurements of molecular interaction kinetics on a microfluidic platform. *Proc. Natl. Acad. Sci. U.S.A.* 109, 16540–16545. doi: 10.1073/pnas.1206011109

Gregorio, N. E., Levine, M. Z., Oza, J. P., Gregorio, N. E., Levine, M. Z., and Oza, J. P. (2019). A user's guide to cell-free protein synthesis. *Methods Protoc.* 2:24. doi: 10.3390/mps2010024

Guo, S., and Murray, R. M. (2019). Construction of incoherent feedforward loop circuits in a cell-free system and in cells. *ACS Synth. Biol.* 8, 606–610. doi: 10.1021/acssynbio.8b00493

Gyorgy, A., and Murray, R. M. (2016). Quantifying resource competition and its effects in the TX-TL system, in *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, Vol. 1 (Las Vegas, NV), 3363–3368. doi: 10.1109/CDC.2016.7798775

Halleran, A. D., and Murray, R. M. (2018). Cell-free and *in vivo* characterization of Lux, Las, and Rpa quorum activation systems in *E. coli*. *ACS Synth. Biol.* 7, 752–755. doi: 10.1021/acssynbio.7b00376

Halter, W., Allgower, F., Murray, R. M., and Gyorgy, A. (2018). Optimal experiment design and leveraging competition for shared resources in cell-free extracts, in *2018 IEEE Conference on Decision and Control (CDC)* (Miami Beach, FL: IEEE), 1872–1879. doi: 10.1109/CDC.2018.8619039

Heinemann, M., and Panke, S. (2006). Synthetic biology–putting engineering into biology. *Bioinformatics* 22, 2790–2799. doi: 10.1093/bioinformatics/btl469

Hillson, N., Caddick, M., Cai, Y., Carrasco, J. A., Chang, M. W., Curach, N. C., et al. (2019). Building a global alliance of biofoundries. *Nat. Commun.* 10, 1–4. doi: 10.1038/s41467-019-10079-2

Horvath, N., Vilkhovoy, M., Wayman, J. A., Calhoun, K., Swartz, J., and Varner, J. D. (2020). Toward a genome scale sequence specific dynamic model of cell-free protein synthesis in *Escherichia coli*. *Metab. Eng. Commun.* 10:e00113. doi: 10.1016/j.mec.2019.e00113

Hsiao, V., Swaminathan, A., and Murray, R. M. (2018). Control theory for synthetic biology. *IEEE Control Syst.* 38, 32–62. doi: 10.1109/MCS.2018.2810459

Hu, C. Y., Takahashi, M. K., Zhang, Y., and Lucks, J. B. (2018). Engineering a functional small RNA negative autoregulation network with model-guided design. *ACS Synth. Biol.* 7, 1507–1518. doi: 10.1021/acssynbio.7b00440

Hu, C. Y., Varner, J. D., and Lucks, J. B. (2015). Generating effective models and parameters for RNA genetic circuits. *ACS Synth. Biol.* 4, 914–926. doi: 10.1021/acssynbio.5b00077

Jewett, M. C., Calhoun, K. A., Voloshin, A., Wuu, J. J., and Swartz, J. R. (2008). An integrated cell-free metabolic platform for protein production and synthetic biology. *Mol. Syst. Biol.* 4, 1–10. doi: 10.1038/msb.2008.57

Jewett, M. C., and Swartz, J. R. (2004). Mimicking the *Escherichia coli* cytoplasmic environment activates long-lived and efficient cell-free protein synthesis. *Biotechnol. Bioeng.* 86, 19–26. doi: 10.1002/bit.20026

Karig, D. K., Jung, S. Y., Srijanto, B., Collier, C. P., and Simpson, M. L. (2013). Probing cell-free gene expression noise in femtoliter volumes. *ACS Synth. Biol.* 2, 497–505. doi: 10.1021/sb400028c

Karim, A. S., and Jewett, M. C. (2018). Cell-free synthetic biology for pathway prototyping. *Methods Enzymol.* 608, 31–57. doi: 10.1016/bs.mie.2018.04.029

Karzbrun, E., Shin, J., Bar-Ziv, R. H., and Noireaux, V. (2011). Coarse-grained dynamics of protein synthesis in a cell-free system. *Phys. Rev. Lett.* 106, 1–4. doi: 10.1103/PhysRevLett.106.048104

Karzbrun, E., Tayar, A. M., Noireaux, V., and Bar-Ziv, R. H. (2014). Programmable on-chip DNA compartments as artificial cells. *Science* 345, 829–832. doi: 10.1126/science.1255550

Kigawa, T., Yabuki, T., Yoshida, Y., Tsutsui, M., Ito, Y., Shibata, T., et al. (1999). Cell-free production and stable-isotope labeling of milligram quantities of proteins. *FEBS Lett.* 442, 15–19. doi: 10.1016/S0014-5793(98)01620-2

Kim, D. M., and Swartz, J. R. (1999). Prolonging cell-free protein synthesis with a novel ATP regeneration system. *Biotechnol. Bioeng.* 66, 180–188. doi: 10.1002/(SICI)1097-0290(1999)66:3<180::AID-BIT6>3.0.CO;2-S

Kim, D. M., and Swartz, J. R. (2001). Regeneration of adenosine triphosphate from glycolytic intermediates for cell-free protein synthesis. *Biotechnol. Bioeng.* 74, 309–316. doi: 10.1002/bit.1121

Kim, H.-C., and Kim, D.-M. (2009). Methods for energizing cell-free protein synthesis. *J. Biosci. Bioeng.* 108, 1–4. doi: 10.1016/j.jbiosc.2009.02.007

Kim, J., and Winfree, E. (2011). Synthetic *in vitro* transcriptional oscillators. *Mol. Syst. Biol.* 7, 1–15. doi: 10.1038/msb.2010.119

Korman, T. P., Opgenorth, P. H., and Bowie, J. U. (2017). A synthetic biochemistry platform for cell free production of monoterpenes from glucose. *Nat. Commun.* 8, 1–8. doi: 10.1038/ncomms15526

Laohakunakorn, N., Grasemann, L., Lavickova, B., Michielin, G., Shahein, A., Swank, Z., et al. (2020). Bottom-up construction of complex biomolecular systems with cell-free synthetic biology. *Front. Bioeng. Biotechnol.* 8:213. doi: 10.3389/fbioe.2020.00213

Lavickova, B., and Maerkl, S. J. (2019). A simple, robust, and low-cost method to produce the PURE cell-free system. *ACS Synth. Biol.* 8, 455–462. doi: 10.1021/acssynbio.8b00427

Lehr, F. X., Hanst, M., Vogel, M., Kremer, J., Göringer, H. U., Suess, B., et al. (2019). Cell-free prototyping of AND-logic gates based on heterogeneous RNA activators. *ACS Synth. Biol.* 8, 2163–2173. doi: 10.1021/acssynbio.9b00238

Lewis, D. D., Villarreal, F. D., Wu, F., and Tan, C. (2014). Synthetic biology outside the cell: linking computational tools to cell-free systems. *Front. Bioeng. Biotechnol.* 2:66. doi: 10.3389/fbioe.2014.00066

Li, J. J., and Kelly, T. K. (1984). Simian virus 40 DNA replication *in vitro*. *Proc. Natl. Acad. Sci. U.S.A.* 81, 6973–6977. doi: 10.1073/pnas.81.22.6973

Libicher, K., Hornberger, R., Heymann, M., and Mutschler, H. (2020). *In vitro* self-replication and multicistronic expression of large synthetic genomes. *Nat. Commun.* 11:904. doi: 10.1038/s41467-020-14694-2

Luisi, P. L., Ferri, F., and Stano, P. (2006). Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften* 93, 1–13. doi: 10.1007/s00114-005-0056-z

MacDonald, J. T., Barnes, C., Kitney, R. I., Freemont, P. S., and Stan, G. B. V. (2011). Computational design approaches and tools for synthetic biology. *Integr. Biol.* 3, 97–108. doi: 10.1039/c0ib00077a

Marshall, R., and Noireaux, V. (2019). Quantitative modeling of transcription and translation of an all-*E. coli* cell-free system. *Sci. Rep.* 9, 1–12. doi: 10.1038/s41598-019-48468-8

Matsuda, T., Watanabe, S., and Kigawa, T. (2013). Cell-free synthesis system suitable for disulfide-containing proteins. *Biochem. Biophys. Res. Commun.* 431, 296–301. doi: 10.1016/j.bbrc.2012.12.107

Matsuura, T., Hosoda, K., and Shimizu, Y. (2018). Robustness of a reconstituted *Escherichia coli* protein translation system analyzed by computational modeling. *ACS Synth. Biol.* 7, 1964–1972. doi: 10.1021/acssynbio.8b00228

Matsuura, T., Tanimura, N., Hosoda, K., Yomo, T., and Shimizu, Y. (2017). Reaction dynamics analysis of a reconstituted *Escherichia coli* protein translation system by computational modeling. *Proc. Natl. Acad. Sci. U.S.A.* 114, E1336–E1344. doi: 10.1073/pnas.1615351114

Mavelli, F., Marangoni, R., and Stano, P. (2015). A simple protein synthesis model for the PURE system operation. *Bull. Math. Biol.* 77, 1185–1212. doi: 10.1007/s11538-015-0082-8

Meyer, A. J., Ellefson, J. W., and Ellington, A. D. (2012). Abiotic self-replication. *Acc. Chem. Res.* 45, 2097–2105. doi: 10.1021/ar200325v

Moore, S. J., MacDonald, J. T., Wienecke, S., Ishwarbhai, A., Tsipa, A., Aw, R., et al. (2018). Rapid acquisition and model-based analysis of cell-free transcription–translation reactions from nonmodel bacteria. *Proc. Natl. Acad. Sci. U.S.A.* 115, E4340–E4349. doi: 10.1073/pnas.1715806115

Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013). Implementation of cell-free biological networks at steady state. *Proc. Natl. Acad. Sci. U.S.A.* 110, 15985–15990. doi: 10.1073/pnas.1311166110

Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., et al. (2015). Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* 4, 1–18. doi: 10.7554/eLife.09771

Nielsen, A. A., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., et al. (2016). Genetic circuit design automation. *Science* 352:aac7341. doi: 10.1126/science.aac7341

Nieß, A., Failmezger, J., Kuschel, M., Siemann-Herzberg, M., and Takors, R. (2017). Experimentally validated model enables debottlenecking of *in vitro* protein synthesis and identifies a control shift under *in vivo* conditions. *ACS Synth. Biol.* 6, 1913–1921. doi: 10.1021/acssynbio.7b00117

Nirenberg, M. W., and Matthaei, J. H. (1961). The dependence of cell-free protein synthesis in *E. coli* upon naturally occurring or synthetic polyribonucleotides. *Proc. Natl. Acad. Sci. U.S.A.* 47, 1588–1602. doi: 10.1073/pnas.47.10.1588

Noireaux, V., and Libchaber, A. (2004). A vesicle bioreactor as a step toward an artificial cell assembly. *Proc. Natl. Acad. Sci. U.S.A.* 101, 17669–17674. doi: 10.1073/pnas.0408236101

Pardee, K., Green, A. A., Ferrante, T., Cameron, D. E., Daleykeyser, A., Yin, P., et al. (2014). Paper-based synthetic gene networks. *Cell* 159, 940–954. doi: 10.1016/j.cell.2014.10.004

Perez, J. G., Stark, J. C., and Jewett, M. C. (2016). Cell-free synthetic biology: engineering beyond the cell. *Cold Spring Harbor Perspect. Biol.* 8:a023853. doi: 10.1101/cshperspect.a023853

Sakatani, Y., Ichihashi, N., Kazuta, Y., and Yomo, T. (2015). A transcription and translation-coupled DNA replication system using rolling-circle replication. *Sci. Rep.* 5, 1–5. doi: 10.1038/srep10404

Sample, P. J., Wang, B., Reid, D. W., Presnyak, V., McFadyen, I. J., Morris, D. R., et al. (2019). Human 5' UTR design and variant effect prediction from a massively parallel translation assay. *Nat. Biotechnol.* 37, 803–809. doi: 10.1038/s41587-019-0164-5

Schwille, P., Spatz, J., Landfester, K., Herminghaus, S., Sourjik, V., Erb, T. J., et al. (2018). MaxSynBio: avenues towards creating cells from the bottom up. *Angew. Chem. Int. Ed.* 57, 13382–13392. doi: 10.1002/anie.201802288

Shimizu, Y., Inoue, A., Tomari, Y., Suzuki, T., Yokogawa, T., Nishikawa, K., et al. (2001). Cell-free translation reconstituted with purified components. *Nat. Biotechnol.* 19, 751–755. doi: 10.1038/90802

Siegal-Gaskins, D., Tuza, Z. A., Kim, J., Noireaux, V., and Murray, R. M. (2014). Gene circuit performance characterization in a cell-free 'breadboard'. *ACS Synth. Biol.* 3, 416–425. doi: 10.1021/sb400203p

Silverman, A. D., Karim, A. S., and Jewett, M. C. (2019a). Cell-free gene expression: an expanded repertoire of applications. *Nat. Rev. Genet.* 21, 151–170. doi: 10.1038/s41576-019-0186-3

Silverman, A. D., Kelley-Loughnane, N., Lucks, J. B., and Jewett, M. C. (2019b). Deconstructing cell-free extract preparation for *in vitro* activation of transcriptional genetic circuitry. *ACS Synth. Biol.* 8, 403–414. doi: 10.1021/acssynbio.8b00430

Sitaraman, K., Esposito, D., Klarmann, G., Le Grice, S. F., Hartley, J. L., and Chatterjee, D. K. (2004). A novel cell-free protein synthesis system. *J. Biotechnol.* 110, 257–263. doi: 10.1016/j.jbiotec.2004.02.014

Spirin, A. S., Baranov, V. I., Ryabova, L. A., Ovodov, S. Y., and Alakhov, Y. B. (1988). A continous cell-free translation system capable of producing polypeptides in high yield. *Science* 242, 1162–1164. doi: 10.1126/science.3055301

Stillman, B. W., and Gluzman, Y. (1985). Replication and supercoiling of simian virus 40 DNA in cell extracts from human cells. *Mol. Cell. Biol.* 5, 2051–2060. doi: 10.1128/MCB.5.8.2051

Stögbauer, T., Windhager, L., Zimmer, R., and Rädler, J. O. (2012). Experiment and mathematical modeling of gene expression dynamics in a cell-free system. *Integr. Biol.* 4, 494–501. doi: 10.1039/c2ib00102k

Swank, Z., Laohakunakorn, N., and Maerkl, S. J. (2019). Cell-free gene-regulatory network engineering with synthetic transcription factors. *Proc. Natl. Acad. Sci. U.S.A.* 116, 5892–5901. doi: 10.1073/pnas.1816591116

Takahashi, M. K., Hayes, C. A., Chappell, J., Sun, Z. Z., Murray, R. M., Noireaux, V., et al. (2015). Characterizing and prototyping genetic networks with cell-free transcription-translation reactions. *Methods* 86, 60–72. doi: 10.1016/j.ymeth.2015.05.020

Tuza, Z. A., Siegal-Gaskins, D., Kim, J., and Szederkenyi, G. (2015). Analysis-based parameter estimation of an *in vitro* transcription-translation system. *Eur. Control Conf.* 2015, 1554–1560. doi: 10.1109/ECC.2015.7330760

van Nies, P., Westerlaken, I., Blanken, D., Salas, M., Mencía, M., and Danelon, C. (2018). Self-replication of DNA by its encoded proteins in liposome-based synthetic cells. *Nat. Commun.* 9:1583. doi: 10.1038/s41467-018-03926-1

Vecchio, D. D., Dy, A. J., and Qian, Y. (2016). Control theory meets synthetic biology. *J. R. Soc. Interface* 13:20160380. doi: 10.1098/rsif.2016.0380

Vilkhovoy, M., Dai, D., Vadhin, S., Adhikari, A., and Varner, J. D. (2019). Absolute quantification of cell-free protein synthesis metabolism by reversed-phase liquid chromatography-mass spectrometry. *J. Vis. Exp.* 152:e60329. doi: 10.3791/60329

Vilkhovoy, M., Horvath, N., Shih, C. H., Wayman, J. A., Calhoun, K., Swartz, J., et al. (2018). Sequence specific modeling of *E. coli* cell-free protein synthesis. *ACS Synth. Biol.* 7, 1844–1857. doi: 10.1021/acssynbio.7b00465

Wayman, J. A., Sagar, A., and Varner, J. D. (2015). Dynamic modeling of cell-free biochemical networks using effective kinetic models. *Processes* 3, 138–160. doi: 10.3390/pr3010138

Westbrook, A., Tang, X., Marshall, R., Maxwell, C. S., Chappell, J., Agrawal, D. K., et al. (2019). Distinct timescales of RNA regulators enable the construction of a genetic pulse generator. *Biotechnol. Bioeng.* 116, 1139–1151. doi: 10.1002/bit.26918

Zubay, G. (1973). *In vitro* synthesis of protein in microbial systems. *Annu. Rev. Genet.* 7, 267–287. doi: 10.1146/annurev.ge.07.120173.001411

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Genetics

ORIGINAL RESEARCH
published: 27 July 2020
doi: 10.3389/fgene.2020.00706

SSRMMD: A Rapid and Accurate Algorithm for Mining SSR Feature Loci and Candidate Polymorphic SSRs Based on Assembled Sequences

Xiangjian Gou[1,2†], Haoran Shi[1†], Shifan Yu[1], Zhiqiang Wang[1], Caixia Li[1], Shihang Liu[1], Jian Ma[1], Guangdeng Chen[3], Tao Liu[4] and Yaxi Liu[1,5*]*

[1] Triticeae Research Institute, Sichuan Agricultural University, Chengdu, China, [2] Maize Research Institute, Sichuan Agricultural University, Chengdu, China, [3] College of Resources, Sichuan Agricultural University, Chengdu, China, [4] College of Information Engineering, Sichuan Agricultural University, Ya'an, China, [5] State Key Laboratory of Crop Gene Exploration and Utilization in Southwest China, Chengdu, China

Microsatellites or simple sequence repeats (SSRs) are short tandem repeats of DNA widespread in genomes and transcriptomes of diverse organisms and are used in various genetic studies. Few software programs that mine SSRs can be further used to mine polymorphic SSRs, and these programs have poor portability, have slow computational speed, are highly dependent on other programs, and have low marker development rates. In this study, we develop an algorithm named Simple Sequence Repeat Molecular Marker Developer (SSRMMD), which uses improved regular expressions to rapidly and exhaustively mine perfect SSR loci from any size of assembled sequence. To mine polymorphic SSRs, SSRMMD uses a novel three-stage method to assess the conservativeness of SSR flanking sequences and then uses the sliding window method to fragment each assembled sequence to assess its uniqueness. Furthermore, molecular biology assays support the polymorphic SSRs identified by SSRMMD. SSRMMD is implemented using the Perl programming language and can be downloaded from https://github.com/GouXiangJian/SSRMMD.

Keywords: bioinformatics, algorithm, simple sequence repeats, conservativeness, uniqueness, polymorphism

OPEN ACCESS

Edited by:
Thomas Edward Gorochowski, University of Bristol, United Kingdom

Reviewed by:
Jian-Feng Mao, Beijing Forestry University, China
Mir Asif Iquebal, Indian Agricultural Statistics Research Institute (ICAR), India
Suresh Babu Mudunuri, Sagi Ramakrishnam Raju Engineering College, India

***Correspondence:**
Tao Liu
liutao@sicau.edu.cn
Yaxi Liu
liuyaxi@sicau.edu.cn

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *07 April 2020*
Accepted: *10 June 2020*
Published: *27 July 2020*

Citation:
Gou X, Shi H, Yu S, Wang Z, Li C, Liu S, Ma J, Chen G, Liu T and Liu Y (2020) SSRMMD: A Rapid and Accurate Algorithm for Mining SSR Feature Loci and Candidate Polymorphic SSRs Based on Assembled Sequences. Front. Genet. 11:706. doi: 10.3389/fgene.2020.00706

INTRODUCTION

Owing to their abundance, codominant inheritance, multi-allelic nature, transferability, and ease of analysis *via* PCR (Varshney et al., 2005; Ramu et al., 2009; Kaur et al., 2015), simple sequence repeat (SSR) markers have been successfully adopted in various genetic studies such as quantitative trait loci mapping (Qin et al., 2015; Wang et al., 2017), genotyping (Gramazio et al., 2018), genetic diversity (Nachimuthu et al., 2015; Zhou R. et al., 2015), and DNA fingerprinting

(Zhang et al., 2015). Indeed, numerous genome-wide SSR markers have been identified in plants and animals in recent years, such as those in rice (Zhang et al., 2007), maize (Xu et al., 2013), cucumber (Liu et al., 2015), bee (Liu et al., 2016), tobacco (Wang et al., 2018), and snake (Liu et al., 2019).

During the development of SSR markers, the first step is the mining of potential SSR loci from assembled sequences. Based on the repetitive architecture of their motifs, SSRs can be classified as perfect (e.g., AGAGAGAGAGAG), and imperfect (including nucleotide substitutions or indels, e.g., AGAGAGACAGAG). However, the application of perfect SSRs in genetic studies far exceeds that of imperfect SSRs because of its higher allelic variability (Zalapa et al., 2012; Xu et al., 2013). Numerous algorithms and software programs have been reported for mining perfect SSRs. For instance, SSRIT (Temnykh, 2001), MISA (Thiel et al., 2003), and GMATo (Wang et al., 2013) use regular expressions based on the greedy matching algorithm to mine SSRs. SA-SSR (Pickett et al., 2016) uses a suffix array-based algorithm to mine SSRs. Kmer-SSR (Pickett et al., 2017) uses Kmer decomposition to identify SSRs. PERF (Avvaru et al., 2017) matches each potential substring in accordance with a set of pre-computed repeat strings. Other programs including TROLL (Castelo et al., 2002), MfSAT (Chen et al., 2011), ProGeRF (Silva et al., 2015), and FullSSR (Metz et al., 2016) have also been developed. In addition, imperfect SSR detection algorithms have also been reported, such as IMEx (Mudunuri and Nagarajaram, 2007), and Krait (Du et al., 2017). However, these programs have many common undesirable features. First, they rely on additional software or modules, often with complex software configuration; second, they have poor portability and can only be run on Linux or Windows platforms; third, they have slow computational speed; and most importantly, polymorphic SSRs cannot be directly found.

With rapid advancements in genomics, software and pipelines for mining polymorphic SSRs have been reported. For instance, CandiSSR (Xia et al., 2016), a candidate polymorphic SSRs identification pipeline, is based on multiple assembly sequences. GMATA (Wang and Wang, 2016) provides a complete process for SSR markers development. IDSSR (Guang et al., 2019) has recently been reported to identify polymorphic SSRs in a single genome sequences using a similar pipeline. However, these programs or pipelines also share certain issues. First, they rely on numerous other programs, such as MISA (Thiel et al., 2003), Primer3 (Untergasser et al., 2012), BLAST (Altschul et al., 1997), and ClustalW (Thompson et al., 2002); second, they have slow computational speed for mining polymorphic SSRs; finally, they have low rates of SSR markers development.

To overcome these limitations, we developed the Simple Sequence Repeat Molecular Marker Developer (SSRMMD) program using the Perl programming language. This program rapidly and exhaustively mines perfect SSR loci through improved regular expressions. For mining polymorphic SSRs, this program uses a high-stringency sequence alignment algorithm to assess the conservativeness and uniqueness of SSR flanking sequences. Compared with other software programs, SSRMMD is more rapid, accurate, and convenient. SSRMMD can be downloaded from https://github.com/GouXiangJian/SSRMMD.

MATERIALS AND METHODS

Implemented Algorithm

The algorithm of SSRMMD involves the mining of perfect SSR loci and the discovery of polymorphic SSRs. The internal methodological details are provided in **Figure 1**, primarily including the following steps:

(1) Mining perfect SSR loci. Similar to programs such as SSRIT (Temnykh, 2001) and MISA (Thiel et al., 2003), SSRMMD uses regular expressions with the greedy matching algorithm to mine SSRs. However, to improve computational speed, SSRMMD was optimized in three aspects: (i) use of multi-threading technology. To maximize the function of each thread, we proposed a novel optimal allocation algorithm to averagely distribute assembled sequences to each thread in accordance with the length of sequences (TOS), including the following: (a) sort sequences by TOS; (b) assignment of the longest i sequences to i threads; (c) thread sorting based on the total TOS; (d) assignment of subsequent sequences to the thread with the smallest TOS; (e) thread sorting in step (d) using the insertion sorting algorithm; and (f) iterative performance of steps (d) and (e) until complete sequence allocation. (ii) Fragmented sequences. After a specific thread is assigned to store each sequence, SSRMMD fragmented each sequence into short 500-kb fragments. At this length, the computational speed was the highest. Furthermore, 5 kb was added to each fragment to prevent potential SSRs from being cut off. (iii) Improved regular expression. Ordinary regular expressions can only mine one type of motif in each match, as indicated using MISA (Thiel et al., 2003). However, by integrating all patterns, SSRMMD can mine all types of motif in each match, indicating that irrespective of the arrangement of the threshold motifs, SSRMMD will only traverse the sequence once. Notably, to completely mine compound SSRs, SSRMMD backtracks after each successful match, and the size of backtracking (B) is as follows:

$$size\,(B) = \begin{cases} length\left(motif_i\right) - 1, & sum\left(motif_i\right) == 1 \\ \max\left\{L_1, L_2, \ldots, L_n\right\}, & \\ \min\left\{S_1, S_2, \ldots, S_n\right\} & > \max\left\{L_1,\ L_2,\ \ldots, L_n\right\} \\ \min\left\{S_1, S_2, \ldots, S_n\right\} - 1, & \\ \min\left\{S_1, S_2, \ldots, S_n\right\} & \leq \max\left\{L_1,\ L_2,\ \ldots, L_n\right\} \end{cases}$$

where n is the number of motif types, S_i is the length of the ith motif types of SSR, and L_i is the length of the ith type of motif.

(2) Assessment of the conservativeness of SSR flanking sequences. To develop polymorphic SSRs, we initially assessed the conservativeness of SSR flanking sequences. To maximize the computational speed, we used a novel three-stage method to align flanking sequences between two assembled sequences files, which included the following steps: (i) first, absolutely conserved flanking sequences were filtered out using HASH structure. Herein, we considered these flanking sequences in the first assembled file as a library, and then we compared

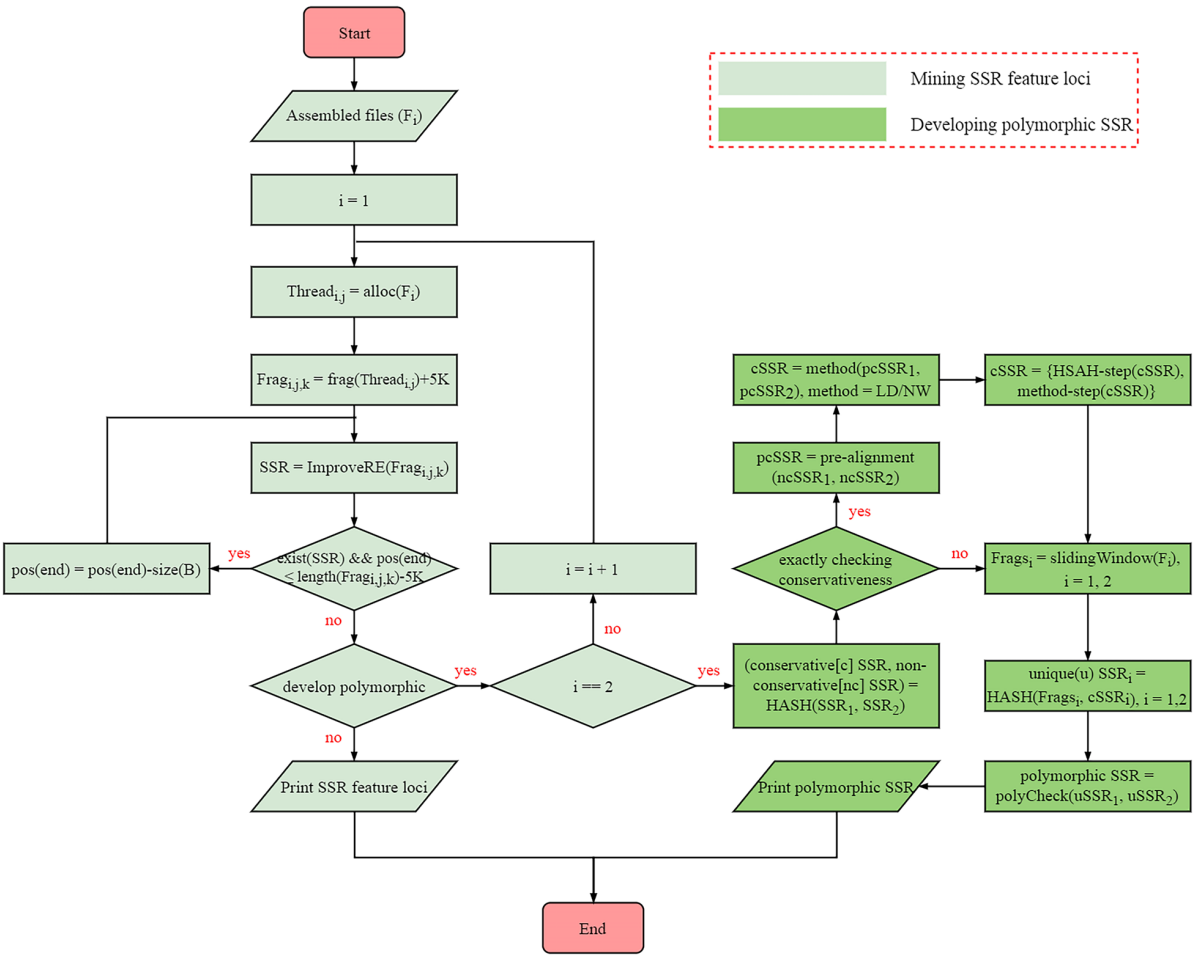


FIGURE 1 | Internal implementation algorithm of Simple Sequence Repeat Molecular Marker Developer (SSRMMD).

these flanking sequences in the another assembled file with the aforementioned library to rapidly identify absolutely conserved flanking sequences. (ii) Second, conservativeness pre-alignment was performed using x% [default is 5% (each side is 5 bp)] flanking sequences near SSRs. Assuming that flanking sequences near SSRs were highly conserved, SSRMMD allowed flanking sequences near SSRs to tolerate up to 2-bp mismatches. Moreover, after extensive assessments, additional mismatches ($\geq$3 bp) did not further benefit the results, consistent with the aforementioned assumption. SSRMMD iteratively replaced mismatched bases and aligned flanking sequences between two assembled files, using a method similar to (i). (iii) Finally, SSRMMD used Levenshtein distance (LD; Levenshtein, 1966), or the Needleman–Wunsch (NW) algorithm (Needleman and Wunsch, 1970) to accurately assess the conservativeness of the flanking sequences retained through pre-alignment. LD was defined as the minimum number of edits required to convert one string to another, thus indirectly reflecting the identity of two DNA sequences. However, the NW algorithm based on dynamic programming has been extensively used for global sequence alignment, directly reflecting the identity of two DNA sequences. Compared with NW algorithm, the LD did not require backtracking; hence, it had a higher computational speed; furthermore, the NW algorithm had a more comprehensive scoring system than LD, thus facilitating more accurate elucidation of the identity of the SSR flanking sequences. The iterative formulae of the LD and NW algorithms are as follows:

$$LD_{a,b}(i,j) = \begin{cases} \max(i,j) & \min(i,j) = 0 \\ \min \begin{cases} LD_{a,b}(i-1,j)+1 \\ LD_{a,b}(i,j-1)+1 \\ LD_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases}$$

$$NW_{a,b}(i,j) = \begin{cases} 0 & \text{i, j} = 0 \\ max \begin{cases} NW_{a,b}(i-1,j) + S_{gap} & a_i \text{ aligned to a gap} \\ NW_{a,b}(i,j-1) + S_{gap} & b_j \text{ aligned to a gap} \\ NW_{a,b}(i-1,j-1) + S_{match/mismatch} & a_i \text{ aligned to } b_j \end{cases} \end{cases}$$

where a and b are 2 strings; i and j are subscripts of a and b, respectively; S_{match} is the score of match; $S_{mismatch}$ is the score of mismatch; and S_{gap} is the score of gap.

(3) Assessment of the uniqueness of SSR flanking sequences. After conservativeness was assessed, SSRMMD further assessed the uniqueness of SSR flanking sequences. Again, assembled sequences were evenly distributed to each thread and were fragmented through the sliding window method, wherein window size was the length of flanking sequences, the step size was 1 bp, and all fragments were stored in a HASH database. Thereafter, flanking sequences with the equal sizes in the aforementioned HASH database were aligned to identify SSRs with unique flanking sequences. Finally, polymorphisms were compared in the two unique SSR sets to distinguish monomorphic and polymorphic SSRs. Notably, to meet different needs, SSRMMD used two computational methods, (i) running in a time-saving manner and (ii) running in a memory-saving manner, indicating that SSRMMD functions adequately, irrespective of the use of a personal computer, or high-performance server.

Input and Output

Assembled sequences (e.g., genome, transcriptome, or a single gene) with a standard FASTA format is required for mining SSRs; to further develop candidate polymorphic SSRs, another assembled sequence is required. Certain parameters can be set to change the SSR mining conditions, including motif threshold and the length of flanking sequences. SSRMMD is allowed to set any size motif (>6 bp), and SSRMMD would then assess the conservativeness and uniqueness of SSR flanking sequences when mining polymorphic SSRs. Notably, setting more threads would significantly enhance the computational speed.

Upon completion of the computation, SSRMMD yields three types of outputs: (i) detailed information record file of SSRs; (ii) statistical file of SSRs, which analyzes the various distribution characteristics of SSRs and helps understand the distribution pattern of the SSRs [including the following: (a) SSR number and density in each assembled sequence; (b) SSR number and proportion per unit length of the motif; and (c) SSR number among different numbers of repeats in each motif]; and (iii) detailed information record file of candidate polymorphic SSRs.

Performance Test Datasets

To assess SSRMMD, we downloaded six genomes of three plants from National Center for Biotechnology Information (NCBI)[1] and Unité de Recherche Génomique Info (URGI)[2]. Three genomes were used to assess the potential for mining SSR feature loci, including rice (Zhenshan97, ∼0.39 Gb), cotton (TM1, ∼2.29 Gb), and wheat [Chinese Spring (CS), ∼14.23 Gb]. All six genomes were used to assess the potential to mine polymorphic SSRs, including two rice genomes, two cotton genomes, and two wheat genomes. To evaluate the complexity and multi-threading of SSRMMD, we extracted 2-Gb sequences from the wheat CS and AK58 genomes, which were evenly divided into 20 sequences. The GenBank assembly accession numbers of the rice genomes were Zhenshan97 (GCA_001623345.2) and Shuhui498 (GCA_002151415.1). The GenBank assembly accession numbers of cotton genomes were TM1 (GCA_006980745.1) and ZM24 (GCA_006980775.1). The wheat CS and AK58 genomes were obtained from URGI.

[1]https://www.ncbi.nlm.nih.gov/

[2]https://urgi.versailles.inra.fr/

Performance Test Parameters

Perfect repeats have higher allelic variability than imperfect repeats, and any SSR used to develop genetic markers should contain a perfect repeat (Xu et al., 2013). Therefore, to assess the potential of SSRMMD for mining SSR loci, we avoided imperfect repeats detection tools, and we selected six popular existing software programs including SSRIT (Temnykh, 2001), MISA (Thiel et al., 2003), GMATA (Wang and Wang, 2016), SA-SSR (Pickett et al., 2016), Kmer-SSR (Pickett et al., 2017), and PERF (Avvaru et al., 2017). In particular, SA-SSR was not included in the results owing to its markedly low computational speed. In each software program, based on previously described methods (Zhang et al., 2007; Xu et al., 2013; Liu et al., 2015), the minimum repeat times of SSR motif lengths of 1, 2, 3, 4, 5, and 6 bp were set to 10, 7, 6, 5, 4, and 4, respectively. Because Kmer-SSR can use multi-threads, we tested SSRMMD and Kmer-SSR with 1 and 12 threads, respectively, to assess its multi-thread support. However, other software programs could only use a single thread.

To assess the potential for mining polymorphic SSRs, we compared two popular existing software programs with SSRMMD, including GMATA (Wang and Wang, 2016), and CandiSSR (Xia et al., 2016). In each software program, SSR flanking sequences were set to 150 bp (Zhang et al., 2007). Because CandiSSR can use multi-threads, we assessed SSRMMD, and CandiSSR with 12 threads; however, GMATA can only use a single thread. On assessing SSRMMD, LD was used to assess the conservativeness of flanking sequences, and the threshold was set to 5% to correspond to the BLAST identity of CandiSSR, and the other parameters (not indicated herein) were retained as default setting. Similarly, parameters not included in GMATA and CandiSSR were used with default setting.

Performance Evaluation Criteria

The performance of SSRMMD and existing software programs for mining perfect SSRs was evaluated in accordance with six criteria. **Table 1** shows the portability, dependence, and function of existing software programs and SSRMMD. The computational accuracy, speed, and memory consumption were evaluated for the test datasets. We used the Linux *time* command to record the computational time and *pmap* command to record the memory peak. All tests were performed using a personal computer with an Intel® Xeonl® CPU E5-2683 v3 @ 2.00 GHz with CentOS Linux release 7.4.1708 and 64 GB RAM.

Experimental Validation

To verify the accuracy of the output through SSRMMD, 80 pairs of polymorphic SSRs were randomly selected from the computational results of wheat for molecular biology assays. These selected polymorphic SSRs were evenly distributed on each chromosome, encompassing differently sized motifs. Genomic

TABLE 1 | Various features of SSRMMD and existing software programs for mining perfect SSRs.

Software	Year	Portability	Dependence[a]	Function
SSRIT	2001	Windows/Linux	No	Mining SSR feature loci
TROLL	2002	Windows/Linux	Staden	Mining SSR feature loci
MISA	2003	Windows/Linux	No	Mining SSR feature loci
MfSAT	2011	Windows	No	Mining SSR feature loci
GMATo	2013	Windows/Linux	No	Mining SSR feature loci
ProGeRF	2015	Linux	No	Mining SSR feature loci
CandiSSR	2016	Linux	MISA, BLAST, Primer3, Clustalw	Developing polymorphic SSRs
FullSSR	2016	Windows/Linux	BioPerl, Bio:Tools:Run: Primer3	Mining SSR feature loci
GMATA	2016	Windows/Linux	Primer3, e-PCR	Developing polymorphic SSRs
SA-SSR	2016	Linux	No	Mining SSR feature loci
Kmer-SSR	2017	Linux	No	Mining SSR feature loci
PERF	2017	Windows/Linux	tqdm, biopython	Mining SSR feature loci
IDSSR	2019	Linux	SSRIT, BLAST, Primer3	Developing polymorphic SSRs
SSRMMD	this	Windows/Linux	No	Developing polymorphic SSRs

Note. SSRMMD, Simple Sequence Repeat Molecular Marker Developer.
[a]Dependence on additional programs or modules.

DNA was extracted using the cetyl trimethylammonium bromide (CTAB) method from fresh leaves of 10 wheat popularized and local cultivars of CS, AK58, CM107, CN16, MM37, ZM012542, ZM000652, ZM018703, ZM003222, and ZM003284.

Additionally, we provided a tool named connectorToPrimer3 to associate SSRMMD with Primer3 (Untergasser et al., 2012); hence, a primer design can be easily performed. The primary parameters were as follows: (1) minimum, optimal, and maximum primer sizes of 18, 20, and 27 bp, respectively; (2) minimum and maximum GC contents of 20% and 80%, respectively; (3) minimum, optimal, and maximum Tm values of 57, 60, and 63°C, respectively; and (4) product lengths of 100–300 bp. Primers were synthesized by Beijing Qingke Biotechnology Co., Ltd.

PCR was performed in 10-μl reactions containing 5 μl of mix buffer (2×), 1.0 μl of template DNA (100 ng/μl), 0.5 μl of primers, and 3 μl of ddH_2O. The PCR conditions were as follows: 1 cycle at 94°C for 5 min, 35 cycles at 94°C for 30 s, 60°C for 30 s, 72°C for 30 s, and 1 cycle at 72°C for 10 min. The PCR products were electrophoresed on a 6% denaturing polyacrylamide gel. SSR polymorphisms in different wheat genotypes were identified on the basis of differences in mobility, as revealed through the electrophoretic bands.

RESULTS

Assessment of Complexity and Threads

On the basis of the 2-Gb base sequences from wheat, we tested the time and space complexity of SSRMMD in a single thread. As shown in **Figures 2A,B**, as the amount of data increased, the time and space consumed by SSRMMD increased linearly when mining SSR feature loci. Similarly, when SSRMMD was used to mine polymorphic SSRs (assessing uniqueness in a memory-saving manner), the time and space were also linearly associated with the amount of data (**Figures 2C,D**). These results suggest that the algorithm of SSRMMD has linear time complexity [$T(n) = O(n)$] and space complexity [$S(n) = O(n)$].

Furthermore, we assessed the multi-threading support of SSRMMD. As shown in **Figures 2E,F**, whether mining in SSR feature loci or polymorphic SSRs, as the number of threads increased, time consumption decayed as a power function with the number of threads; however, memory consumption scaled linearly. Notably, despite using 10 threads, memory consumption of SSRMMD did not exceed the size of test data (2 Gb). In total, SSRMMD adequately supported multi-threading.

Verification of the Performance of SSRMMD to Mine Simple Sequence Repeat Feature Loci

Based on the high citation rate and novel principles, six software programs were compared with SSRMMD (SA-SSR is not indicated). As shown in **Table 2**, SSRMMD identified the most SSRs. This was larger than other regular expression-based programs including MISA and GMATA. Furthermore, SSRMMD had the highest computational speed when running on a single thread and better supported multi-threading than Kmer-SSR. Additionally, we analyzed the validity of SSRs found by SSRMMD and compared them with four other programs (PERF, Kmer-SSR, GMATA, and MISA). As shown in **Figures 3A–C**, numerous common products were identified in these software programs, accounting for 76.95% (rice), 85.96% (cotton), and 74.21% (wheat) of SSRMMD, respectively.

Verification of the Performance of SSRMMD to Mine Polymorphic Simple Sequence Repeats

CandiSSR and GMATA were compared with SSRMMD. First, compared with CandiSSR, SSRMMD mined approximately doubled the number of polymorphic SSRs, and CandiSSR discarded numerous monomorphic SSRs from among these candidate markers (**Table 3**). Second, compared with GMATA, SSRMMD mined more polymorphic SSRs in rice and cotton, but less in wheat. However, because GMATA identified polymorphisms through e-PCR amplification products, which yield two forms of false positives, (1) the target SSR did not exist in the product and (2) the target SSR in the product was the same size as the reference SSR. Hence, we generated a script[3] to rectify the output of

[3]https://github.com/GouXiangJian/CorrectGMATA

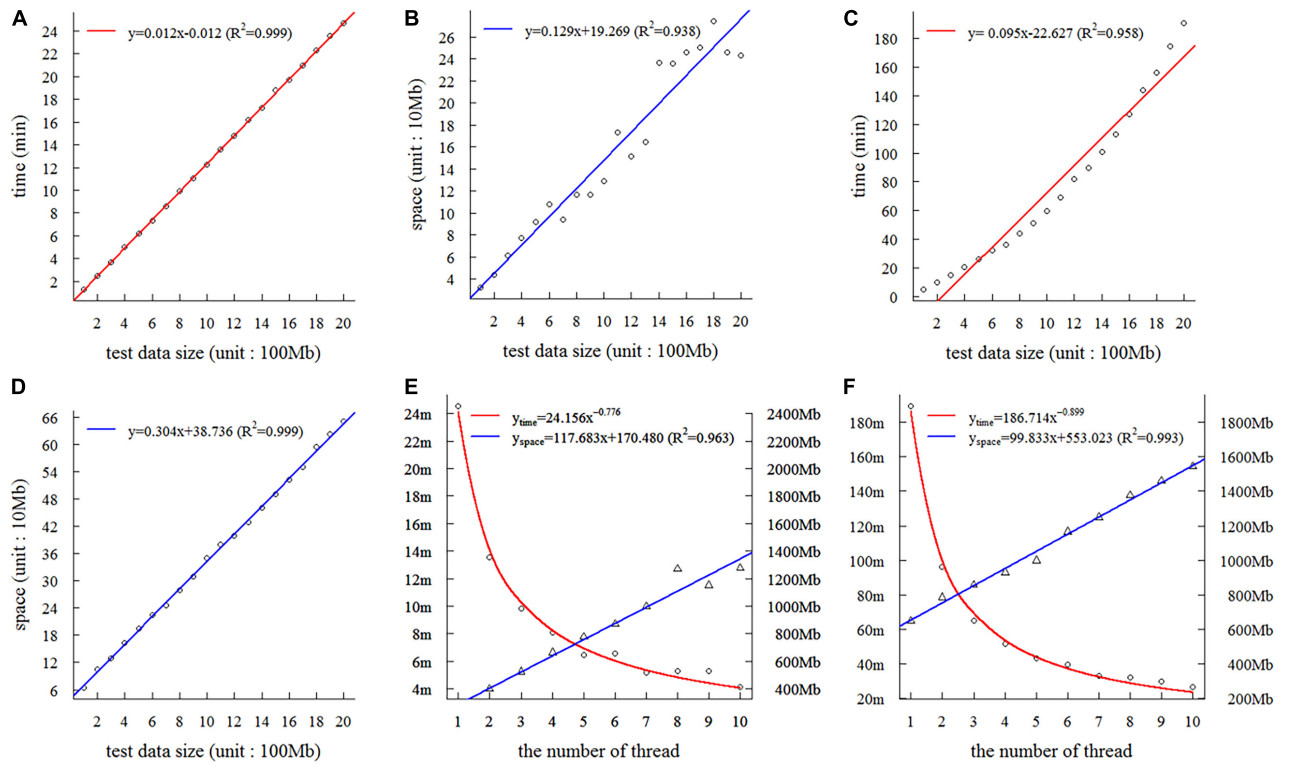


FIGURE 2 | Assessment of complexity and threads of Simple Sequence Repeat Molecular Marker Developer (SSRMMD). **(A)** The time complexity in mining SSR feature loci. **(B)** The space complexity in mining simple sequence repeat (SSR) feature loci. **(C)** Time complexity in developing polymorphic SSRs. **(D)** The space complexity in developing polymorphic SSRs. **(E)** The impact on time and space when increasing threads in mining SSR feature loci. **(F)** The impact on time and space when increasing threads in developing polymorphic SSRs.

TABLE 2 | Performance comparison between SSRMMD and other software programs for identifying genome-wide SSR feature loci.

Software	Thread	Rice (Zhenshan97, ~0.39 Gb)			Cotton (TM1, ~2.29 Gb)			Wheat (CS, ~14.23 Gb)		
		Number	Time (m:s)	Mem (Mb)	Number	Time (m:s)	Mem (Mb)	Number	Time (m:s)	Mem (Mb)
SSRIT	1	111,960	5:53	131.55	384,488	34:45	343.04	1,345,128	210:47	2,503.85
MISA	1	111,905	5:50	205.52	384,400	35:31	382.60	1,343,830	212:45	4,195.61
GMATA[a]	1	111,905	7:28	85.48	384,400	42:15	361.72	1,343,831	254:46	1,739.50
PERF	1	111,960	8:49	211.60	384,488	52:32	522.91	1,345,128	320:36	3,325.42
Kmer-SSR	1	111,960	14:08	123.05	384,488	83:14	169.00	1,345,128	516:19	1,028.44
Kmer-SSR	12	111,960	5:28	321.95	384,488	28:20	353.96	1,345,128	205:19	1,251.40
SSRMMD	1	111,960	4:49	139.38	384,488	28:36	421.95	1,345,128	175:19	2,404.68
SSRMMD	6	111,960	1:08	466.19	384,488	6:46	1,044.88	1,345,128	43:40	5,972.30
SSRMMD	12	111,960	0:49	731.02	384,488	4:28	1,717.92	1,345,128	27:05	11,248.89

Note. SSRMMD, Simple Sequence Repeat Molecular Marker Developer; SSR, simple sequence repeat. [a]*Because GMATA and Kmer-SSR could not simultaneously mine different types of motifs in one task, these two programs were multiply performed to identify SSRs, then the time was added, and memory peak was selected as the maximum among all tasks.*

GMATA, and we found that the actual polymorphic SSR numbers of GMATA were 7,452 (rice), 21,675 (cotton), and 188,342 (wheat). GMATA had a high false-positive rate, being 36.86% (4,350 of 11,802), 30.06% (9,317 of 30,992), and 25.33% (63,902 of 252,244) for rice, cotton, and wheat, respectively, thus implying a defect in the GMATA pipeline. SSRMMD required markedly less time, especially in the wheat genome (**Table 3**). Unfortunately, we could not quantify memory consumption because GMATA and CandiSSR called numerous other programs and scripts for mining polymorphic SSRs. Furthermore, we compared the output of polymorphic SSRs between SSRMMD and two other software programs. As shown in **Figures 3D–F**, approximately 70.48% (rice), 37.11% (cotton), and 49.19% (wheat) of the SSRMMD outputs were novel in comparison with other two software programs.

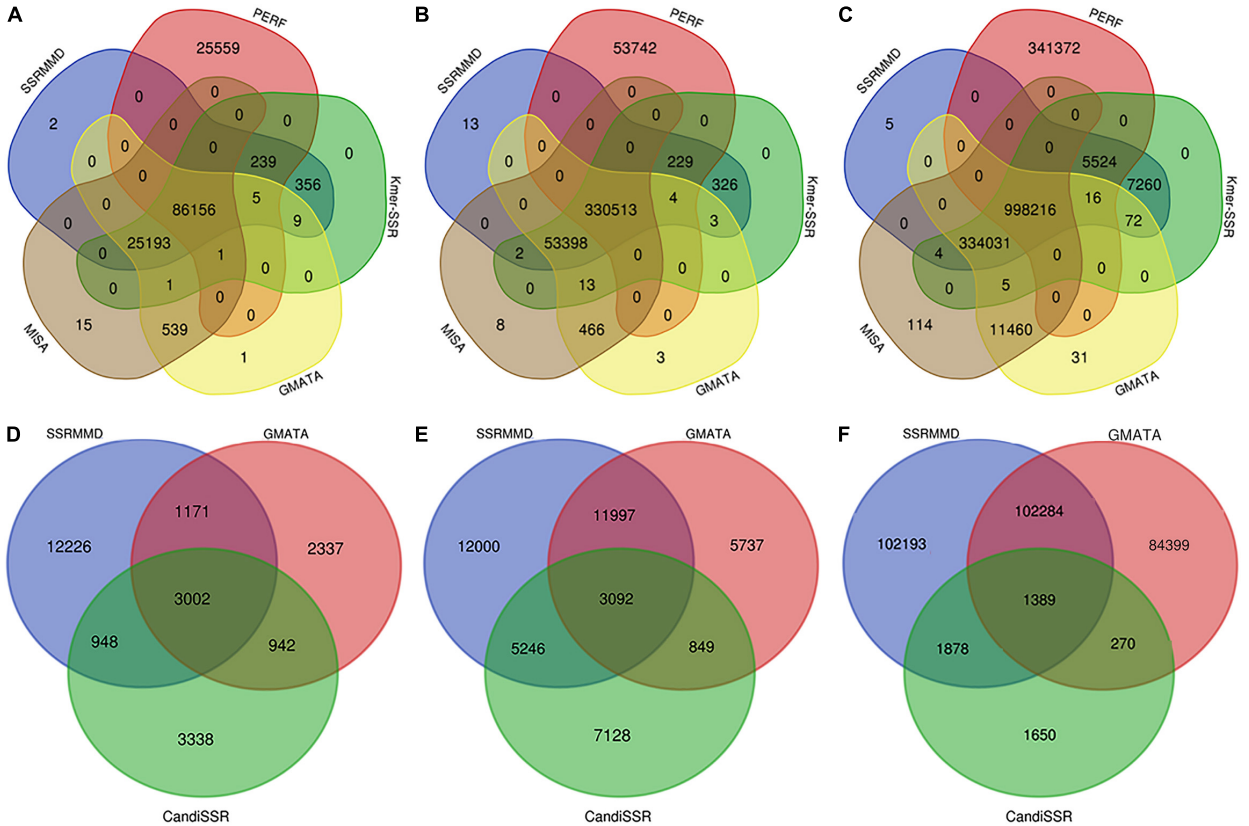


FIGURE 3 | Overlapping simple sequence repeats (SSRs) calculated by five programs [Simple Sequence Repeat Molecular Marker Developer (SSRMMD), PERF, Kmer-SSR, GMATA, and MISA], and overlapping polymorphic SSRs calculated by three programs (SSRMMD, GMATA, and CandiSSR). If the sequence name, start, and end position were the same in all programs, the SSR was considered overlapping. **(A)** Rice. **(B)** Cotton. **(C)** Wheat. **(D)** Rice. **(E)** Cotton. **(F)** Wheat. Notably, CandiSSR only mined 5,187 polymorphic SSRs of wheat chromosome 5D.

Experimental Verification of the Accuracy of Polymorphic Simple Sequence Repeats

To verify the accuracy of the output by SSRMMD, 80 pairs of polymorphic SSRs were randomly selected to identify polymorphisms in 10 wheat cultivars. As shown in **Figure 4** and **Supplementary Table 1**, 56 independent products were successfully amplified using 56 primer pairs. However, the remaining 24 primer pairs did not yield stable or clear bands, and the reasons may include the following: (1) we did not optimize the PCR amplification conditions for each primer pair and used a uniform annealing temperature for all primer amplifications; and (2) some primer designs were created in batches generated under uniform conditions, which may have defects. Forty-four (∼79%) among these 56 primer pairs revealed polymorphisms in CS and AK58, suggesting that SSRMMD had a high accuracy.

DISCUSSION

With rapid innovations in sequencing technologies, third-generation DNA markers such as single-nucleotide polymorphisms have become widely used (Zhou Z. et al., 2015; Yang et al., 2017). However, SSRs are still used in various genetic studies such as quantitative trait loci mapping, genotyping, genetic diversity, and marker-assisted selection because of their codominant inheritance, multi-allelic nature, and ease of amplification *via* PCR operation (Varshney et al., 2005; Ramu et al., 2009; Kaur et al., 2015). These features are not applicable to single-nucleotide polymorphisms. Therefore, development of SSR markers from diverse organisms still is important in biological studies.

In vitro SSR marker development based on the creation of a genomic library, screening of positive clones, and subsequent sequencing is time-consuming and expensive. Song et al. (2005) only developed 540 SSR flanking primer pairs in the wheat mapping study by *in vitro* methods. However, we easily obtained millions of SSR loci from the wheat CS genome using our *in silico* methods (**Table 1**). Undoubtedly, it is more rapid and economical to develop SSR markers by using bioinformatics tools and genotypic data, and *in silico* methods have gradually replaced *in vitro* methods.

Although numerous software programs have been developed for mining perfect SSRs from assembled sequences, the accuracy, speed, and flexibility of these programs need to be balanced

TABLE 3 | Performance comparison between SSRMMD and other two software programs for developing candidate polymorphic SSRs.

Organism	Rice (~0.39 Gb)			Cotton (~2.29 Gb)			Wheat (~14.23 Gb)		
Software	**SSRMMD**	**GMATA**	**CandiSSR[c]**	**SSRMMD**	**GMATA**	**CandiSSR**	**SSRMMD**	**GMATA**	**CandiSSR[d]**
Total number of SSR[a]	111,960	111,905	111,905	384,488	384,400	384,400	1,345,128	1,343,831	1,331,146
Number of candidate marker	68,242	34,667	8,230	292,307	166,813	16,315	572,023	477,531	129,461
Candidate marker rate (%)	60.95	30.98	7.35	76.02	43.40	4.24	42.53	35.54	9.73
Number of monomorphic SSR	50,895	22,865	0	259,972	135,821	0	364,279	225,287	0
Number of polymorphic SSR	17,347	11,802	8,230	32,335	30,992	16,315	207,744	252,244	129,461
Polymorphic rate (%)	15.49	10.55	7.35	8.41	8.06	4.24	15.44	18.77	9.73
Time (min:s)[b]	6:08	16:15	8,117:37	30:30	119:06	1,746:23	184:55	6,363:45	118,953:53

Note. SSRMMD, Simple Sequence Repeat Molecular Marker Developer; SSR, simple sequence repeat. [a]Total number of SSR referred to Zhenshan97 (rice), TM1 (cotton), and CS (wheat). [b]Because GMATA could not simultaneously mine different types of motifs in one task, it was multiply performed to identify SSRs, and then the time was added. [c]Because CandiSSR could not normally calculate chromosome 8 of rice (the program stopped at the BLAST stage), the results of CandiSSR did not include chromosome 8. [d]Because CandiSSR spent numerous time to mine polymorphic SSRs of wheat, we only selected chromosome 5D (closest to the total SSR density of wheat) to estimate the number of polymorphic SSRs.

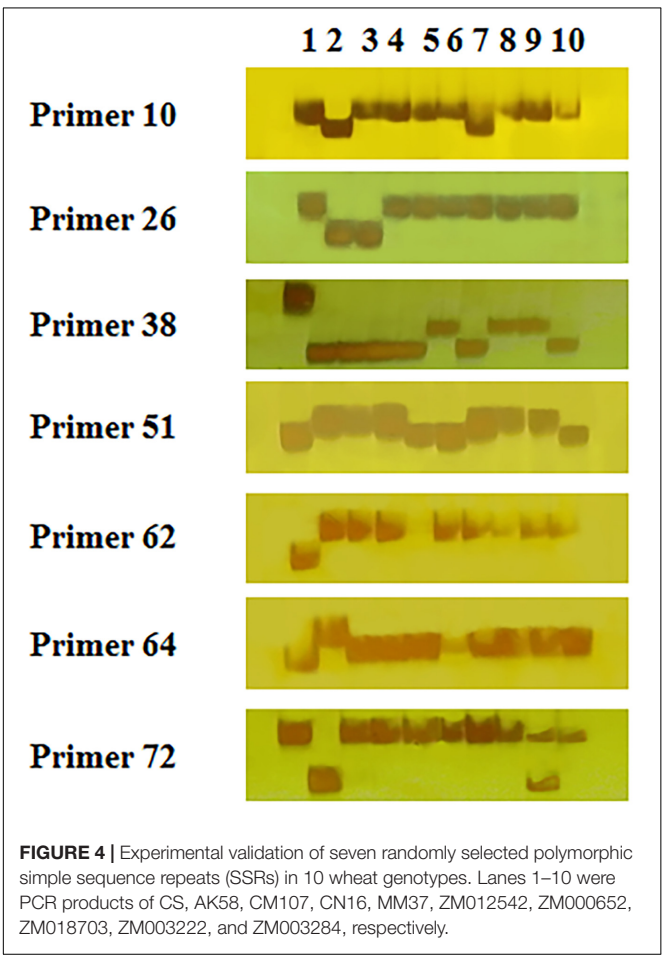


FIGURE 4 | Experimental validation of seven randomly selected polymorphic simple sequence repeats (SSRs) in 10 wheat genotypes. Lanes 1–10 were PCR products of CS, AK58, CM107, CN16, MM37, ZM012542, ZM000652, ZM018703, ZM003222, and ZM003284, respectively.

to suit the users' needs. SSRIT can completely mine SSRs (**Table 2**); however, when used only for mining a certain motif, such as tetra-nucleotide and hexa-nucleotide motifs for rice (Zhenshan97), SSRIT displayed 82.94% and 87.36% error rates, respectively (data not shown), implying a defect in the algorithm of SSRIT. In contrast, MISA and GMATA were inadequate for SSR mining. Although Kmer-SSR supported multi-threading, this support was inadequate, and this program can only be run on Linux. Furthermore, GMATA and Kmer-SSR had inflexible motif thresholds; these two programs needed to be performed in multiple tasks to identify SSRs. PERF was inflexible owing to its dependence on other modules (**Table 1**), and the computational speed was highly dependent on the motif thresholds, thus displaying a poor performance in the present tests. However, SSRMMD displayed an adequate performance in all aspects. SSRMMD completely mined credible SSRs (**Figures 3A–C**); furthermore, SSRMMD was rapid, especially for large genomes (**Table 2**); moreover, SSRMMD was flexible and did not rely on additional modules and could theoretically be run on any machine with PERL5 installed (**Table 1**).

The ever-increasing availability of plant and animal genomes and transcriptomes (Kersey et al., 2017; Marschall et al., 2018) has resulted in large data resources for developing polymorphic SSRs. In the past 3 years, certain software programs were reported for this purpose; however, they were all based on a complex pipeline and utilize numerous other software programs, increasing their dependence and decreasing their portability. For example, CandiSSR called numerous other programs during development, including MISA, Primer3, BLAST, and ClustalW (**Table 1**), among which BLAST was the most prominent reason for its low computational speed. Furthermore, the formatdb program in BLAST could not build an entire wheat genome library; hence, to complete the assessment, we artificially modified the source code of CandiSSR to enable it to normally perform computations with wheat. Similarly, GMATA depended on other programs when developing polymorphic SSRs, including e-PCR and Primer3. However, our proposed SSRMMD did not have these limitations, and SSRMMD used a stringent algorithm to assess the conservativeness and uniqueness of SSR flanking sequences. Performance assessments revealed that SSRMMD identified more novel polymorphic SSRs at extremely high speed (**Table 3** and **Figures 3D–F**).

Furthermore, we performed molecular biology assays for 80 randomly selected polymorphic SSRs of wheat to confirm the accuracy of SSRMMD, and we found that SSRMMD had an accuracy of up to 79% (**Figure 4** and **Supplementary Table 1**). We further examined 24 pairs of SSRs not yielding stable or

clear bands, and we found that 19 of them were developed through GMATA. Similarly, 7 of the 12 non-polymorphic SSRs were developed using GMATA (data not shown), indicating that these inaccurate results may have been obtained from the wheat genome itself.

Nonetheless, Gao et al. (2019) recently used SSRMMD to assess the barley genome in quantitative trait loci mapping study, and they reported that SSRMMD has an excellent algorithm for mining polymorphic SSRs.

CONCLUSION

In this study, we proposed a rapid, accurate, and flexible algorithm named SSRMMD for mining perfect SSR loci and further mining candidate polymorphic SSRs in accordance with any size of assembled sequence. Our program can easily collect numerous polymorphic SSRs from genomes and transcriptomes of diverse organisms and will undoubtedly accelerate numerous types of genetic studies including those of quantitative trait loci mapping, genotyping, and genetic diversity.

DATA AVAILABILITY STATEMENT

All datasets presented in this study are included in the article/**Supplementary Material**.

ETHICS STATEMENT

The experiments comply with the ethical standards in the country in which they were performed.

AUTHOR CONTRIBUTIONS

XG and HS conducted data analysis and drafted the manuscript. SY, ZW, CL, SL, and JM performed the validation experiment and helped with data analysis. GC helped to draft the manuscript. TL participated in the design of the study and partially revised the manuscript. YL designed and coordinated this study and revised the manuscript. All authors have read and approved the final manuscript.

FUNDING

This study was supported by the National Natural Science Foundation of China (31771794 and 91731305), the National Key Research and Development Program of China (2016YFD0101004 and 2017YFD0100900), the outstanding Youth Foundation of the Department of Science and Technology of Sichuan Province (2016JQ0040), and the International Science and Technology Cooperation Program of the Bureau of Science and Technology of Chengdu China (No. 2015DFA306002015-GH03-00008-HZ).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2020.00706/full#supplementary-material

TABLE S1 | Molecular experiments to verify the accuracy of the output by SSRMMD in wheat.

REFERENCES

Altschul, S. F., Madden, T. L., Schffer, A. A., Jinghui, Z., Zheng, Z., Webb, M., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402. doi: 10.1093/nar/25.17.3389

Avvaru, A. K., Sowpati, D. T., and Mishra, R. K. (2017). PERF: an exhaustive algorithm for ultra-fast and efficient identification of microsatellites from large DNA sequences. *Bioinformatics* 34, 943–948. doi: 10.1093/bioinformatics/btx721

Castelo, A. T., Martins, W., and Gao, G. R. (2002). TROLL–Tandem Repeat Occurrence Locator. *Bioinformatics* 18, 634–636. doi: 10.1093/bioinformatics/18.4.634

Chen, M., Tan, Z., and Zeng, G. (2011). MfSAT: detect simple sequence repeats in viral genomes. *Bioinformation* 6, 171–172. doi: 10.6026/97320630006171

Du, L. M., Zhang, C., Liu, Q., Zhang, X. Y., and Yue, B. S. (2017). Krait: an ultrafast tool for genome-wide survey of microsatellites and primer design. *Bioinformatics* 4, 681–683. doi: 10.1093/bioinformatics/btx665

Gao, S., Zheng, Z., Hu, H. Y., Shi, H. R., Ma, J., Liu, Y. X., et al. (2019). A novel QTL conferring fusarium crown rot resistance located on chromosome arm 6HL in barley. *Front. Plant Sci.* 10:1206. doi: 10.3389/fpls.2019.01206

Gramazio, P., Plesa, I. M., Truta, A. M., Sestras, A. F., and Sestras, R. E. (2018). Highly informative SSR genotyping reveals large genetic diversity and limited differentiation in European larch (*Larix decidua*) populations from Romania. *Turk. J. Agric. For.* 42, 165–175. doi: 10.3906/tar-1801-41

Guang, X. M., Xia, J. Q., Lin, J. Q., Yu, J., Wan, Q. H., and Fang, S. G. (2019). IDSSR: an efficient pipeline for identifying polymorphic microsatellites from a single genome sequence. *Int. J. Mol. Sci.* 20:3497. doi: 10.3390/ijms20143497

Kaur, S., Panesar, P. S., Bera, M. B., and Kaur, V. (2015). Simple sequence repeat markers in genetic divergence and marker-assisted selection of rice cultivars: a review. *Crit. Rev. Food Sci. Nutr.* 55, 41–49. doi: 10.1080/10408398.2011.646363

Kersey, P. J., Allen, J. E., Allot, A., Barba, M., Boddu, S., Bolt, B. J., et al. (2017). Ensembl Genomes 2018: an integrated omics infrastructure for non-vertebrate species. *Nucleic Acids Res.* 46, D802–D808. doi: 10.1093/nar/gkx1011

Levenshtein, V. (1966). Binary codes capable of correcting insertions and reversals. *Soviet Phys. Doklady* 10, 707–710.

Liu, J., Qu, J. T., Hu, K. H., Zhang, L., Li, J. W., Wu, B., et al. (2015). Development of genome wide simple sequence repeat fingerprints and highly polymorphic markers in cucumbers based on next-generation sequence data. *Plant Breed.* 134, 605–611. doi: 10.1111/pbr.12304

Liu, L., Qin, M., Yang, L., Song, Z., Luo, L., Bao, H., et al. (2016). A genome-wide analysis of simple sequence repeats in *Apis cerana* and its development as polymorphism markers. *Gene* 599, 53–59. doi: 10.1016/j.gene.2016.11.016

Liu, W. C., Xu, Y. T., Li, Z. K., Fan, J., and Yang, Y. (2019). Genome-wide mining of microsatellites in king cobra (*Ophiophagus hannah*) and cross-species development of tetranucleotide SSR markers in Chinese cobra (*Naja atra*). *Mol. Biol. Rep.* 46, 6087–6098. doi: 10.1007/s11033-019-05044-7

Marschall, T., Marz, M., Abeel, T., Dijkstra, L., Dutilh, B. E., Ghaffaari, A., et al. (2018). Computational pan-genomics: status, promises and challenges. *Brief. Bioinform.* 19, 118–135. doi: 10.1093/bib/bbw089

Metz, S., Manuel, C. J., Eva, R., Federico, G., and Patricia, A. (2016). FullSSR: microsatellite finder and primer designer. *Adv. Bioinform.* 2016, 1–4. doi: 10.1155/2016/6040124

Mudunuri, S. B., and Nagarajaram, H. A. (2007). IMEx: imperfect microsatellite extractor. *Bioinformatics* 23, 1181–1187. doi: 10.1093/bioinformatics/btm097

Nachimuthu, V. V., Muthurajan, R., Duraialaguraja, S., Sivakami, R., and Sabariappan, R. (2015). Analysis of population structure and genetic diversity in rice germplasm using SSR markers: an initiative towards association mapping of agronomic traits in *Oryza Sativa*. *Rice* 8:30. doi: 10.1186/s12284-015-0062-5

Needleman, S. B., and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.* 48, 443–453. doi: 10.1016/0022-2836(70)90057-4

Pickett, B. D., Karlinsey, S. M., Penrod, C. E., Cormier, M. J., Ebbert, M. T. W., Shiozawa, D. K., et al. (2016). SA-SSR: a suffix array-based algorithm for exhaustive and efficient SSR discovery in large genetic sequences. *Bioinformatics* 32, 2707–2709. doi: 10.1093/bioinformatics/btw298

Pickett, B. D., Miller, J. B., and Ridge, P. G. (2017). Kmer-SSR: a fast and exhaustive SSR search algorithm. *Bioinformatics* 33, 3922–3928. doi: 10.1093/bioinformatics/btx538

Qin, H., Chen, M., Yi, X., Bie, S., Zhang, C., Zhang, Y., et al. (2015). Identification of associated SSR markers for yield component and fiber quality traits based on frame map and upland cotton collections. *PLoS One* 10:e0118073. doi: 10.1371/journal.pone.0118073

Ramu, P., Kassahun, B., Senthilvel, S., Kumar, C. A., Jayashree, B., Folkertsma, R. T., et al. (2009). Exploiting rice–sorghum synteny for targeted development of EST-SSRs to enrich the sorghum genetic linkage map. *Theor. Appl. Genet.* 119, 1193–1204. doi: 10.1007/s00122-009-1120-4

Silva, L. R. D., Lopes, M. W. J., Souza, R. T. D., and Castanheira, B. D. (2015). ProGeRF: proteome and genome repeat finder utilizing a fast parallel hash function. *BioMed Res. Int.* 2015, 1–9. doi: 10.1155/2015/394157

Song, Q. J., Shi, J. R., Singh, S., Fickus, E. W., Costa, J. M., Lewis, J., et al. (2005). Development and mapping of microsatellite (SSR) markers in wheat. *Theor. Appl. Genet.* 110, 550–560. doi: 10.1007/s00122-004-1871-x

Temnykh, S. (2001). Computational and experimental analysis of microsatellites in rice (*Oryza sativa* L.) : frequency, length variation, transposon associations, and genetic marker potential. *Genome Res.* 11, 1441–1452. doi: 10.1016/j.ces.2004.03.045

Thiel, T., Michalek, W., Varshney, R., and Graner, A. (2003). Exploiting EST databases for the development and characterization of gene-derived SSR-markers in barley (*Hordeum vulgare* L.). *Theor. Appl. Genet.* 106, 411–422. doi: 10.1007/s00122-002-1031-0

Thompson, J. D., Gibson, T. J., and Higgins, D. G. (2002). Multiple sequence alignment using ClustalW and ClustalX. *Curr. Protoc. Bioinformatics* 2:3. doi: 10.1002/0471250953.bi0203s00

Untergasser, A., Cutcutache, I., Koressaar, T., Ye, J., Faircloth, B. C., Remm, M., et al. (2012). Primer3–new capabilities and interfaces. *Nucleic Acids Res.* 40:e115. doi: 10.1093/nar/gks596

Varshney, R. K., Graner, A., and Sorrells, M. E. (2005). Genic microsatellite markers in plants: features and applications. *Trends Biotechnol.* 23, 48–55. doi: 10.1016/j.tibtech.2004.11.005

Wang, L., Zhang, Y., Zhu, X., Zhu, X., and Zhang, X. (2017). Development of an SSR-based genetic map in sesame and identification of quantitative trait loci associated with charcoal rot resistance. *Sci. Rep.* 7:8349. doi: 10.1038/s41598-017-08858-2

Wang, X., Lu, P., and Luo, Z. (2013). GMATo: a novel tool for the identification and analysis of microsatellites in large genomes. *Bioinformation* 9, 541–544. doi: 10.6026/97320630009541

Wang, X., and Wang, L. (2016). GMATA: an integrated software package for genome-scale SSR mining, marker development and viewing. *Front. Plant Sci.* 7:1350. doi: 10.3389/fpls.2016.01350

Wang, X., Yang, S., Chen, Y., Zhang, S., Zhao, Q., Li, M., et al. (2018). Comparative genome-wide characterization leading to simple sequence repeat marker development for *Nicotiana*. *BMC Genomics* 19:500. doi: 10.1186/s12864-018-4878-4

Xia, E., Yao, Q., Zhang, H., Jiang, J., Zhang, L., and Gao, L. (2016). CandiSSR: an efficient pipeline used for identifying candidate polymorphic ssrs based on multiple assembled sequences. *Front. Plant Sci.* 6:1171. doi: 10.3389/fpls.2015.01171

Xu, J., Liu, L., Xu, Y., Chen, C., Rong, T., Ali, F., et al. (2013). Development and characterization of simple sequence repeat markers providing genome-wide coverage and high resolution in maize. *DNA Res.* 20, 497–509. doi: 10.1093/dnares/dst026

Yang, N., Xu, X. W., Wang, R. R., Peng, W. L., Cai, L., Song, J. M., et al. (2017). Contributions of *Zea mays* subspecies mexicana haplotypes to modern maize. *Nat. Commun.* 8:1874. doi: 10.1038/s41467-017-02063-5

Zalapa, J. E., Cuevas, H., Zhu, H., Steffan, S., Senalik, D., Zeldin, E., et al. (2012). Using next-generation sequencing approaches to isolate simple sequence repeat (SSR) loci in the plant sciences. *Am. J. Bot.* 99, 193–208. doi: 10.3732/ajb.1100394

Zhang, L., Cai, R., Yuan, M., Tao, A., Xu, J., Lin, L., et al. (2015). Genetic diversity and DNA fingerprinting in jute(*Corchorus* spp.) based on SSR markers. *Crop J.* 3, 416–422. doi: 10.1016/j.cj.2015.05.005

Zhang, Z., Deng, Y., Tan, J., Hu, S., Yu, J., and Xue, Q. (2007). A genome-wide microsatellite polymorphism database for the indica and japonica rice. *DNA Res.* 14, 37–45. doi: 10.1093/dnares/dsm005

Zhou, R., Wu, Z., Cao, X., and Jiang, F. L. (2015). Genetic diversity of cultivated and wild tomatoes revealed by morphological traits and SSR markers. *Genet. Mol. Res. GMR* 14, 13868–13879. doi: 10.4238/2015.october.29.7

Zhou, Z., Jiang, Y., Wang, Z., Gou, Z., Lyu, J., Li, W., et al. (2015). Resequencing 302 wild and cultivated accessions identifies genes related to domestication and improvement in soybean. *Nat. Biotechnol.* 4, 408–414. doi: 10.1038/nbt.3096

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

PERSPECTIVE
published: 07 August 2020
doi: 10.3389/fbioe.2020.00942

Computer-Aided Whole-Cell Design: Taking a Holistic Approach by Integrating Synthetic With Systems Biology

Lucia Marucci[1,2,3], Matteo Barberis[4,5,6]*†, Jonathan Karr[7]*†, Oliver Ray[8]*†, Paul R. Race[3,9]*†, Miguel de Souza Andrade[10,11], Claire Grierson[3,12]*, Stefan Andreas Hoffmann[13], Sophie Landon[1,3], Elibio Rech[10]*, Joshua Rees-Garbutt[3,12], Richard Seabrook[14]*, William Shaw[15] and Christopher Woods[3,16]**

[1] Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom, [2] School of Cellular and Molecular Medicine, University of Bristol, Bristol, United Kingdom, [3] Bristol Centre for Synthetic Biology (BrisSynBio), University of Bristol, Bristol, United Kingdom, [4] Systems Biology, School of Biosciences and Medicine, Faculty of Health and Medical Sciences, University of Surrey, Guildford, United Kingdom, [5] Centre for Mathematical and Computational Biology, CMCB, University of Surrey, Guildford, United Kingdom, [6] Synthetic Systems Biology and Nuclear Organization, Swammerdam Institute for Life Sciences, University of Amsterdam, Amsterdam, Netherlands, [7] Icahn Institute for Data Science and Genomic Technology, Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY, United States, [8] Department of Computer Science, University of Bristol, Bristol, United Kingdom, [9] School of Biochemistry, University of Bristol, Bristol, United Kingdom, [10] Brazilian Agricultural Research Corporation/National Institute of Science and Technology – Synthetic Biology, Brasília, Brazil, [11] Department of Cell Biology, Institute of Biological Sciences, University of Brasília, Brasília, Brazil, [12] School of Biological Sciences, University of Bristol, Bristol, United Kingdom, [13] Manchester Institute of Biotechnology, The University of Manchester, Manchester, United Kingdom, [14] Elizabeth Blackwell Institute for Health Research (EBI), University of Bristol, Bristol, United Kingdom, [15] Department of Bioengineering, Imperial College London, London, United Kingdom, [16] School of Chemistry, University of Bristol, Bristol, United Kingdom

Computer-aided design (CAD) for synthetic biology promises to accelerate the rational and robust engineering of biological systems. It requires both detailed and quantitative mathematical and experimental models of the processes to (re)design biology, and software and tools for genetic engineering and DNA assembly. Ultimately, the increased precision in the design phase will have a dramatic impact on the production of designer cells and organisms with bespoke functions and increased modularity. CAD strategies require quantitative models of cells that can capture multiscale processes and link genotypes to phenotypes. Here, we present a perspective on how whole-cell, multiscale models could transform design-build-test-learn cycles in synthetic biology. We show how these models could significantly aid in the design and learn phases while reducing experimental testing by presenting case studies spanning from genome minimization to cell-free systems. We also discuss several challenges for the realization of our vision. The possibility to describe and build whole-cells *in silico* offers an opportunity to develop increasingly automatized, precise and accessible CAD tools and strategies.

Keywords: whole-cell models, synthetic biology, systems biology, multiscale models, bioengineering, biodesign

OPEN ACCESS

Edited by:
Dong-Yup Lee, Sungkyunkwan University, South Korea

Reviewed by:
Hyun Uk Kim, Korea Advanced Institute of Science and Technology, South Korea
*Meiyappan Lakshmanan, Bioprocessing Technology Institute (A*STAR), Singapore*

****Correspondence:***
Lucia Marucci
lucia.marucci@bristol.ac.uk
Matteo Barberis
m.barberis@surrey.ac.uk; matteo@barberislab.com
Jonathan Karr
karr@mssm.edu
Oliver Ray
csxor@bristol.ac.uk
Paul R. Race
Paul.Race@bristol.ac.uk
Claire Grierson
claire.grierson@bristol.ac.uk
Elibio Rech
elibio.rech@embrapa.br
Richard Seabrook
richard.seabrook@bristol.ac.uk
Christopher Woods
Christopher.Woods@bristol.ac.uk

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *29 May 2020*
Accepted: *21 July 2020*
Published: *07 August 2020*

INTRODUCTION

Whole-cell models (WCMs) are state-of-the-art Systems Biology formalisms: they aim at representing and integrating all cellular functions within a unique computational framework, ultimately enabling a holistic, and quantitative understanding of cell biology (Tomita, 2001; Karr et al., 2015a). Quantitative and high-throughput *in silico* experiments generated from WCMs promise to significantly shorten the distance between hypothesis/design formulation and testing (Carrera and Covert, 2015).

While simplified models for specific cellular functions were first developed over 30 years ago [e.g., gene expression regulation (McAdams and Arkin, 1997), signaling (Morton-Firth and Bray, 1998) and metabolic pathways (Cornish-Bowden and Hofmeyr, 1991), cell growth (Shu and Shuler, 1989) and the cell cycle (Goldbeter, 1991; Tyson, 1991; Novak and Tyson, 1993)], the first WCM, the E-Cell model, was only derived in the 1990s for *Mycoplasma genitalium*, which has the smallest genome among freely living organisms (Tomita et al., 1999). The so-called virtual self-surviving cell (SSC) model is partially stochastic; it includes only a subset of protein-coding genes and enables dynamic simulations which encompass various subcellular processes, including enzymatic reactions, complex formation and substance translocation. In parallel, the first genome-scale metabolic models (GSMMs) were developed by Palsson's group (Varma and Palsson, 1994) using flux balance analysis (FBA) in the 1990s.

More recently, hundreds of GSMMs have been reconstructed for different organisms, with an increasing number of represented genes (McCloskey et al., 2013; Yilmaz and Walhout, 2017; Mendoza et al., 2019). GSMMs have been complemented with a mathematical description of other processes, such as transcription, translation, and signaling (Lee et al., 2008; Thiele et al., 2009). Less than a decade ago a more complete, hybrid WCM, representing all genes and molecular functions known for an organism, was reported by Covert's group (Karr et al., 2012). In this pioneering work, Karr and colleagues integrated 28 sub-models to represent one cell cycle of *M. genitalium*; each sub-model is represented with a distinct formalism, including ordinary differential equations (ODEs), FBA, stochastic simulations and Boolean rules.

Substantial research and effort are still needed to improve WCMs' descriptive power and to increase the complexity of organisms they can represent. Developing a WCM is a challenging task, which requires the collection of extensive experimental data, integration of sub-cellular models and *in silico*/*in vivo* model validation. A complete WCM should ideally integrate multiscale interactions at the cellular level (Karr et al., 2012; King et al., 2016) while accounting for the overall cellular structure (Betts and Russell, 2007), the dynamic structure of molecular interactions (Noske et al., 2008; McGuffee and Elcock, 2010; Yu et al., 2016), and the spatial compartment of the subcellular components (Ander et al., 2004; Takahashi et al., 2005; Thul et al., 2017). Ensuring an accurate representation of all of the cellular processes across organisms of increasing complexity is highly challenging (Bouhaddou et al., 2018; Singla et al., 2018; Szigeti et al., 2018). It is therefore not surprising that, to date, only the *M. genitalium* and, very recently, *E. Coli* (Macklin et al., 2020). WCMs have been released, although several other WCMs are currently under development[1]. We refer the reader to recent efforts which provide an overview of the state-of-the-art in the development of WCMs (Goldberg et al., 2018; Feig and Sugita, 2019).

Here, we focus on the enormous potential we believe WCMs have for design-build-test cycles integrating synthetic with systems biology (**Figure 1**). While the applications are diverse, they share a high degree of complexity which would require extensive trial and error experimental cycles in the absence of robust computational design algorithms based on predictive models. We conclude by considering relevant challenges that must be addressed by interdisciplinary communities to fully realize our vision, discussing future directions for integrating WCMs through synthetic and systems biology.

WHOLE-CELL DESIGN STRATEGIES IN SYNTHETIC BIOLOGY

Model Granularity of Gene Network (re)Design

Mathematical models can be instrumental for the (re)design of network circuits that recapitulate definite biological functions. Knowledge of regulatory mechanisms in biological pathways has been gained by considering living systems as a composition of functional modules, which are investigated through minimal computer models. Examples include controllable oscillators (Marucci et al., 2009; Purcell et al., 2010, 2013; Tomazou et al., 2018), circadian clocks (Gerard et al., 2009; Ananthasubramaniam et al., 2020), signaling networks (Prescott and Abel, 2017), the metabolism (Castellanos et al., 2004; Pandit et al., 2017), and transcriptional regulation (Carrera et al., 2009). Existing minimal and detailed computer models span a broad range of granularity in their biochemical details. However, one may expect that, if the core design of a minimal and a detailed model is similar, their general properties will match.

The understanding of a living organism at a system's level may be reached through decomposing it into functional modules or modular circuits (Hartwell et al., 1999; Kitano, 2002; Ravasz et al., 2002). The capability to sustain viability through autonomously generated offspring is essential. It is therefore a feature that WCMs shall account for through modeling cell division, which is intimately integrated with various layers of cellular regulation (metabolism, signaling, gene regulation, transcription, etc.). A number of minimal models have been developed for the eukaryotic cell cycle by Barberis', Tyson's and Novák's groups (Battogtokh and Tyson, 2004; Barberis et al., 2012; Gerard et al., 2013, 2015; Linke et al., 2017; Mondeel et al., 2020).

Currently, the majority of multiscale models (not WCMs) lack components able to bridge cellular networks or function (cell cycle, metabolism, signaling, gene regulation, etc.). Identification of hubs, i.e., elements with high connectivity in the cellular environment that integrate cellular networks, is a critical feature

[1] https://www.wholecell.org/

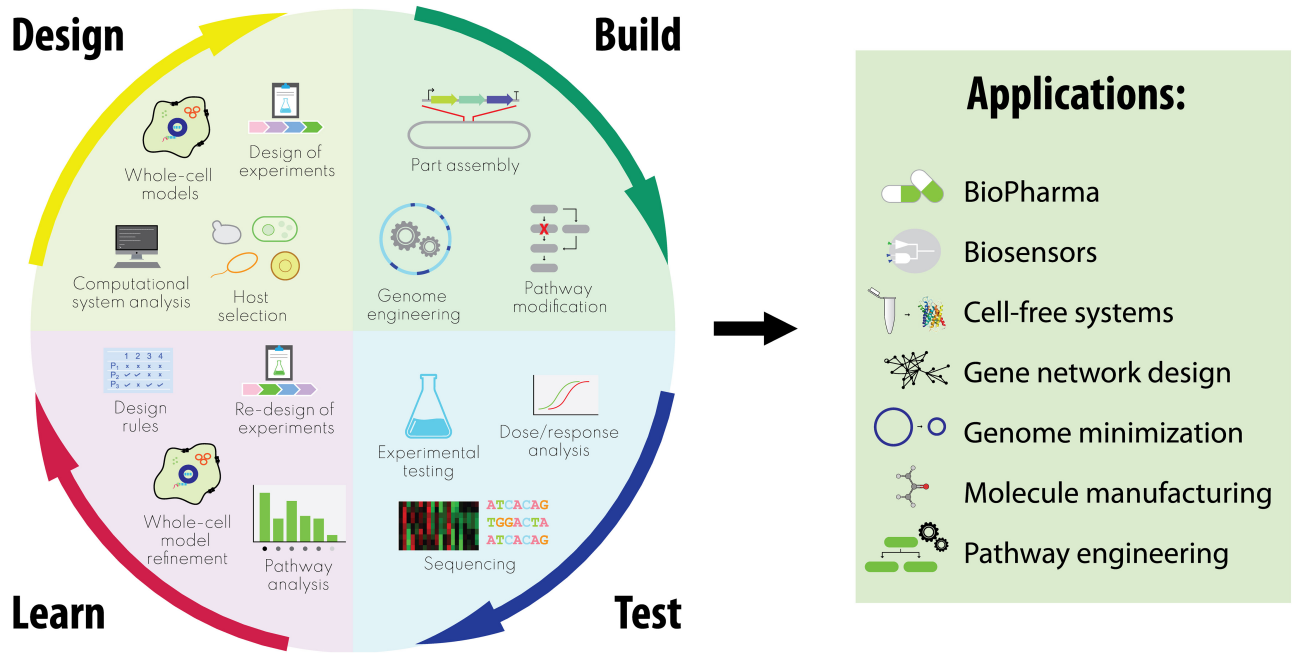


FIGURE 1 | Integrated design-build-test-learn cycles in synthetic biology encompassing whole-cell model-guided approaches, and relative applications.

of WCMs. Transcription factors have recently been identified as hubs that integrate multiscale networks, potentially connecting the cell cycle to metabolism (Mondeel et al., 2019), and can be among the parts of a system that influence its state as a whole. Multiscale frameworks coupling networks of differing granularity are being developed, by identifying the relevant regulations occurring among common network nodes and through the use of different mathematical formalisms (van der Zee and Barberis, 2019). These and other strategies are also being developed to integrate networks of cellular functional modules (Prescott et al., 2015). Together with the identification of networks underlying the cell's autonomous oscillations, these strategies can rationalize the proper timing of offspring generation accounted by WCMs.

Designing synthetic gene networks by modeling and integrating them within WCM formalisms [as in Purcell et al. (2013)] could be critical to investigate how gene expression correlates with codon usage, explore possible cell burden effects (Borkowski et al., 2016), and predict modularity of synthetic gene networks and tools to modulate gene expression across different chassis (Way et al., 2014; Pedone et al., 2019; Gomide et al., 2020).

Design and Engineering of Reduced Genomes

Minimal genomes can be defined as reduced genomes containing only the genetic material which is essential for a cell to reproduce (Glass et al., 2017). Studying and engineering minimal genomes can be instrumental both to understand the most essential tasks a cell must perform to sustain life, and to obtain optimal chassis for synthetic biology applications, with reduced cell burden and superior robustness (Moya et al., 2009; Hutchison et al., 2016; Ceroni and Ellis, 2018; Mol et al., 2018; Landon et al., 2019).

Exhaustive experimental characterization of a minimized genome is unfeasible: even for an organism as small as *M. genitalium* (0.58 mb and 525 genes), there are thousands of possible combinations of gene knockouts to be performed. Of note, this figure is most probably underestimated, accounting for the fact that the order in which gene deletions are performed can alter the resulting phenotypes (Gawand et al., 2015). Genome-scale computational models of cells could be instrumental to fully understand the dynamic and context-dependent nature of gene essentiality (Rancati et al., 2018), and to rationally design minimized genomes *in silico*. Computer-aided minimal genome engineering could significantly reduce the time and cost to reduce genomes compared to current approaches based on extensive experimental iterations (Posfai et al., 2006; Iwadate et al., 2011; Hirokawa et al., 2013; Hutchison et al., 2016; Zhou et al., 2016; Reuss et al., 2017; Breuer et al., 2019).

To the best of our knowledge, two top-down genome reduction approaches have been proposed so far based on genome-scale models. The MinGenome algorithm applies a mixed-integer linear programming (MILP) algorithm to a GSMM of *Escherichia coli*, using information pertaining to essential genes and synthetic lethal pairs within the optimization (Wang and Maranas, 2018). In contrast, Minesweeper and GAMA are top-down genome minimization algorithms based on the *M. genitalium* WCM. They exploit a divide-and-conquer approach and a biased genetic algorithm, respectively, to iteratively simulate reduced genomes (Rees-Garbutt et al., 2020); their *in silico* predictions have not been tested in the laboratory yet.

GSMM-based genome reduction algorithms such as MinGenome or analogous, adaptable metaheuristic techniques [e.g., (Burgard et al., 2003; Tang et al., 2015; Mutturi, 2017)] are currently more broadly applicable across organisms given the large availability of these formalisms. Still, as more WCMs become available, we expect WCM-based genome reduction algorithms to provide superior predictions of cellular processes and genetic interactions, thanks to their richness of multiscale cellular process representation.

Design and Prototyping of Cell-Free Systems

Cell-free transcription/translation systems, based on crude cellular extracts, are a valuable platform to address fundamental biological questions in a controllable and reproducible way. In recent years, the decrease of costs associated with this technology and significant improvements in synthesis yield capabilities (Calhoun and Swartz, 2005) have made cell-free systems increasingly popular in synthetic biology for the prototyping and testing of engineered biological parts (McCloskey et al., 2013; Reuss et al., 2017; Yilmaz and Walhout, 2017; Mendoza et al., 2019) and networks (Noireaux et al., 2003; Siegal-Gaskins et al., 2014; Takahashi et al., 2015). As the possible applications of cell-free systems grow [see (Silverman et al., 2020) for a recent review], mathematical models are being developed to quantitatively formalize how biological processes perform within cell-free platforms (Koch et al., 2018).

So far, deterministic models (ODEs, or constraint-based) have been proposed to describe specific processes within cell-free platforms such as transcription and translation (Karzbrun et al., 2011; Stogbauer et al., 2012; Siegal-Gaskins et al., 2014), resource competition (Underwood et al., 2005; Borkowski et al., 2018; Matsuura et al., 2018; Moore et al., 2018), and metabolism (Vilkhovoy et al., 2018). The integration of mathematical formalisms across scales for cell-free platforms, building toward WCMs, could be highly beneficial to both facilitate *de novo* design of circuits, and to quantitatively compare *in vitro* cell-free products with their *in vivo* counterparts.

Whole-Cell Biosensor Design and Testing

Biosensors are analytical devices which can convert a biochemical reaction into a measurable signal. The recognition unit in a biosensor can be composed of whole cells, nucleic acids, enzymes, proteins, antibodies or combinations thereof. Synthetic biology has significantly accelerated biosensor development; new generation whole-cell biosensors (i.e., sensors implemented throughout living cells) have been engineered, allowing, for example: arsenic detection (Diesel et al., 2009), detection of pollutants and antibiotics (van der Meer and Belkin, 2010), microbial detection in industrial settings (Lu et al., 2013) and *in vivo* diagnostic applications [e.g., detection of environmental signals in the gut (Kotula et al., 2014) and diagnosis of liver metastases (Danino et al., 2015); see (Slomovic et al., 2015) for an overview].

The application of WCMs to the design, prototyping and testing of whole-cell biosensors could suggest rational approaches to tune their sensitivity, stability, and dynamic range while facilitating the choice of the ideal chassis and, if needed, guide its re-engineering to optimize biosensor performance (Hicks et al., 2020). If WCMs become available for different chassis and entire organisms, they could also support the design of optimized targeted delivery of genetically encoded biosensors.

Industrial Implications of Whole-Cell Models

Although the intellectual merit of pursuing a computer-aided whole-cell design approach is unquestioned, it is clear that the success of this endeavor will ultimately be judged by its impact on science, medicine, and industry. The increasing drive of computer-aided designs (CADs) toward "green" chemistry approaches, allied to increases in gene synthesis speed and capability and associated cost reductions, are making biosynthesis an increasingly appealing route for the manufacture of high-value chemicals (El Karoui et al., 2019). This includes a plethora of opportunities across the pharmaceutical, agrochemical, commodity chemical, and materials sectors, amongst others.

A major challenge, however, remains the development of robust, scalable microbial chassis, whose metabolic processes can be predictably tuned for a desired outcome (Xu et al., 2020). Currently, chassis choice is largely restricted to a subset of genetically tractable microorganisms, whose physiology and performance during fermentation are well understood, and for whom effective molecular genetic tools required for their manipulation exist. Chassis optimization to date has relied exclusively on incremental, stepwise improvements in desired host strain characteristics, including growth rate, feedstock utilization, and product yield (Calero and Nikel, 2019). For these reasons, the process of chassis optimization remains prohibitively slow and expensive, accounting in part for the paucity of high-value small molecules that are currently manufactured using synthetic biology processes. Targeted manipulations often lead to unanticipated off-target effects, linked to the co-dependency of metabolic processes, which generally function in concert within interdependent cellular networks (Woolston et al., 2013): perturbations may compromise rather than enhance desirable characteristics, leading to undesired outcomes. Clearly, robust, predictable WCMs represent an attractive solution to the problem of chassis optimization, affording a catch-all tool that can be used to unpick dependencies and ensure that performance criteria can be met.

Additionally, the complexities associated with population heterogeneity during chassis fermentation must be resolved (Danchin, 2012). For fermentation-based industrial processes to be tractable, product yields must be sufficiently high to make biosynthesis financially viable. The emergence of "cheaters" or slow-growers within microbial populations should be tackled

with tunable regulatory processes that operate throughout populations. The introduction of such characteristics is a major challenge to conventional chassis design approaches. WCM-driven approaches could more easily implement and test these processes.

Critical to the success of a computer-aided whole-cell design approach is the quality of the employed model (Fernandez-Castane et al., 2014). Microbial systems with small genomes represent a compelling entry point for study, with model development possibly being facilitated by ongoing studies focused on establishing the core constituents of a functional genome. These studies are in part driven by genome minimization experiments, which in turn can be used to further refine model performance. Importantly, fundamental gaps remain in our understanding of microbial metabolic processes, and this will unquestionably hinder progress (Price et al., 2018). However, the capacity of WCMs to predict previously unidentified metabolic dependencies should be viewed as an acid test of model validity. Indeed, GSMMs often fail due to their inability to account for metabolic dependencies, a feature which has led to skepticism within industrial circles, questioning the value of such models. Whole-cell approaches offer a mechanism to circumvent this issue. This is of particular significance when developing chassis for "non-natural" products whose chemistries sit outside those of metabolites found in nature (Calero and Nikel, 2019). Expanding the metabolic capacity of chassis organisms to deliver such novel products risks introducing additional complexities, including excessive depletion of core metabolite pools or the generation of toxic products or intermediates. Design approaches driven by WCMs are uniquely placed to identify such issues and provide a route to their circumvention.

The capacity to design-in explicit control over cellular behavior is also critical for industrial adoption of model-derived chassis. It can be argued that the ability to regulate cellular processes is as important as defining the processes themselves. Tunable regulatory systems must afford a degree of both intrinsic and extrinsic control. Synthetic biology-based approaches for constructing genetic circuitry are now placing us on a path to broad-reaching cellular regulation, though issues still exist. These systems are often insufficiently orthogonal, with bespoke designs required for different chassis due to variations in core metabolic process (Pandit et al., 2017). Again, whole-cell design approaches offer a solution to this issue, as such systems can be predefined and tested for functionality *in silico* prior to undertaking costly lab experimentation.

WHAT'S NEXT? GOING BEYOND THE PROTOTYPE

In recent years, advances in genomic measurement technologies for data generation, the establishment of data repositories, and the development of WCM simulation platforms have significantly facilitated the derivation of WCMs [see (Goldberg et al., 2018) for a review]. Nevertheless, the implementation of WCM-based design-build-test cycles for genome-scale engineering requires further challenges to be addressed (Bartley et al., 2020).

If a model has to be used for the design and prototyping of an engineered living system, the model needs to be reliable. Even for a simple organism, the number of kinetic parameters raises as the complexity and the level of detail of a mathematical model increase; constraining parameters thus becomes harder and requires extensive experimental data. Mathematical models can be used to produce predictions of missing data, however, they often abstract physical processes using simplifying assumptions which might hold in specific conditions (Babtie and Stumpf, 2017). To set the 1,462 quantitative parameters of the *M. genitalium* WCM, values from related organisms were incorporated due to a lack of organism-specific data (Macklin et al., 2014); a combination of parameter values reported from previous experiments and numerical optimization on a reduced model was performed. While, ideally, we would like to measure all kinetic parameters directly from experiments, we still lack the ability to measure each state in individual cells over time, and across all possible environmental conditions. A combination of direct experimental estimation and parameter inference will likely be needed for genome-scale formalisms and WCMs.

Sensitivity analysis, usually performed by perturbing parameters to understand how uncertainties affect the model outputs (Erguler and Stumpf, 2011), can become extremely computationally expensive when applied to genome-scale models. Alternatively, statistical approaches such as those based on Bayesian methods (Vernon et al., 2018) or the Fisher information matrix (Rand, 2008) could be carefully carried out at least at the sub-model level, and possibly scaled up to WCMs. The Reverse Engineering Assessments and Methods (DREAM8) parameter estimation challenge (Karr et al., 2015b) was organized to develop new parameter estimation techniques specific for WCMs. It suggested possible interesting avenues for WCM parameterization (i.e., model reduction and a combination of differential evolution and random forests), and highlighted that the availability of comprehensive data is critical to ensure the model is practically identifiable (Ashyraliyev et al., 2009), and to calibrate WCMs.

Researchers have started to collect data needed for WCM development into public repositories [e.g., (Wittig et al., 2012; Kolesnikov et al., 2015; Sajed et al., 2016; UniProt Consortium, 2018; Caspi et al., 2020)]; still, the data needed to derive and fit WCMs are dispersed across many repositories and publications and often not annotated or normalized, ultimately requiring a massive manual effort. Federated archives of repositories, such as the PDB-Dev system to deposit Integrative/Hybrid models and corresponding data (Burley et al., 2017), also exist and might be well placed to archive and disseminate both data and models, while enabling different researchers to attempt alternative modeling/parameterization approaches. Covert's group developed the WholeCellKB database (Karr et al., 2013) to organize the quantitative measurements (over 1,400) from which the *M. genitalium* WCM was derived; it would be ideal to enable automatic access and querying in such databases.

To enhance WCM reproducibility and collaboration, new standards and simulations software are also needed (Medley et al., 2016). Researchers should invest efforts to use and expand the capabilities of standard formats such as the Systems Biology Markup Language (SBML) (Hucka et al., 2003) and the Systems Biology Graphical Notation (SBGN) (Le Novere et al., 2009) to be suitable for WCMs. For example, several aspects of the *M. genitalium* WCM cannot be represented by SBML, such as the multi-algorithmic nature of the model (Waltemath et al., 2016). Further development of standard modeling formats is needed to enable reproducible WCM simulations, e.g., by including in the SMBL Hierarchical Model Composition package ontologies which could represent the algorithm needed for specific sub-models (Courtot et al., 2011). In the context of synthetic biology applications, we believe it would be appropriate and beneficial to report and deposit data related to various iterations of WCM-generated *in silico* predictions, *in vivo* testing and possible model/design refinement; this would establish the predictive power of WCMs and illuminate steps to make design-build-test-learn cycles more effective.

It is also important to consider the structural uncertainties in the model, which depend on model assumptions. While, for certain sets of models (e.g., small ODE systems for signaling pathways), likelihood- and Bayesian-based approaches have been proposed for model selection (Wilkinson, 2007; Kirk et al., 2013) and semidefinite programming for model invalidation (Anderson and Papachristodoulou, 2009), no suitable techniques for WCMs have been proposed to date.

We foresee that automation will play a fundamental role in the derivation of WCMs for eukaryotic organisms and in their application to design complex processes. Ideally, we would like to introduce automation at different stages, such as data extraction from the literature, model derivation, and model/data integration both within the model fitting and validation steps, and when comparing *in silico* design prediction with *in vivo* tests (Bartley et al., 2020). This, in turn, will require the adoption of standards for both data and model repositories. Also, laboratory automation, coupled to WCM-based CAD, is expected to transform design-build-test cycles. As the use of robotics becomes increasingly common in both academia and industry, the throughput and reproducibility of experiments needed for both WCM derivation and validation can be significantly increased, and protocol sharing across research communities facilitated (Jessop-Fabre and Sonnenschein, 2019).

To assist the adoption of WCMs for synthetic biology applications, high-performance parallelized computer clusters are required to run the models with lengthy runtimes, coordinate the corresponding databases, parameterize and validate the models, and then to integrate WCMs in design cycles in combination with optimization algorithms (Macklin et al., 2014; Chalkley et al., 2019).

The implementation of standardized tools to share data and simulate WCMs would, in turn, facilitate model validation. This should involve the definition of proper metrics and formal model verification techniques such as those developed for SBML-encoded models (Kwiatkowska et al., 2011).

(RE)THINKING SYSTEM APPROACHES: A COLLABORATIVE EFFORT

In addressing the aforementioned challenges, we believe there is a tremendous opportunity to rethink approaches used so far to generate genome-scale models, including WCMs, and to integrate with broader communities including software engineers, computer scientists, structural biologists, bioinformaticians, and systems and synthetic biologists.

We do anticipate that, as diverse communities synergize on WCM-related research, different kinds of formalisms might be integrated within genome-scale models. Symbolic reasoning provides a range of expressive and intuitive logical frameworks that could potentially complement and help glue together sub-models at different scales. Such methods are routinely applied to complex systems in the electronics and software industries, and have been applied to biological systems for nearly a decade (Iyengar, 2011). Recent work showed the feasibility of applying logic programming methods to signaling pathways (Ray et al., 2011), metabolic networks (Bragagli and Ray, 2015) and automating a mechanistic philosophy of scientific discovery in simulated organisms (Rozanski et al., 2015); it should be feasible to integrate such sub-models within a WCM framework.

We believe there is scope to further increase the descriptive and predictive ability of WCMs across spatial and temporal scales by integrating the structural biology and the molecular modeling communities to carefully consider not only the biochemical, but also the physical, molecular and structural components of cells. The development of the so-called "physical" WCMs [see (Feig and Sugita, 2019) and (Feig and Sugita, 2013) for comprehensive reviews] is an emerging field, with the first models describing minimal cellular environments in full atomistic detail (Feig et al., 2015; Yu et al., 2016). With the final aim to integrate biochemical and physical WCMs within a multiscale framework (Sali et al., 2015), we need approaches which can cope with the limitations of atomistic models of biomolecules (mainly in terms of computational resources), possibly exploiting coarse-grained (Ando and Skolnick, 2010; Hyeon and Thirumalai, 2011) or continuum (Solernou et al., 2018) approaches.

By collaborating with software engineers, we need to develop tools which can enable, and possibly automate, the integration of different data types across scales, model derivation, fitting and validation, and visualization and interpretation of results (Szigeti et al., 2018).

Moreover, rule-based models might become the new standard to represent each molecular species with the required level of granularity and multi-algorithmic sub-models (e.g., FBA and stochastic dynamical models). Frameworks where intuitive logic is coupled to rule-based models have started to be developed recently (van der Zee and Barberis, 2019).

As we produce ever-increasing amounts of experimental data and increasingly sophisticated computational tools to realize detailed and complex representations of actual cells, approaches instead focusing on deliberately abstract and parsimonious simulations of artificial cellular systems provide

a valuable change of perspective. Such "toy models" might be a valuable tool to test different algorithms for model derivation and fitting, while offering an opportunity to engage with broader research communities and with the public (Castiglione et al., 2014).

Finally, we believe there is tremendous potential for applying machine learning techniques to both WCM derivation and their applications in synthetic biology. Two recent works (Lin et al., 2017; Ma et al., 2018) showed that deep neural networks are well placed to reconstruct the architecture of living systems [namely, the hierarchical organization of nuclear transcriptional factors in the nucleus (Lin et al., 2017) and of a basic eukaryotic cell (Ma et al., 2018)] and predict cell states and phenotypes. In both cases, the configuration of network layers and thus the biological structure were formulated using extensive prior knowledge, ultimately enabling fully "visible" systems, where all the internal biological states can be interrogated mechanistically (Yu et al., 2018). Machine learning could be beneficial to systematically process large *in vivo* and *in silico* whole-cell datasets, for example by applying Bayesian inference, to integrate data from diverse sources and supplement sparse data (Perdikaris and Karniadakis, 2016), and to help to automatically classify WCM simulations and link phenotypes to genotypes (Alber et al., 2019). Ensemble methods, which combine multiple independent models into a single predictive model for increasing the overall robustness of predictions, might also be adopted to develop subcellular formalisms and support their integration across chassis (Camacho et al., 2018). Additionally, machine learning might assist in WCM parameter identification, for example applying Bayesian parameter estimation (Vyshemirsky and Girolami, 2008), regression models and reinforcement learning techniques (Alber et al., 2019). Optimal experimental design techniques might also offer a valuable methodology to select the best experimental datasets for both model identification and validation (Smucker et al., 2018).

DISCUSSION

We have shown that WCMs are likely to be instrumental to inform design-build-test cycles across synthetic biology applications. WCMs can accelerate the realization of "designer" cells and organisms tailored to specific functions, reducing experimental iterations and increasing the predictive power of computational formalisms used so far.

In the (re)design of cellular network functionalities, it is therefore important to quantitatively analyze and predict, through dedicated modeling strategies, the dynamics of interactions between various layers of cellular regulation. Thus, WCMs should take into account how different cellular layers are integrated, and how regulatory feedback among these layers occurs in time. These challenges must be tackled through integrative computational and experimental collaborative efforts aimed, respectively, toward: (i) engineering *in vivo* network designs which, through predictive systems biology, may be able to autonomously oscillate, sustaining generation of offspring, and (ii) extraction, visualization and functional exploration of regulatory interactions among cellular layers through novel multiscale modeling frameworks.

As synthetic biology moves toward the (re)engineering of entire genomes and multicellular systems, interdisciplinary communities need to collaborate for the development of tools that are required to improve the predictive power of WCMs. Although challenges remain, it is clear that the adoption of model-based methods has the potential to transform both basic research and the current bioproduction development process, leading to marked improvements in host performance and product yield on an industrial scale.

Ultimately, as the development of human genome-scale kinetic models becomes more feasible (Bordbar et al., 2015; Szigeti et al., 2018), we anticipate that whole-cell formalisms will become an indispensable tool to study human variation, and design treatments and synthetic cellular screening systems.

AUTHOR CONTRIBUTIONS

LM, MB, JK, OR, and PR wrote the manuscript. MS prepared the figure. All other authors participated to discussion within the workshop, helped with editing, and/or provided feedback.

FUNDING

LM was funded by the Engineering and Physical Sciences Research Council (EPSRC, grants EP/R041695/1 and EP/S01876X/1) and Horizon 2020 (CosyBio, grant agreement 766840); MB was funded by the Systems Biology Grant of the University of Surrey; JK was funded by the National Institutes of Health (award R35GM119771); PR was funded by the EPSRC (EP/R020957/1) and the Biotechnology and Biological Sciences Research Council (BBSRC, BB/T001968/1); CG, LM, and PR were funded by the BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre (BB/L01386X/); SL and JR-G were funded by the EPSRC Future Opportunity Ph.D. scholarships; ER was funded by the INCT BioSyn (National Institute of Science and Technology in Synthetic Biology), CNPq (National Council for Scientific and Technological Development), CAPES (Coordination for the Improvement of Higher Education Personnel), Brazilian Ministry of Health, and FAPDF (Research Support Foundation of the Federal District), Brazil.

ACKNOWLEDGMENTS

This work captures discussions between participants at the "Computer-Aided Whole-Cell Design and Engineering" Workshop held on the 02-03 July 2019 at the University of Bristol, United Kingdom, and funded by the Engineering and Physical Sciences Research Council (EPSRC) within the remits of the Big Ideas initiative. We sincerely thank Dr. Kathleen Sedgley for her support with the workshop organization, and Dr. Thomas Gorochowski for participating in discussions.

REFERENCES

Alber, M., Buganza Tepole, A., Cannon, W. R., De, S., Dura-Bernal, S., Garikipati, K., et al. (2019). Integrating machine learning and multiscale modeling-perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digit. Med.* 2:115.

Ananthasubramaniam, B., Schmal, C., and Herzel, H. (2020). Amplitude effects allow short jet lags and large seasonal phase shifts in minimal clock models. *J. Mol. Biol.* 432, 3722–3737. doi: 10.1016/j.jmb.2020.01.014

Ander, M., Beltrao, P., Di Ventura, B., Ferkinghoff-Borg, J., Foglierini, M., Kaplan, A., et al. (2004). SmartCell, a framework to simulate cellular processes that combines stochastic approximation with diffusion and localisation: analysis of simple networks. *Syst. Biol.* 1, 129–138. doi: 10.1049/sb:20045017

Anderson, J., and Papachristodoulou, A. (2009). On validation and invalidation of biological models. *BMC Bioinform.* 10:132. doi: 10.1186/s12918-017-0484-132

Ando, T., and Skolnick, J. (2010). Crowding and hydrodynamic interactions likely dominate in vivo macromolecular motion. *Proc. Natl. Acad. Sci. U.S.A.* 107, 18457–18462. doi: 10.1073/pnas.1011354107

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. A., and Blom, J. G. (2009). Systems biology: parameter estimation for biochemical models. *FEBS J.* 276, 886–902. doi: 10.1111/j.1742-4658.2008.06844.x

Babtie, A. C., and Stumpf, M. P. H. (2017). How to deal with parameters for whole-cell modelling. *J. R. Soc. Interf.* 14:237.

Barberis, M., Linke, C., Adrover, M. A., Gonzalez-Novo, A., Lehrach, H., Krobitsch, S., et al. (2012). Sic1 plays a role in timing and oscillatory behaviour of B-type cyclins. *Biotechnol. Adv.* 30, 108–130. doi: 10.1016/j.biotechadv.2011.09.004

Bartley, B. A., Beal, J., Karr, J. R., and Strychalski, E. A. (2020). Organizing genome engineering for the gigabase scale. *Nat. Commun.* 11:689.

Battogtokh, D., and Tyson, J. J. (2004). Bifurcation analysis of a model of the budding yeast cell cycle. *Chaos* 14, 653–661. doi: 10.1063/1.1780011

Betts, M. J., and Russell, R. B. (2007). The hard cell: from proteomics to a whole cell model. *FEBS Lett.* 581, 2870–2876. doi: 10.1016/j.febslet.2007.05.062

Bordbar, A., McCloskey, D., Zielinski, D. C., Sonnenschein, N., Jamshidi, N., and Palsson, B. O. (2015). Personalized whole-cell kinetic models of metabolism for discovery in genomics and pharmacodynamics. *Cell Syst.* 1, 283–292. doi: 10.1016/j.cels.2015.10.003

Borkowski, O., Bricio, C., Murgiano, M., Rothschild-Mancinelli, B., Stan, G. B., and Ellis, T. (2018). Cell-free prediction of protein expression costs for growing cells. *Nat. Commun.* 9:1457.

Borkowski, O., Ceroni, F., Stan, G. B., and Ellis, T. (2016). Overloaded and stressed: whole-cell considerations for bacterial synthetic biology. *Curr. Opin. Microbiol.* 33, 123–130. doi: 10.1016/j.mib.2016.07.009

Bouhaddou, M., Barrette, A. M., Stern, A. D., Koch, R. J., DiStefano, M. S., Riesel, E. A., et al. (2018). A mechanistic pan-cancer pathway model informed by multi-omics data interprets stochastic cell fate responses to drugs and mitogens. *PLoS Comput. Biol.* 14:e1005985. doi: 10.1371/journal.pcbi.1005985

Bragagli, S., and Ray, O. (2015). "Nonmonotonic learning in large biological networks," in *Inductive Logic Programming. Lecture Notes in Computer Science*, Vol. 9046, eds J. Davis and J. Ramon (Cham: Springer).

Breuer, M., Earnest, T. M., Merryman, C., Wise, K. S., Sun, L., Lynott, M. R., et al. (2019). Essential metabolism for a minimal cell. *eLife* 8:e36842.

Burgard, A. P., Pharkya, P., and Maranas, C. D. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol. Bioeng.* 84, 647–657. doi: 10.1002/bit.10803

Burley, S. K., Kurisu, G., Markley, J. L., Nakamura, H., Velankar, S., Berman, H. M., et al. (2017). PDB-Dev: a prototype system for depositing integrative/hybrid structural models. *Structure* 25, 1317–1318. doi: 10.1016/j.str.2017.08.001

Calero, P., and Nikel, P. I. (2019). Chasing bacterial chassis for metabolic engineering: a perspective review from classical to non-traditional microorganisms. *Microb. Biotechnol.* 12, 98–124. doi: 10.1111/1751-7915.13292

Calhoun, K. A., and Swartz, J. R. (2005). Energizing cell-free protein synthesis with glucose metabolism. *Biotechnol. Bioeng.* 90, 606–613. doi: 10.1002/bit.20449

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C., and Collins, J. J. (2018). Next-Generation machine learning for biological networks. *Cell* 173, 1581–1592. doi: 10.1016/j.cell.2018.05.015

Carrera, J., and Covert, M. W. (2015). Why build whole-cell models? *Trends Cell Biol.* 25, 719–722. doi: 10.1016/j.tcb.2015.09.004

Carrera, J., Rodrigo, G., and Jaramillo, A. (2009). Model-based redesign of global transcription regulation. *Nucleic Acids Res.* 37:e38. doi: 10.1093/nar/gkp022

Caspi, R., Billington, R., Keseler, I. M., Kothari, A., Krummenacker, M., Midford, P. E., et al. (2020). The MetaCyc database of metabolic pathways and enzymes - a 2019 update. *Nucleic Acids Res.* 48, D445–D453.

Castellanos, M., Wilson, D. B., and Shuler, M. L. (2004). A modular minimal cell model: purine and pyrimidine transport and metabolism. *Proc. Natl. Acad. Sci. U.S.A.* 101, 6681–6686. doi: 10.1073/pnas.0400962101

Castiglione, F., Pappalardo, F., Bianca, C., Russo, G., and Motta, S. (2014). Modeling biology spanning different scales: an open challenge. *Biomed. Res. Int.* 2014:902545.

Ceroni, F., and Ellis, T. (2018). The challenges facing synthetic biology in eukaryotes. *Nat. Rev. Mol. Cell Biol.* 19, 481–482. doi: 10.1038/s41580-018-0013-2

Chalkley, O., Purcell, O., Grierson, C., and Marucci, L. (2019). The genome design suite: enabling massive in-silico experiments to design genomes. *bioRxiv* [Preprint]. doi: 10.1101/681270

Cornish-Bowden, A., and Hofmeyr, J. H. (1991). MetaModel: a program for modelling and control analysis of metabolic pathways on the IBM PC and compatibles. *Comput. Appl. Biosci.* 7, 89–93. doi: 10.1093/bioinformatics/7.1.89

Courtot, M., Juty, N., Knupfer, C., Waltemath, D., Zhukova, A., Drager, A., et al. (2011). Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.* 7:543. doi: 10.1038/msb.2011.77

Danchin, A. (2012). Scaling up synthetic biology: do not forget the chassis. *FEBS Lett.* 586, 2129–2137. doi: 10.1016/j.febslet.2011.12.024

Danino, T., Prindle, A., Kwong, G. A., Skalak, M., Li, H., Allen, K., et al. (2015). Programmable probiotics for detection of cancer in urine. *Sci. Transl. Med.* 7:289ra84. doi: 10.1126/scitranslmed.aaa3519

Diesel, E., Schreiber, M., and van der Meer, J. R. (2009). Development of bacteria-based bioassays for arsenic detection in natural waters. *Anal. Bioanal. Chem.* 394, 687–693. doi: 10.1007/s00216-009-2785-x

El Karoui, M., Hoyos-Flight, M., and Fletcher, L. (2019). Future trends in synthetic biology-a report. *Front. Bioeng. Biotechnol.* 7:175. doi: 10.3389/fbioe.2018.00175

Erguler, K., and Stumpf, M. P. (2011). Practical limits for reverse engineering of dynamical systems: a statistical analysis of sensitivity and parameter inferability in systems biology models. *Mol. Biosyst.* 7, 1593–1602.

Feig, M., Harada, R., Mori, T., Yu, I., Takahashi, K., and Sugita, Y. (2015). Complete atomistic model of a bacterial cytoplasm for integrating physics, biochemistry, and systems biology. *J. Mol. Graph. Model.* 58, 1–9. doi: 10.1016/j.jmgm.2015.02.004

Feig, M., and Sugita, Y. (2013). Reaching new levels of realism in modeling biological macromolecules in cellular environments. *J. Mol. Graph. Model.* 45, 144–156. doi: 10.1016/j.jmgm.2013.08.017

Feig, M., and Sugita, Y. (2019). Whole-cell models and simulations in molecular detail. *Annu. Rev. Cell Dev. Biol.* 35, 191–211. doi: 10.1146/annurev-cellbio-100617-062542

Fernandez-Castane, A., Feher, T., Carbonell, P., Pauthenier, C., and Faulon, J. L. (2014). Computer-aided design for metabolic engineering. *J. Biotechnol.* 192(Pt B), 302–313.

Gawand, P., Said Abukar, F., Venayak, N., Partow, S., Motter, A. E., and Mahadevan, R. (2015). Sub-optimal phenotypes of double-knockout mutants of *Escherichia coli* depend on the order of gene deletions. *Integr. Biol.* 7, 930–939. doi: 10.1039/c5ib00096c

Gerard, C., Gonze, D., and Goldbeter, A. (2009). Dependence of the period on the rate of protein degradation in minimal models for circadian oscillations. *Philos. Trans. A Math. Phys. Eng. Sci.* 367, 4665–4683. doi: 10.1098/rsta.2009.0133

Gerard, C., Tyson, J. J., Coudreuse, D., and Novak, B. (2015). Cell cycle control by a minimal Cdk network. *PLoS Comput. Biol.* 11:e1004056. doi: 10.1371/journal.pone.0004056

Gerard, C., Tyson, J. J., and Novak, B. (2013). Minimal models for cell-cycle control based on competitive inhibition and multisite phosphorylations of Cdk substrates. *Biophys. J.* 104, 1367–1379. doi: 10.1016/j.bpj.2013.02.012

Glass, J. I., Merryman, C., Wise, K. S., Hutchison, C. A. III, and Smith, H. O. (2017). Minimal Cells-Real and imagined. *Cold Spring Harb. Perspect. Biol.* 9:a023861. doi: 10.1101/cshperspect.a023861

Goldberg, A. P., Szigeti, B., Chew, Y. H., Sekar, J. A., Roth, Y. D., and Karr, J. R. (2018). Emerging whole-cell modeling principles and methods. *Curr. Opin. Biotechnol.* 51, 97–102. doi: 10.1016/j.copbio.2017.12.013

Goldbeter, A. (1991). A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. *Proc. Natl. Acad. Sci. U.S.A.* 88, 9107–9111. doi: 10.1073/pnas.88.20.9107

Gomide, M. S., Sales, T. T., Barros, L. R. C., Limia, C. G., de Oliveira, M. A., Florentino, L. H., et al. (2020). Genetic switches designed for eukaryotic cells and controlled by serine integrases. *Commun. Biol.* 3:255.

Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From molecular to modular cell biology. *Nature* 402(Suppl.), C47–C52.

Hicks, M., Bachmann, T. T., and Wang, B. (2020). Synthetic biology enables programmable cell-based biosensors. *Chemphyschem* 21:131. doi: 10.1002/cphc.201901191

Hirokawa, Y., Kawano, H., Tanaka-Masuda, K., Nakamura, N., Nakagawa, A., Ito, M., et al. (2013). Genetic manipulations restored the growth fitness of reduced-genome *Escherichia coli*. *J. Biosci. Bioeng.* 116, 52–58. doi: 10.1016/j.jbiosc.2013.01.010

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.

Hutchison, C. A. III, Chuang, R. Y., Noskov, V. N., Assad-Garcia, N., Deerinck, T. J., Ellisman, M. H., et al. (2016). Design and synthesis of a minimal bacterial genome. *Science* 351:aad6253.

Hyeon, C., and Thirumalai, D. (2011). Capturing the essence of folding and functions of biomolecules using coarse-grained models. *Nat. Commun.* 2:487.

Iwadate, Y., Honda, H., Sato, H., Hashimoto, M., and Kato, J. (2011). Oxidative stress sensitivity of engineered *Escherichia coli* cells with a reduced genome. *FEMS Microbiol. Lett.* 322, 25–33. doi: 10.1111/j.1574-6968.2011.02331.x

Iyengar, S. (2011). *Symbolic Systems Biology: Theory and Methods*. Burlington, MA: Jones and Bartlett Learning.

Jessop-Fabre, M. M., and Sonnenschein, N. (2019). Improving reproducibility in synthetic biology. *Front. Bioeng. Biotechnol.* 7:18. doi: 10.3389/fbioe.2018.0018

Karr, J. R., Sanghvi, J. C., Macklin, D. N., Arora, A., and Covert, M. W. (2013). WholeCellKB: model organism databases for comprehensive whole-cell models. *Nucleic Acids Res.* 41, D787–D792.

Karr, J. R., Sanghvi, J. C., Macklin, D. N., Gutschow, M. V., Jacobs, J. M., Bolival, B. Jr., et al. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell* 150, 389–401. doi: 10.1016/j.cell.2012.05.044

Karr, J. R., Takahashi, K., and Funahashi, A. (2015a). The principles of whole-cell modeling. *Curr. Opin. Microbiol.* 27, 18–24. doi: 10.1016/j.mib.2015.06.004

Karr, J. R., Williams, A. H., Zucker, J. D., Raue, A., Steiert, B., Timmer, J., et al. (2015b). Summary of the DREAM8 parameter estimation challenge: toward parameter identification for whole-cell models. *PLoS Comput. Biol.* 11:e1004096. doi: 10.1371/journal.pone.1004096

Karzbrun, E., Shin, J., Bar-Ziv, R. H., and Noireaux, V. (2011). Coarse-grained dynamics of protein synthesis in a cell-free system. *Phys. Rev. Lett.* 106:048104.

King, Z. A., Lu, J., Drager, A., Miller, P., Federowicz, S., Lerman, J. A., et al. (2016). BiGG Models: a platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res.* 44, D515–D522.

Kirk, P., Thorne, T., and Stumpf, M. P. (2013). Model selection in systems and synthetic biology. *Curr. Opin. Biotechnol.* 24, 767–774. doi: 10.1016/j.copbio.2013.03.012

Kitano, H. (2002). Computational systems biology. *Nature* 420, 206–210.

Koch, M., Faulon, J.-L., and Borkowski, O. (2018). Models for cell-free synthetic biology: make prototyping easier, better, and faster. *Front. Bioeng. Biotechnol.* 6:182. doi: 10.3389/fbioe.2018.00182

Kolesnikov, N., Hastings, E., Keays, M., Melnichuk, O., Tang, Y. A., Williams, E., et al. (2015). Array Express update–simplifying data submissions. *Nucleic Acids Res.* 43, D1113–D1116.

Kotula, J. W., Kerns, S. J., Shaket, L. A., Siraj, L., Collins, J. J., Way, J. C., et al. (2014). Programmable bacteria detect and record an environmental signal in the mammalian gut. *Proc. Natl. Acad. Sci. U.S.A.* 111, 4838–4843. doi: 10.1073/pnas.1321321111

Kwiatkowska, M., Norman, G., and Parker, D. (eds) (2011). *PRISM 4.0: Verification of Probabilistic Real-Time Systems 2011*. Berlin: Springer.

Landon, S., Rees-Garbutt, J., Marucci, L., and Grierson, C. (2019). Genome-driven cell engineering review: in vivo and in silico metabolic and genome engineering. *Essays Biochem.* 63, 267–284. doi: 10.1042/ebc20180045

Le Novere, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., et al. (2009). The systems biology graphical notation. *Nat. Biotechnol.* 27, 735–741.

Lee, J. M., Gianchandani, E. P., Eddy, J. A., and Papin, J. A. (2008). Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLoS Comput. Biol.* 4:e1000086. doi: 10.1371/journal.pcbi.1000086.g002

Lin, C., Jain, S., Kim, H., and Bar-Joseph, Z. (2017). Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Res.* 45:e156. doi: 10.1093/nar/gkx681

Linke, C., Chasapi, A., Gonzalez-Novo, A., Al Sawad, I., Tognetti, S., Klipp, E., et al. (2017). A Clb/Cdk1-mediated regulation of Fkh2 synchronizes CLB expression in the budding yeast cell cycle. *NPJ Syst. Biol. Appl.* 3:7.

Lu, T. K., Bowers, J., and Koeris, M. S. (2013). Advancing bacteriophage-based microbial diagnostics with synthetic biology. *Trends Biotechnol.* 31, 325–327. doi: 10.1016/j.tibtech.2013.03.009

Ma, J., Yu, M. K., Fong, S., Ono, K., Sage, E., Demchak, B., et al. (2018). Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* 15, 290–298. doi: 10.1038/nmeth.4627

Macklin, D. N., Ahn-Horst, T. A., Choi, H., Ruggero, N. A., Carrera, J., Mason, J. C., et al. (2020). Simultaneous cross-evaluation of heterogeneous *E. coli* datasets via mechanistic simulation. *Science* 369:eaav3751. doi: 10.1126/science.aav3751

Macklin, D. N., Ruggero, N. A., and Covert, M. W. (2014). The future of whole-cell modeling. *Curr. Opin. Biotechnol.* 28, 111–115. doi: 10.1016/j.copbio.2014.01.012

Marucci, L., Barton, D. A., Cantone, I., Ricci, M. A., Cosma, M. P., Santini, S., et al. (2009). How to turn a genetic circuit into a synthetic tunable oscillator, or a bistable switch. *PLoS One* 4:e8083. doi: 10.1371/journal.pone.0008083

Matsuura, T., Hosoda, K., and Shimizu, Y. (2018). Robustness of a reconstituted *Escherichia coli* protein translation system analyzed by computational modeling. *ACS Synth. Biol.* 7, 1964–1972. doi: 10.1021/acssynbio.8b00228

McAdams, H. H., and Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. U.S.A.* 94, 814–819.

McCloskey, D., Palsson, B. O., and Feist, A. M. (2013). Basic and applied uses of genome-scale metabolic network reconstructions of *Escherichia coli*. *Mol. Syst. Biol.* 9:661. doi: 10.1038/msb.2013.18

McGuffee, S. R., and Elcock, A. H. (2010). Diffusion, crowding & protein stability in a dynamic molecular model of the bacterial cytoplasm. *PLoS Comput. Biol.* 6:e1000694. doi: 10.1371/journal.pcbi.10000694

Medley, J. K., Goldberg, A. P., and Karr, J. R. (2016). Guidelines for reproducibly building and simulating systems biology models. *IEEE Trans. Biomed. Eng.* 63, 2015–2020. doi: 10.1109/tbme.2016.2591960

Mendoza, S. N., Olivier, B. G., Molenaar, D., and Teusink, B. (2019). A systematic assessment of current genome-scale metabolic reconstruction tools. *Genome Biol.* 20:158.

Mol, M., Kabra, R., and Singh, S. (2018). Genome modularity and synthetic biology: engineering systems. *Prog. Biophys. Mol. Biol.* 132, 43–51. doi: 10.1016/j.pbiomolbio.2017.08.002

Mondeel, T., Holland, P., Nielsen, J., and Barberis, M. (2019). ChIP-exo analysis highlights Fkh1 and Fkh2 transcription factors as hubs that integrate multi-scale networks in budding yeast. *Nucleic Acids Res.* 47, 7825–7841. doi: 10.1093/nar/gkz603

Mondeel, T., Ivanov, O., Westerhoff, H. V., Liebermeister, W., and Barberis, M. (2020). Clb3-centered regulations are recurrent across distinct parameter regions in minimal autonomous cell cycle oscillator designs. *NPJ Syst. Biol. Appl.* 6:8.

Moore, S. J., MacDonald, J. T., Wienecke, S., Ishwarbhai, A., Tsipa, A., Aw, R., et al. (2018). Rapid acquisition and model-based analysis of cell-free transcription-translation reactions from nonmodel bacteria. *Proc. Natl. Acad. Sci. U.S.A.* 115, E4340–E4349.

Morton-Firth, C. J., and Bray, D. (1998). Predicting temporal fluctuations in an intracellular signalling pathway. *J. Theor. Biol.* 192, 117–128. doi: 10.1006/jtbi.1997.0651

Moya, A., Gil, R., Latorre, A., Pereto, J., Pilar Garcillan-Barcia, M., and de la Cruz, F. (2009). Toward minimal bacterial cells: evolution vs. design. *FEMS Microbiol. Rev.* 33, 225–235. doi: 10.1111/j.1574-6976.2008.00151.x

Mutturi, S. (2017). FOCuS: a metaheuristic algorithm for computing knockouts from genome-scale models for strain optimization. *Mol. Biosyst.* 13, 1355–1363. doi: 10.1039/c7mb00204a

Noireaux, V., Bar-Ziv, R., and Libchaber, A. (2003). Principles of cell-free genetic circuit assembly. *Proc. Natl. Acad. Sci. U.S.A.* 100, 12672–12677. doi: 10.1073/pnas.2135496100

Noske, A. B., Costin, A. J., Morgan, G. P., and Marsh, B. J. (2008). Expedited approaches to whole cell electron tomography and organelle mark-up in situ in high-pressure frozen pancreatic islets. *J. Struct. Biol.* 161, 298–313. doi: 10.1016/j.jsb.2007.09.015

Novak, B., and Tyson, J. J. (1993). Numerical analysis of a comprehensive model of M-phase control in *Xenopus oocyte* extracts and intact embryos. *J. Cell Sci.* 106(Pt 4), 1153–1168.

Pandit, A. V., Srinivasan, S., and Mahadevan, R. (2017). Redesigning metabolism based on orthogonality principles. *Nat. Commun.* 8:15188.

Pedone, E., Postiglione, L., Aulicino, F., Rocca, D. L., Montes-Olivas, S., Khazim, M., et al. (2019). A tunable dual-input system for on-demand dynamic gene expression regulation. *Nat. Commun.* 10:4481.

Perdikaris, P., and Karniadakis, G. E. (2016). Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. *J. R. Soc. Interf.* 13:20151107. doi: 10.1098/rsif.2015.1107

Posfai, G., Plunkett, G. III, Feher, T., Frisch, D., Keil, G. M., Umenhoffer, K., et al. (2006). Emergent properties of reduced-genome *Escherichia coli*. *Science* 312, 1044–1046. doi: 10.1126/science.1126439

Prescott, A. M., and Abel, S. M. (2017). Combining in silico evolution and nonlinear dimensionality reduction to redesign responses of signaling networks. *Phys. Biol.* 13:066015. doi: 10.1088/1478-3975/13/6/066015

Prescott, T. P., Lang, M., and Papachristodoulou, A. (2015). Quantification of interactions between dynamic cellular network functionalities by cascaded layering. *PLoS Comput. Biol.* 11:e1004235. doi: 10.1371/journal.pone.1004235

Price, M. N., Wetmore, K. M., Waters, R. J., Callaghan, M., Ray, J., Liu, H., et al. (2018). Mutant phenotypes for thousands of bacterial genes of unknown function. *Nature* 557, 503–509. doi: 10.1038/s41586-018-0124-0

Purcell, O., Jain, B., Karr, J. R., Covert, M. W., and Lu, T. K. (2013). Towards a whole-cell modeling approach for synthetic biology. *Chaos* 23:025112. doi: 10.1063/1.4811182

Purcell, O., Savery, N. J., Grierson, C. S., and di Bernardo, M. (2010). A comparative analysis of synthetic genetic oscillators. *J. R. Soc. Interf.* 7, 1503–1524. doi: 10.1098/rsif.2010.0183

Rancati, G., Moffat, J., Typas, A., and Pavelka, N. (2018). Emerging and evolving concepts in gene essentiality. *Nat. Rev. Genet.* 19, 34–49. doi: 10.1038/nrg.2017.74

Rand, D. A. (2008). Mapping global sensitivity of cellular network dynamics: sensitivity heat maps and a global summation law. *J. R. Soc. Interf.* 5(Suppl. 1), S59–S69.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabasi, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science* 297, 1551–1555. doi: 10.1126/science.1073374

Ray, O., Soh, T., and Inoue, K. (2011). "Analysing pathways using ASP-based approaches," in *Proceedings of the 2010 Conference on Algebraic and Numeric Biology*, Berlin.

Rees-Garbutt, J., Chalkley, O., Landon, S., Purcell, O., Marucci, L., and Grierson, C. (2020). Designing minimal genomes using whole-cell models. *Nat. Commun.* 11:836.

Reuss, D. R., Altenbuchner, J., Mader, U., Rath, H., Ischebeck, T., Sappa, P. K., et al. (2017). Large-scale reduction of the *Bacillus subtilis* genome: consequences for the transcriptional network, resource allocation, and metabolism. *Genome Res.* 27, 289–299. doi: 10.1101/gr.215293.116

Rozanski, R., Ray, O., King, R., and Bragaglia, S. (2015). "Automating development of metabolic network models," in *Computational Methods in Systems Biology. CMSB 2015. Lecture Notes in Computer Science*, Vol. 9308, eds O. Roux and J. Bourdon (Cham: Springer).

Sajed, T., Marcu, A., Ramirez, M., Pon, A., Guo, A. C., Knox, C., et al. (2016). ECMDB 2.0: A richer resource for understanding the biochemistry of *E. coli*. *Nucleic Acids Res.* 44, D495–D501.

Sali, A., Berman, H. M., Schwede, T., Trewhella, J., Kleywegt, G., Burley, S. K., et al. (2015). Outcome of the first wwPDB hybrid/integrative methods task force workshop. *Structure* 23, 1156–1167. doi: 10.1016/j.str.2015.05.013

Shu, J., and Shuler, M. L. (1989). A mathematical model for the growth of a single cell of *E. coli* on a glucose/glutamine/ammonium medium. *Biotechnol. Bioeng.* 33, 1117–1126. doi: 10.1002/bit.260330907

Siegal-Gaskins, D., Tuza, Z. A., Kim, J., Noireaux, V., and Murray, R. M. (2014). Gene circuit performance characterization and resource usage in a cell-free "breadboard". *ACS Synth. Biol.* 3, 416–425. doi: 10.1021/sb400203p

Silverman, A. D., Karim, A. S., and Jewett, M. C. (2020). Cell-free gene expression: an expanded repertoire of applications. *Nat. Rev. Genet.* 21, 151–170. doi: 10.1038/s41576-019-0186-3

Singla, J., McClary, K. M., White, K. L., Alber, F., Sali, A., and Stevens, R. C. (2018). Opportunities and challenges in building a spatiotemporal multi-scale model of the human pancreatic beta cell. *Cell* 173, 11–19. doi: 10.1016/j.cell.2018.03.014

Slomovic, S., Pardee, K., and Collins, J. J. (2015). Synthetic biology devices for in vitro and in vivo diagnostics. *Proc. Natl. Acad. Sci. U.S.A.* 112, 14429–14435. doi: 10.1073/pnas.1508521112

Smucker, B., Krzywinski, M., and Altman, N. (2018). Optimal experimental design. *Nat. Methods* 15, 559–560.

Solernou, A., Hanson, B. S., Richardson, R. A., Welch, R., Read, D. J., Harlen, O. G., et al. (2018). Fluctuating finite element analysis (FFEA): a continuum mechanics software tool for mesoscale simulation of biomolecules. *PLoS Comput. Biol.* 14:e1005897. doi: 10.1371/journal.pone.1005897

Stogbauer, T., Windhager, L., Zimmer, R., and Radler, J. O. (2012). Experiment and mathematical modeling of gene expression dynamics in a cell-free system. *Integr. Biol.* 4, 494–501.

Szigeti, B., Roth, Y. D., Sekar, J. A. P., Goldberg, A. P., Pochiraju, S. C., and Karr, J. R. (2018). A blueprint for human whole-cell modeling. *Curr. Opin. Syst. Biol.* 7, 8–15. doi: 10.1016/j.coisb.2017.10.005

Takahashi, K., Arjunan, S. N., and Tomita, M. (2005). Space in systems biology of signaling pathways–towards intracellular molecular crowding in silico. *FEBS Lett.* 579, 1783–1788. doi: 10.1016/j.febslet.2005.01.072

Takahashi, M. K., Chappell, J., Hayes, C. A., Sun, Z. Z., Kim, J., Singhal, V., et al. (2015). Rapidly characterizing the fast dynamics of RNA genetic circuitry with cell-free transcription-translation (TX-TL) systems. *ACS Synth. Biol.* 4, 503–515. doi: 10.1021/sb400206c

Tang, P. W., Chua, P. S., Chong, S. K., Mohamad, M. S., Choon, Y. W., Deris, S., et al. (2015). A review of gene knockout strategies for microbial cells. *Recent. Pat. Biotechnol.* 9, 176–197. doi: 10.2174/1872208310666160517115047

Thiele, I., Jamshidi, N., Fleming, R. M., and Palsson, B. O. (2009). Genome-scale reconstruction of *Escherichia coli*'s transcriptional and translational machinery: a knowledge base, its mathematical formulation, and its functional characterization. *PLoS Comput. Biol.* 5:e1000312. doi: 10.1371/journal.pcbi.10000312

Thul, P. J., Akesson, L., Wiking, M., Mahdessian, D., Geladaki, A., Ait Blal, H., et al. (2017). A subcellular map of the human proteome. *Science* 356:6340.

Tomazou, M., Barahona, M., Polizzi, K. M., and Stan, G. B. (2018). Computational Re-design of synthetic genetic oscillators for independent amplitude and frequency modulation. *Cell Syst.* 6:50.

Tomita, M. (2001). Whole-cell simulation: a grand challenge of the 21st century. *Trends Biotechnol.* 19, 205–210. doi: 10.1016/s0167-7799(01)01636-5

Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T. S., Matsuzaki, Y., Miyoshi, F., et al. (1999). E-CELL: software environment for whole-cell simulation. *Bioinformatics* 15, 72–84. doi: 10.1093/bioinformatics/15.1.72

Tyson, J. J. (1991). Modeling the cell division cycle: cdc2 and cyclin interactions. *Proc. Natl. Acad. Sci. U.S.A.* 88, 7328–7332. doi: 10.1073/pnas.88.16.7328

Underwood, K. A., Swartz, J. R., and Puglisi, J. D. (2005). Quantitative polysome analysis identifies limitations in bacterial cell-free protein synthesis. *Biotechnol. Bioeng.* 91, 425–435. doi: 10.1002/bit.20529

UniProt Consortium, T. (2018). UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* 46, 2699. doi: 10.1093/nar/gky092

van der Meer, J. R., and Belkin, S. (2010). Where microbiology meets microengineering: design and applications of reporter bacteria. *Nat. Rev. Microbiol.* 8, 511–522. doi: 10.1038/nrmicro2392

van der Zee, L., and Barberis, M. (2019). Advanced modeling of cellular proliferation: toward a multi-scale framework coupling cell cycle to metabolism

by integrating logical and constraint-based models. *Methods Mol. Biol.* 2049, 365–385. doi: 10.1007/978-1-4939-9736-7_21

Varma, A., and Palsson, B. O. (1994). Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110. *Appl. Environ. Microbiol.* 60, 3724–3731. doi: 10.1128/aem.60.10.3724-3731.1994

Vernon, I., Liu, J., Goldstein, M., Rowe, J., Topping, J., and Lindsey, K. (2018). Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions. *BMC Syst. Biol.* 12:1. doi: 10.1186/s12918-017-0484-3

Vilkhovoy, M., Horvath, N., Shih, C. H., Wayman, J. A., Calhoun, K., Swartz, J., et al. (2018). Sequence specific modeling of *E. coli* cell-free protein synthesis. *ACS Synth. Biol.* 7, 1844–1857. doi: 10.1021/acssynbio.7b00465

Vyshemirsky, V., and Girolami, M. (2008). BioBayes: a software package for bayesian inference in systems biology. *Bioinformatics* 24, 1933–1934. doi: 10.1093/bioinformatics/btn338

Waltemath, D., Karr, J. R., Bergmann, F. T., Chelliah, V., Hucka, M., Krantz, M., et al. (2016). Toward community standards and software for whole-cell modeling. *IEEE Trans. Biomed. Eng.* 63:14.

Wang, L., and Maranas, C. D. (2018). MinGenome: an in silico top-down approach for the synthesis of minimized genomes. *ACS Synth. Biol.* 7, 462–473. doi: 10.1021/acssynbio.7b00296

Way, J. C., Collins, J. J., Keasling, J. D., and Silver, P. A. (2014). Integrating biological redesign: where synthetic biology came from and where it needs to go. *Cell* 157, 151–161. doi: 10.1016/j.cell.2014.02.039

Wilkinson, D. J. (2007). Bayesian methods in bioinformatics and computational systems biology. *Brief Bioinform.* 8, 109–116. doi: 10.1093/bib/bbm007

Wittig, U., Kania, R., Golebiewski, M., Rey, M., Shi, L., Jong, L., et al. (2012). SABIO-RK–database for biochemical reaction kinetics. *Nucleic Acids Res.* 40, D790–D796.

Woolston, B. M., Edgar, S., and Stephanopoulos, G. (2013). Metabolic engineering: past and future. *Annu. Rev. Chem. Biomol. Eng.* 4, 259–288. doi: 10.1146/annurev-chembioeng-061312-103312

Xu, X., Liu, Y., Du, G., Ledesma-Amaro, R., and Liu, L. (2020). Microbial chassis development for natural product biosynthesis. *Trends Biotechnol.* 38, 779–796. doi: 10.1016/j.tibtech.2020.01.002

Yilmaz, L. S., and Walhout, A. J. (2017). Metabolic network modeling with model organisms. *Curr. Opin. Chem. Biol.* 36, 32–39. doi: 10.1016/j.cbpa.2016.12.025

Yu, I., Mori, T., Ando, T., Harada, R., Jung, J., Sugita, Y., et al. (2016). Biomolecular interactions modulate macromolecular structure and dynamics in atomistic model of a bacterial cytoplasm. *eLife* 5:e19274.

Yu, M. K., Ma, J., Fisher, J., Kreisberg, J. F., Raphael, B. J., and Ideker, T. (2018). Visible machine learning for biomedicine. *Cell* 173, 1562–1565. doi: 10.1016/j.cell.2018.05.056

Zhou, J., Wu, R., Xue, X., and Qin, Z. (2016). CasHRA (Cas9-facilitated homologous recombination assembly) method of constructing megabase-sized DNA. *Nucleic Acids Res.* 44, e124. doi: 10.1093/nar/gkw475

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Citation: Marucci L, Barberis M, Karr J, Ray O, Race PR, de Souza Andrade M, Grierson C, Hoffmann SA, Landon S, Rech E, Rees-Garbutt J, Seabrook R, Shaw W and Woods C (2020) Computer-Aided Whole-Cell Design: Taking a Holistic Approach by Integrating Synthetic With Systems Biology. Front. Bioeng. Biotechnol. 8:942. doi: 10.3389/fbioe.2020.00942

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 28 August 2020
doi: 10.3389/fbioe.2020.00893

Novel Tunable Spatio-Temporal Patterns From a Simple Genetic Oscillator Circuit

Guillermo Yáñez Feliú[1†], *Gonzalo Vidal*[2†], *Macarena Muñoz Silva*[2] *and Timothy J. Rudge*[1,2*]

[1] *Department of Chemical and Bioprocess Engineering, School of Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile,* [2] *Institute for Biological and Medical Engineering, Schools of Engineering, Biology and Medicine, Pontificia Universidad Católica de Chile, Santiago, Chile*

Multicellularity, the coordinated collective behavior of cell populations, gives rise to the emergence of self-organized phenomena at many different spatio-temporal scales. At the genetic scale, oscillators are ubiquitous in regulation of multicellular systems, including during their development and regeneration. Synthetic biologists have successfully created simple synthetic genetic circuits that produce oscillations in single cells. Studying and engineering synthetic oscillators in a multicellular chassis can therefore give us valuable insights into how simple genetic circuits can encode complex multicellular behaviors at different scales. Here we develop a study of the coupling between the repressilator synthetic genetic ring oscillator and constraints on cell growth in colonies. We show *in silico* how mechanical constraints generate characteristic patterns of growth rate inhomogeneity in growing cell colonies. Next, we develop a simple one-dimensional model which predicts that coupling the repressilator to this pattern of growth rate via protein dilution generates traveling waves of gene expression. We show that the dynamics of these spatio-temporal patterns are determined by two parameters; the protein degradation and maximum expression rates of the repressors. We derive simple relations between these parameters and the key characteristics of the traveling wave patterns: firstly, wave speed is determined by protein degradation and secondly, wavelength is determined by maximum gene expression rate. Our analytical predictions and numerical results were in close quantitative agreement with detailed individual based simulations of growing cell colonies. Confirming published experimental results we also found that static ring patterns occur when protein stability is high. Our results show that this pattern can be induced simply by growth rate dilution and does not require transition to stationary phase as previously suggested. Our method generalizes easily to other genetic circuit architectures thus providing a framework for multi-scale rational design of spatio-temporal patterns from genetic circuits. We use this method to generate testable predictions for the synthetic biology design-build-test-learn cycle.

Keywords: genetic circuits, repressilator, biodesign, spatio-temporal patterns, traveling waves, cellModeller, synthetic biology

OPEN ACCESS

Edited by:
Thomas E. Gorochowski, University of Bristol, United Kingdom

Reviewed by:
Andras Gyorgy, New York University Abu Dhabi, United Arab Emirates
Angel Goñi-Moreno, Newcastle University, United Kingdom

***Correspondence:**
Timothy J. Rudge
trudge@uc.cl

[†] *These authors have contributed equally to this work*

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *08 June 2020*
Accepted: *13 July 2020*
Published: *28 August 2020*

Citation:
Yáñez Feliú G, Vidal G, Muñoz Silva M and Rudge TJ (2020) Novel Tunable Spatio-Temporal Patterns From a Simple Genetic Oscillator Circuit. Front. Bioeng. Biotechnol. 8:893. doi: 10.3389/fbioe.2020.00893

1. INTRODUCTION

Multicellularity and collective cell behavior exemplify the emergence of complex patterns and structures across scales in living systems. When cells interact they can generate higher order patterns of gene expression (differentiation) as well as patterns of mechanical stresses and strains (Chan et al., 2017; Vining and Mooney, 2017). This process takes place in natural phenomena such as embryonic development, tumor formation, wound healing, among others (Velardo et al., 2004; Aboobaker et al., 2005; Khain and Sander, 2006; Gjorevski and Nelson, 2010; Santos-Moreno and Schaerli, 2019). Understanding how these patterns are generated and maintained will enable applications in tissue engineering and regenerative medicine. However, natural emergent multicellular phenomena present numerous unknown processes that pose difficulties for understanding the fundamental mechanisms underlying pattern formation.

Synthetic biology applies design principles to generate combinations of genetic parts that perform a given function, for example oscillation, which at the same time helps us to understand the complexity inherent to natural systems. The prototypical engineering process is the design-build-test-learn cycle, which is an iterative process relying heavily on models of genetic circuit function. A variety of genetic circuits have been designed, analyzed, simulated, and then implemented in this way. These synthetic circuits simplify biological systems reproducing a specific function (Xie and Fussenegger, 2018) such as toggle switches (Gardner et al., 2000; Yeung et al., 2017), oscillators (Elowitz and Leibler, 2000; Stricker et al., 2008; Danino et al., 2010; Potvin-Trottier et al., 2016), logic gates (Tamsir et al., 2011; Nielsen et al., 2016; Green et al., 2017; Kim et al., 2018), and arithmetic operators (Wong et al., 2015; Ausländer et al., 2018).

While these circuits are often studied as dynamical systems in single cells or well mixed populations, the function of genetic circuits has also been studied in cell colonies (Luo et al., 2019; Santos-Moreno and Schaerli, 2019) through the engineering of patterns of gene expression such as symmetry breaking (Nuñez et al., 2017), Turing patterns (Karig et al., 2018), fractal patterns (Rudge et al., 2013), tissue like structures (Toda et al., 2018; Healy and Deans, 2019), among others. These emergent spatio-temporal patterns depend on mechanical constraints on cells, which are the result of cell-cell and cell-substrate interactions. Thus, synthetic gene circuits can be engineered to generate higher order spatio-temporal patterns when coupled to mechanical constraints.

We focus here on the repressilator (Elowitz and Leibler, 2000), a gene network that encodes a ring oscillator topology consisting of three repressors, where repressor 1 inhibits repressor 2, repressor 2 inhibits repressor 3, and repressor 3 inhibits repressor 1 (**Figure 1**). In the original realization of this circuit topology (Elowitz and Leibler, 2000) the circuit was subject to significant effects of noise and oscillations quickly became desynchronized. Recently, the circuit was revisited by Potvin-Trottier et al. (2016) with microfluidics systems that allowed them to observe single cells oscillating synchronously in chemostatic conditions for long periods of time. In this work sources of noise were reduced in several ways. Firstly, the fluorescent reporters were integrated into the same low-copy plasmid as the repressilator reducing the standard deviation in amplitude greatly. They also removed the degradation tags and used a protease deletion strain (ΔclpXP) as the chassis to remove noise from enzymatic queuing (Cookson et al., 2011; Steiner et al., 2016). They also increased the effective repression threshold with a high-copy titration "sponge" plasmid that sequesters a proportion of the TetR repressor, since low repression thresholds imply sensitivity to noisy repressor expression levels. These modifications allowed regular and sustained synchronous oscillations that peaked around every 14 generations. The circuit oscillated for approximately 18 periods before accumulating half a period of drift, demonstrating that cells remained synchronized for more than 250 generations. Strikingly, Potvin-Trottier et al. (2016) were able to observe whole flasks of liquid bacterial culture oscillate synchronously, and bacterial colonies form coherent ring patterns at macroscopic scale. These findings show that the repressilator can be effectively isolated from noise, function in a robust and synchronous fashion, and is capable of forming spatial patterns.

Models are essential in the design process, they allow engineers to screen the parameter space looking for possible functional constructions (Endy and Brent, 2001; De Jong, 2002). Synthetic biology has gone from intracellular dynamic models using ODEs (Elowitz and Leibler, 2000; Gardner et al., 2000), and SSA (Potvin-Trottier et al., 2016; Karig et al., 2018) to sophisticated collective behavior models based on individual agents (Rudge et al., 2012; Gorochowski, 2016) and integrated circuit-host models (Sickle et al., 2020). Using cellular scale individual-based models (IBMs) gives rich information about the emergent collective properties of cell populations due to the interactions between themselves and their environment. These models track cell growth and gene expression in ways analogous to experiments performed in controlled environmental conditions with specified properties such as viscosity, chemical concentrations, etc. This makes them an essential tool in the analysis and design of emergent properties of genetic circuits operating in multicellular chassis. However these models are complex and require significant computation time, highlighting the need for simple tractable mathematical and computational methods.

In this study we uncover novel spatio-temporal patterns of gene expression generated by the repressilator in growing cell colonies, and establish a simple method for their design. Since it is generalizable, this work provides a quantitative framework for multi-scale rational design of spatio-temporal patterns from genetic circuits. We provide testable predictions for the synthetic biology design-build-test-learn cycle for engineering repressilator spatio-temporal pattern.

2. RESULTS

2.1. Growth Rate Variation in Growing Microcolonies

We consider the case of *Escherichia coli*, the cellular chassis for which the repressilator (**Figure 1**) was designed,

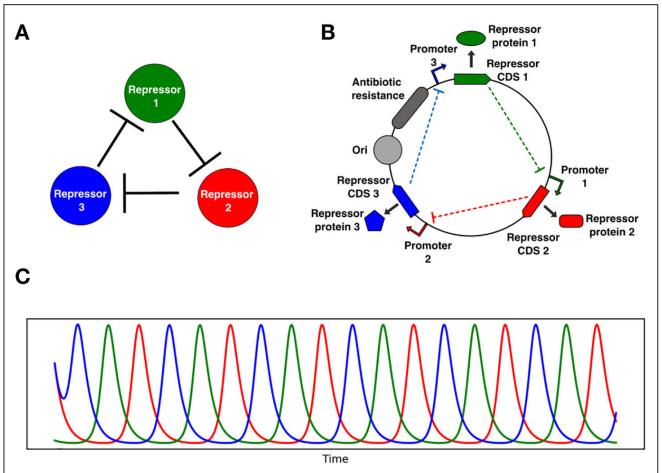


FIGURE 1 | Repressilator genetic oscillator circuit. **(A)** Network representation of the repressilator, in the schematic 1 represses 2, 2 represses 3, and 3 represses 1. **(B)** Genetic circuit diagram of a plasmid encoding the repressilator. **(C)** Oscillating repressor concentrations over time computed by solving Equation (11) with fixed growth rate. CDS 1, 2, and 3 are the complementary DNA sequences coding for each repressor, and Ori is the origin of replication.

growing on a viscous substrate such as a hydrogel or PDMS (polydimethylsiloxane) and supplied with fresh nutrients via microfluidic channels. Growth is constrained by forces between cells and between cells and the substrate due to viscous drag (Rudge et al., 2012). The cells in such a system are constrained to a monolayer and form a quasi-two-dimensional array of extending rod shapes (Farrell et al., 2013; Grant et al., 2014). We used an individual-based model (Rudge et al., 2012) to characterize the distribution of cell growth rates across a two dimensional cell monolayer growing in such conditions over time. We simulated the growth of microcolonies from single cells to populations of approximately 60,000 cells with a radius of approximately 200 cell diameters. **Figures 2A–C** show the development of a colony from approximately 5,000 to 50,000 cells. The distribution of growth rates across the colony has a clear radially symmetric pattern with a maximum at the edge of the colony (**Figure S1**). This is as expected (Vicsek et al., 1990; Smith et al., 2017) since the cells at the edge of the colony are relatively unconstrained. Thus individual bacteria inside microcolonies perceive a different biophysical environment depending on their spatial position. Taking radial averages of growth rate on growing colonies over a range of time points we see the same exponential decay relative to the colony edge (**Figure 2E**), leading to a simple model for the cell growth rate as a function of the radial position of the cell with respect to the colony edge $r(t)$,

$$\bar{\mu}(t) = e^{-r(t)/r_0}, \tag{1}$$

where r_0 (8.23 ± 1.69 cell diameters) is the characteristic length scale of the radial variation in growth rate, and we have normalized by μ_0—the maximal unconstrained growth rate of the cells. These results suggest that growth rate time dynamics are determined by radial distance from the colony edge. After a short transient, the colony edge moves with constant velocity $v_{front} = 5.00$ so that R_{max} increases linearly (**Figure 2F**).

Assuming a two-dimensional densely packed monolayer with random cell orientations, growth is isotropic and expansion is equal in all directions. The area expansion rate approximates the growth rate and is given by the divergence of the velocity field,

$$\nabla \cdot \mathbf{v} = \frac{1}{A}\frac{dA}{dt}, \tag{2}$$

where A is the cell area and $\mathbf{v}$ is the velocity. Since growth is isotropic we may decompose the expansion rate equally into its radial and perpendicular components. Considering the velocity v in the radial direction r, and velocity w in the perpendicular direction s, and expanding the divergence term, Equation (2) gives,

$$\frac{dv}{dr} + \frac{dw}{ds} = 2\frac{dv}{dr} = \mu(r). \tag{3}$$

Hence, with $v = dr/dt$, and considering only the radial direction, we can rescale time as $t \to t\mu_0$ and radial distance as $r \to r/r_0$ to obtain,

$$\frac{d}{dr}\left(\frac{dr}{dt}\right) = \frac{1}{2}e^{-r}. \tag{4}$$

Integrating by r and t results in,

$$v(t) = \frac{1}{2}\left(1 + \exp\left(\frac{\tau - t}{2}\right)\right)^{-1}, \tag{5}$$

$$r(t) = \log\left(1 + \exp\left(\frac{t - \tau}{2}\right)\right), \tag{6}$$

$$\bar{\mu}(t) = \left(1 + \exp\left(\frac{t - \tau}{2}\right)\right)^{-1}, \tag{7}$$

where $\tau = -2\log(\exp(r(0)) - 1)$, and $r(0)$ is the initial radial position of the cell.

Equations (5)–(7) give us valuable insights into the system behavior (**Figure 3**). The velocity in the radial direction (away from the colony edge) is a sigmoidal logistic function and saturates to a velocity of $v = 1/2$ as r increases (**Figure 3A**). This gives the front velocity as $v_{front} = 1/2$ in rescaled units. Correspondingly the radial position relative to the colony edge r tends toward linear increase at velocity $v = 1/2$ as the cell becomes effectively stationary relative to the colony center (**Figure 3B**). In real units this means that the front velocity is $v_{front} = r_0/2$, where r_0 is the length scale of growth rate variation (Equation 1). This is consistent with our individual based simulations (**Figure 2F**), in which $v_{front} = 5.00$ and $r_0 = 8.23 \pm 1.69$. The growth rate is also sigmoidal and tends from maximum at the colony edge to zero as the cell moves away from the growing front of the colony (**Figure 3C**).

The critical time τ, depending on the initial cell position, is the time at which the growth rate and velocity are at their half maximum values, and the cell is at radial position $r = \log(2)$ (**Figure 3**, dashed lines). At this time the cell switches from a high

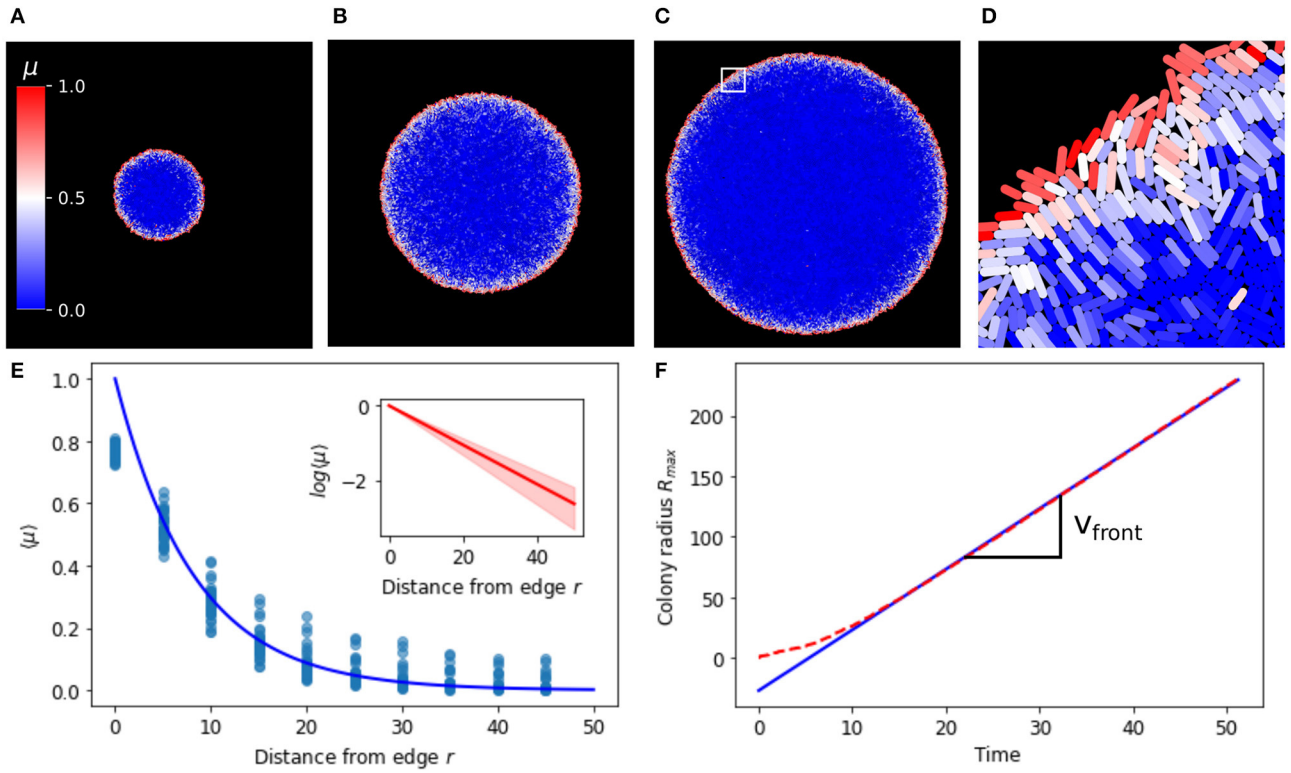


FIGURE 2 | Average growth rate decays exponentially with distance relative to the edge. **(A–C)** Are growing colonies at 5,000, 25,000 and 50,000 cells, respectively. **(D)** Is a zoom from the white square region in **(C)**. Single cells are colored according to their growth rate, which ranges between 0 and 1. **(E)** Average growth rate $\langle\mu\rangle$ at different distances from the colony edge, blue line is an exponential function ($e^{-r/r0}$) with $r0 = 8.23$ fitted to the binned data (blue dots). Inset shows the log scale of the average growth rate $\langle\mu\rangle$ at different distances from the edge. **(F)** Colony radius (R_{max}) over time (dashed red line), linear fit to data at t>10 giving $v_{front} = 5.00$ the velocity of the colony edge (blue line).

growth, low velocity regime (remaining close to the colony edge), to a high velocity, low growth regime (remaining stationary while the colony edge propagates). The time τ for this switch to occur is greater for cells closer to the colony edge, that is they remain in the fast growing regime for longer. Thus, cells effectively experience a switch in growth rate and velocity at their critical time τ, which depends on their initial radial position in the colony.

2.2. Dynamic Growth Rate Dependent Mathematical Model of the Repressilator

Here we develop a simple mathematical model coupling the repressilator genetic circuit to growth rate variation via simple dilution of proteins. The repressilator can be considered as an abstract genetic circuit topology. We consider an implementation of this topology following the design modifications made by Potvin-Trottier et al. (2016), which essentially isolated the circuit from noise and allowed sustained and synchronous oscillations over time scales up to 250 generations. Stochastic simulations performed with relevant parameters reproduced this behavior, showing essentially deterministic behavior (**Figure S2**), therefore we may use a simpler differential equation model to track the repressor concentration of each cell over time. A simple two-step model of the balanced repressilator genetic circuit (**Figure 1**), a type of genetic ring oscillator, can be formulated as follows,

$$\frac{dm_i}{dt} = \frac{a + b(p_j/K_j)^n}{1 + (p_j/K_j)^n} - \delta m_i, \tag{8}$$

$$\frac{dp_i}{dt} = cm_i - \gamma p_i - \mu(t)p_i, \tag{9}$$

where i is one of the three genes, j is its corresponding repressor gene, m_i, p_i are the mRNA and protein concentrations respectively, a is the constitutive transcription rate, b is the leaky or repressed transcription rate, μ is the instantaneous growth rate of the cell or population of cells, γ is the protein degradation rate, and δ is the mRNA degradation rate. Order of magnitude estimates of these parameters are given in **Supplementary Material**.

Since mRNAs are typically short lived (see **Supplementary Material**), we may assume quasi-steady state concentrations and the system is,

$$\frac{dp_i}{dt} = \frac{c}{\delta}\frac{a + b(p_j/K_j)^n}{1 + (p_j/K_j)^n} - \gamma p_i - \mu(t)p_i. \tag{10}$$

Rescaling protein concentration as $p_j \rightarrow p_j/K_j$ and time by $t \rightarrow t\mu_0$ with μ_0 the maximal growth rate, assuming that basal

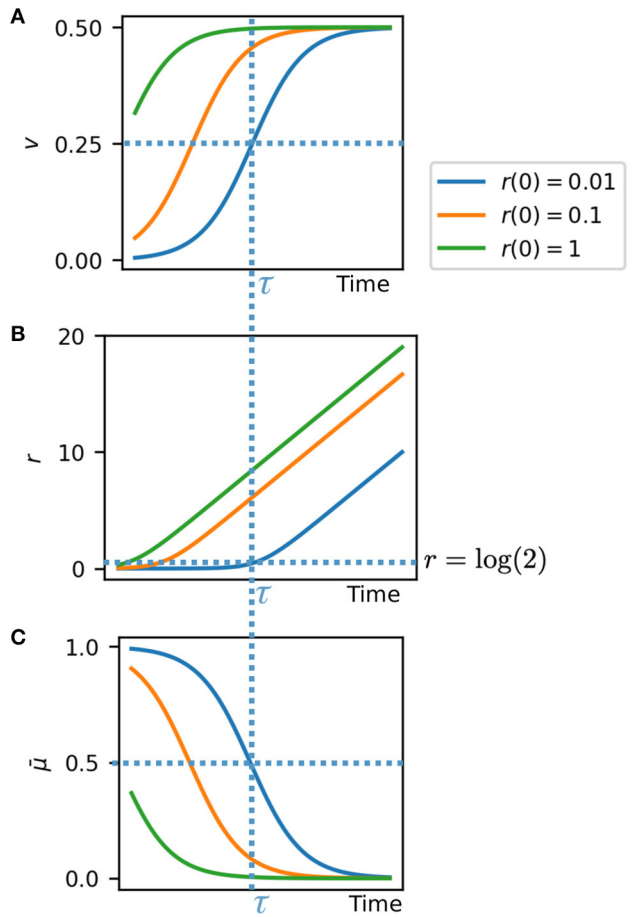


FIGURE 3 | Analytical solutions for cell velocity **(A)** and position **(B)** relative to the colony edge, and growth rate **(C)**, based on exponential decay of growth rate with distance from colony edge. At critical time τ the system switches between two regimes: high growth/low velocity, and low growth/high velocity.

expression is negligible, and combining with (Equation 7),

$$\frac{dp_i}{dt} = \frac{\alpha}{1+p_j^n} - \bar{\gamma}p_i - p_i\left(1+\exp\left(\frac{t-\tau}{2}\right)\right)^{-1}, \qquad (11)$$

where $\alpha = ca/\delta\mu_0 K$ (order of magnitude 10^4, see **Supplementary Material**) is the steady state maximal gene expression rate and the constant $\bar{\gamma} = \gamma/\mu_0$ is the protein degradation rate as a fraction of the maximal growth rate (order of magnitude 1). This model depends on three dimensionless parameters α, $\bar{\gamma}$, and n.

In this model we assume that the dominant effect of growth rate variation is by dilution of proteins. While there is evidence for growth rate dependencies of transcription and translation rates and plasmid copy number (Neubauer et al., 2003; Klumpp et al., 2009; Klumpp, 2011), all of which affect the parameters of the model, these effects have only been observed due to different biochemical environments. In spatially constrained cell populations the shape of the growth profile $\bar{\mu}(t)$ depends both upon the biochemical environment and mechanical constraints (Andersen and von Meyenburg, 1980; Matsushita and Fujikawa, 1990; Tuson et al., 2012; Farrell et al., 2013; Rudge et al., 2013; Smith et al., 2017; Winkle et al., 2018). At the typical bacterial microcolony scale the biochemical environment is essentially uniform in space due to the fast diffusion of small molecules like sugars and aminoacids (Matson and Characklis, 1976; Fraleigh and Bungay, 1986; Guélon et al., 2012). Using microfluidic devices cells can be maintained in constant fresh media allowing observation of the long term dynamics of genetic circuits (Danino et al., 2010; Long et al., 2013; Potvin-Trottier et al., 2016). Under these conditions then the predominant factors determining growth rate are physical forces and constraints.

2.3. Protein Dilution Enables the Repressilator to Produce Traveling Waves in Growing Microcolonies

The model presented above (Equations 5–7 and 11) describes the trajectories of cells as they move in the radial direction and change their protein concentrations over time. Assuming that the motion and growth of cells is not affected by the expression of repressor genes, this model describes the mean behavior of cells starting from some initial radial position. It is obvious from these equations that in the absence of growth the system can only produce plane wave, homogeneously synchronized oscillations. However, in the presence of growth we have an explicit relation between cell position and protein expression rate, enabling spatio-temporal pattern formation.

Since growth dilutes proteins the effective degradation rate of the repressors is $\bar{\gamma} + \bar{\mu}(t)$. The effect of the sigmoidal growth rate switch on the repressilator is therefore to decrease the effective degradation rate of the repressors from $\bar{\gamma} + 1$ to $\bar{\gamma}$ as cells move out of the growing regime (Equation 7). Potvin-Trottier et al. (2016) showed that increasing the degradation rate of repressors by appending a degradation tag reduced the period of oscillations T. This decrease was by approximately a factor of two at 37°C, with less effect at lower temperatures likely due to decrease in protease activity (Purcell et al., 2012). We confirmed this result using our model by numerically integrating Equation (11) at fixed effective degradation rates $\bar{\gamma} + \bar{\mu}$ (**Figure 4**). The frequency of oscillations $f = 1/T$ was proportional to the degradation rate, with a slope depending on α, the maximum gene expression rate. Hence in colonies, as cells move away from the edge due to mechanical constraints the effective repressor degradation rate decreases and the period of their oscillations increases.

After the critical time τ the period of oscillations increases as the cell switches from high growth rate and low velocity to low growth rate and high velocity (Equations 5–7). This means that there is effectively an interior region oscillating with long period T_{int} and an exterior region oscillating with short period T_{ext}. The phase offset between peaks of the two signals after some time Δt is,

$$\Delta T = \left(\frac{T_{int} - T_{ext}}{T_{int}}\right)\Delta t. \qquad (12)$$

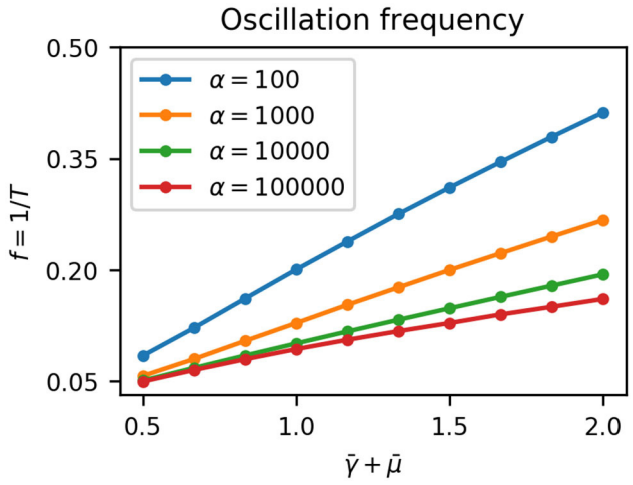


FIGURE 4 | Oscillation frequency f (or period T) depends on the effective degradation rate of repressors $\bar{\gamma} + \bar{\mu}$. For a given maximum gene expression rate α the frequency is proportional to the effective degradation rate, and increasing α decreases the frequency.

When the phase offset $\Delta T = T_{ext}$ we have in phase oscillations. The time required for a cell to achieve this phase offset is the time spent in the high growth regime τ, plus the time spent in the low growth regime, hence,

$$t^* = \tau + \Delta t = \tau + \frac{T_{int}T_{ext}}{T_{int} - T_{ext}}, \tag{13}$$

is the time at which the cell is in phase with the edge of the colony, the wave source. At time t^* the distance from the edge $r(t^*)$ of this cell can be obtained from Equation (6), and since this is the peak-to-peak distance it gives the wavelength λ. Assuming that $\exp(t^*) \gg 1$,

$$\lambda = \frac{T_{int}T_{ext}}{2(T_{int} - T_{ext})}. \tag{14}$$

The wave propagation velocity is $v_p = \lambda/T_{ext} - \bar{v}_{front} = \lambda/T_{ext} - \frac{1}{2}$, where $\bar{v}_{front}$ is the velocity of the colony edge in normalized distance units, hence,

$$v_p = \frac{T_{ext}}{2(T_{int} - T_{ext})}. \tag{15}$$

Equations (14) and (15) show that when $T_{ext} < T_{int}$ the system generates traveling waves with finite wavelength and wave speed. When $T_{int} = T_{ext}$, there is no effect of mechanical constraint on oscillation period and we find that $v_p = \infty$ and $\lambda = \infty$, meaning that the system forms homogeneous plane waves with the whole colony oscillating synchronously. As $T_{int} \to \infty$ the interior does not oscillate and we find that $v_p \to 0$ and $\lambda \to T_{ext}/2$ and we thus have static rings of gene expression with spatial wavelength $T_{ext}/2 = \bar{v}_{front}T_{ext}$. This is the case when protein degradation is negligible ($\bar{\gamma} = 0$), a condition under which the repressilator has been shown to form static rings in growing colonies (Potvin-Trottier et al., 2016). Thus we have shown analytically that growth rate heterogeneity induces the repressilator to form either static rings or traveling waves in growing cell colonies, depending on the degradation rate of the repressors.

2.4. Novel Spatio-Temporal Patterns Emerging From Repressilator Dynamics

To test the predicted spatio-temporal patterns we integrated Equation (11) from a range of initial cell radial positions to construct the kymograph $p_i(R, t)$, where $R = t/2 - r$ is the rescaled distance from the center of the colony. A kymograph (**Figures 5A–C**) represents the spatial dynamics of a one-dimensional system, such as ours, evolving over time. Each point in the kymograph represents the state of the system at a given time (x-axis) and position (y-axis) with a color. By taking radial averages the kymograph fully characterizes radially symmetric spatio-temporal patterns with the vertical axis representing distance from the center of the colony. Here we reflect the kymograph to represent the symmetric structure of the pattern (**Figures 5A–C**), whereby the growth of the colony can be seen as two linearly expanding borders forming a triangular shape. The slope of this border is the front speed $\bar{v}_{front} = 1/2$. The color represents the radially averaged repressor protein concentrations (red, green, and blue) at each position at each time point, normalized to their maximum values. The corresponding predicted colony pattern is shown inset in **Figure 5C**. Stripes in the kymograph represent rings of gene expression. Horizontal stripes show static rings since their radial position does not change (**Figure 5A**). Vertical stripes would represent in-phase homogeneous oscillations since they do not vary in space (**Figure 5B**). Diagonal stripes represent traveling waves, moving rings of repressor expression, since they vary in both space and time (**Figure 5C**). Hence we confirm our theoretical prediction of traveling wave patterns.

The spatio-temporal dynamics of the system are described by two parameters that can be extracted from the kymographs. The slope of each stripe gives the wave speed v_p, and the vertical peak-to-peak distance gives the spatial wavelength $\lambda = (v_p + \bar{v}_{front})T = (v_p + \frac{1}{2})T$, where T is the period of oscillations at the colony edge (the wave front) and $\bar{v}_{front}$ is the front velocity in normalized distance units (**Figure 5C**).

2.5. Tuning the Repressilator to Control Spatio-Temporal Pattern Formation

In order to quantitatively test our theoretical predictions and to characterize the design space of these traveling wave patterns we scanned the parameter space within physiologically relevant ranges. We measured the wave speed v_p and wavelength λ of the system for 625 combinations of α and $\bar{\gamma}$ spanning four orders of magnitude to construct the phase space (**Figures 5D,E**). We see that wavelength was predominantly determined by α, while wave speed depended on $\bar{\gamma}$. Traveling waves were observed for all values of α but clearly require non-zero protein degradation rate $\bar{\gamma}$ (**Figure 5F**).

We observed that wave speed was proportional to $\bar{\gamma}$, with $v_p \approx \bar{\gamma}/2$ (linear fit $v_p = 0.535\bar{\gamma} + 0.0214$) . Static

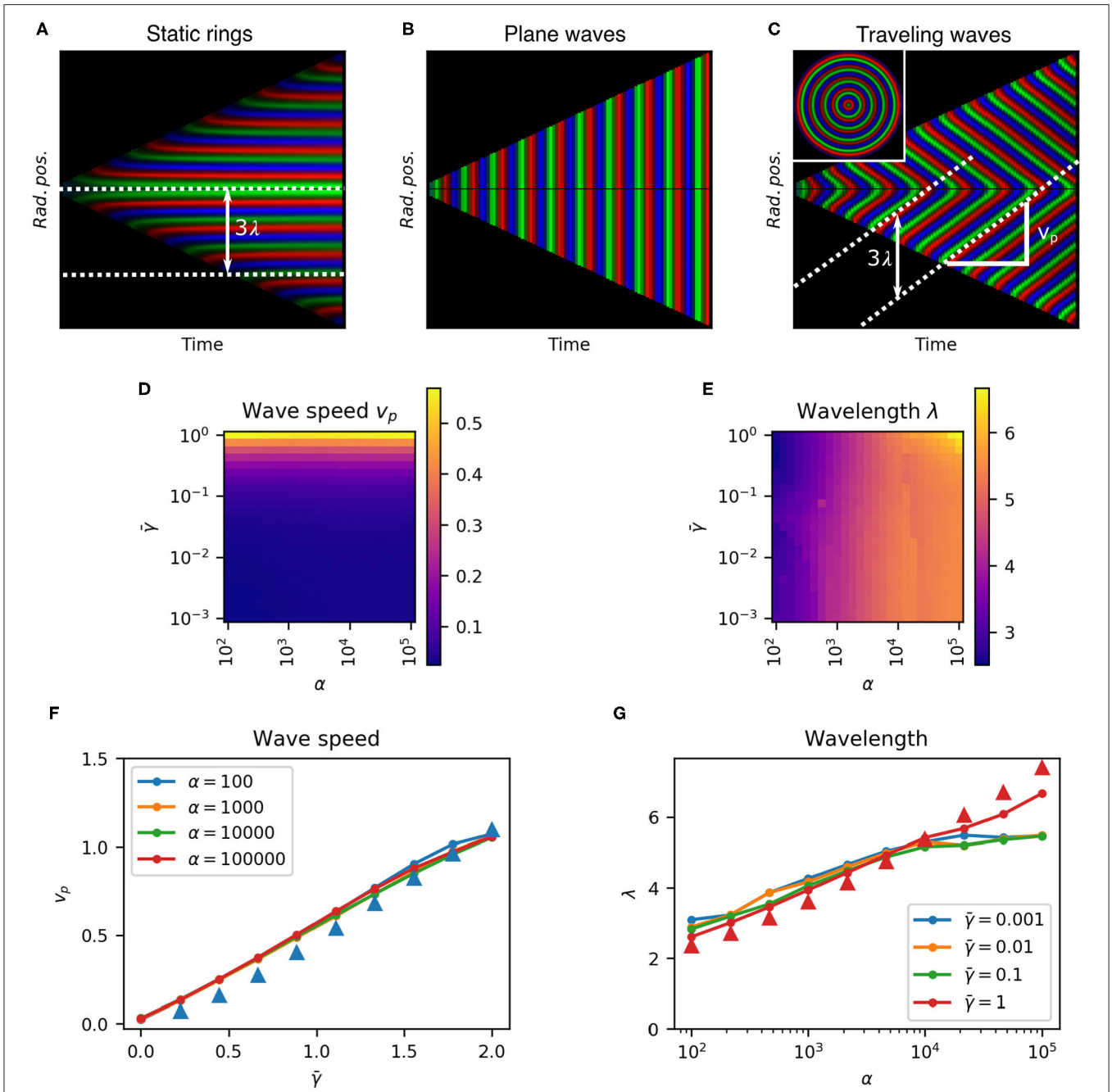


FIGURE 5 | Parameter dependence characterization for 1D model. **(A–C)** Kymographs of growing colonies show protein concentrations as a function of radial position over time, with no protein degradation (**A**, $\bar{\gamma} = 0$, $\alpha = 10^4$) we obtain static rings, with no growth rate dilution (**B**, $\bar{\gamma} = 1.5$, $\alpha = 10^4$) we see plane waves, and with growth rate dilution and protein degradation (**C**, $\bar{\gamma} = 1.5$, $\alpha = 10^4$) we have traveling waves. White dashed lines show wave trajectories. The distance between two trajectories is the wavelength λ, and the slope is the wave speed v_p. Inset in **(C)** is the whole colony at the end of the experiment. **(D,E)**. Heatmaps of wave speed v_p and wavelength λ, respectively over a range of α and $\bar{\gamma}$. **(F)**. Effect of $\bar{\gamma}$ on wave speed v_p at different α. **(G)**. Effect of α on wavelength λ at different $\bar{\gamma}$. Triangles in **(F,G)** show analytical estimates for $\alpha = 100$ and $\bar{\gamma} = 1$, respectively.

rings ($v_p = 0$) form when $\bar{\gamma} = 0$. As $\bar{\gamma} \to \infty$, as is the case at growth arrest ($\mu_0 = 0$), we saw earlier that the system can only form plane waves with all parts of the colony oscillating in phase, which corresponds to $v_p = \infty$. Wavelength was predominantly but weakly affected by maximum gene expression rate α (**Figure 5G**) following approximately $\alpha \approx 10^{\lambda-1}$ [from the linear fit $\lambda = 1.01 \log_{10}(\alpha) + 0.998$]. These results are consistent with our theoretical predictions based on the oscillation period from Equations (14) to (15) (**Figures 5F,G** triangles).

We demonstrate the tuning parameters α and $\bar{\gamma}$ above using kymographs in **Figure 6**. With no protein degradation ($\bar{\gamma} = 0$) the system produces static rings of gene expression following the phase of each repressor (**Figures 6A,B**). In the kymograph this spatio-temporal pattern is observed as horizontal stripes of consecutive red, green, and blue, representing the three repressors. This confirms the observation of fixed ring patterns in colonies hosting a repressilator with stable repressor proteins (Potvin-Trottier et al., 2016). The static ring patterns observed are therefore a special case of the more general traveling wave solution with velocity $v_p = 0$. These traveling waves are induced and modulated by protein degradation. In the intermediate case when protein degradation and growth are similar (**Figures 6C,D,F**) we see the clear emergence of a traveling wave solution. This spatio-temporal pattern is characterized by diagonal stripes in the kymograph, with steeper sloped lines indicating higher velocity of the waves (**Figure 5A**). At lower protein degradation rate ($\bar{\gamma} = 0.3$) we see traveling waves with lower velocity (**Figures 6C,D**). Hence protein degradation rate tunes the speed of the traveling waves. Changing α however does not affect the speed of the waves (**Figures 6A–F**) but does change the wavelength of the traveling waves resulting in more spatial rings at lower α values. We note that increasing α also stabilizes the oscillations as found by Osella and Lagomarsino (2013) and Potvin-Trottier et al. (2016) (**Figure 6** panels below each kymograph).

2.6. Growing Cell Colonies Generate Traveling Waves in Quantitative Agreement With Predictions

To test if these predictions hold in constrained growing microcolonies of cells we used our individual based biophysical model of bacterial cell growth and division. We grew colonies from a single cell up to 60,000 cells, tracking each cell's motion and protein expression levels according to Equation (11) without the growth rate term (dilution was computed by the biophysical model). The results show, as predicted, the formation of symmetrical rings relative to the center of the colony (**Figure 7A**, **Supplementary Material**, **Video 1** and **Video 2**).

In order to test the dependency of wavelength and wave speed on protein degradation and maximal expression rate, we simulated a range of $\bar{\gamma}$ and α (kymographs in **Figures 7B–E**). Our findings matched with the prediction of the 1D model; no waves were formed for $\bar{\gamma} = 0$, wave speed increased when $\bar{\gamma}$ was increased, and wavelength increased when α was increased. The spatio-temporal dynamics of each repressor is regulated by protein dilution ($\bar{\gamma}$), which moves the system from fixed rings (**Figure 7B**) to an oscillatory behavior which gives rise to traveling waves with different wavelengths controlled by maximum expression levels and speeds controlled by protein degradation (**Figures 7C–E**).

Since we tracked every cell as they grow, replicate, and express proteins (**Figures 7B–E** right column) we could follow the dynamics of individual cells as they move through the colony, changing their growth rate depending on their mechanical environment (**Figures 7B–E** middle column). We selected central and peripheral cells for each of the colonies to study the most restricted and the most unconstrained cells. For colonies with traveling waves the constrained non-growing central cells oscillated with constant frequency and phase. Cells starting at the periphery of the colony however experience changes in growth rate as they move away from the colony edge that cause a sharp decrease in frequency and a resulting phase offset with respect to the central cell. We found that cells in the periphery exhibit higher frequency oscillations compared to central cells, and that difference is increased when increasing $\bar{\gamma}$ (**Figures 7B–E** central column). This is consistent with our theoretical prediction that growth rate reduction increases the period of oscillations, causing a phase offset between the interior and peripheral regions of the colony.

The wavelength and wave speed obtained from growing microcolonies was closely correlated to the predictions of our simple model (Pearson's correlation coefficient 0.983 for wavelength and 0.999 for wave speed, **Figure 8**). The length scale of wave speed and wavelength is set by r_0, hence fitting a linear model between the predicted and simulated speed and wavelengths gives an estimate of r_0 for the growing colony. From the wave speed values we obtained $r_0 = 11.6$ and from wavelength $r_0 = 9.01$, which is in close agreement with that estimated from the growth rate distribution of colonies (**Figure 2**). Our one dimensional model predicts that the front velocity of the colony should be $v_{front} = r_0/2$. From wave speed we obtain a value of $v_{front} = 5.80$ and from wavelength $v_{front} = 4.51$, which is extremely close to the estimated value of $v_{front} = 5.00$ for our individual based model (**Figure 2**).

2.7. Manipulating Mechanical Growth Constraints to Control Pattern Formation

Microfabricated cell culture devices and microfluidics provide fine control over the mechanical as well as biochemical conditions in which cells grow. As well as providing fresh nutrients via flow, maintaining cells in steady state, these devices provide techniques to physically constrain cell growth and therefore another mode of design of spatio-temporal patterns induced by growth rate heterogeneity as we have demonstrated. Commonly microfluidic devices are designed to constrain cells to a monolayer, while allowing loading of seed cells into a chamber or channel (**Figures 9A,B**). We imposed two such constraints on our colonies. In a long thin channel ($400 \times 20 \times 1$ cell diameters) cells form one dimensional traveling waves directed along the channel axis (**Figure 9A**, See **Supplementary Material, Video 4, Video 5, Video 6**). When constrained to a chamber ($80 \times 80 \times 1$ cell diameters) we observed the emergence of traveling waves during unconstrained growth (**Figure 9B**, $t+4$, See **Supplementary Material, Video 3**). These waves were sustained over long time periods after the cells became constrained and stopped growing altogether (**Figure 9B**, $t + 8$, $t + 12$). Growth is necessary to form traveling waves but the established phase offsets between different radial positions are maintained after growth arrest, continuing to produce

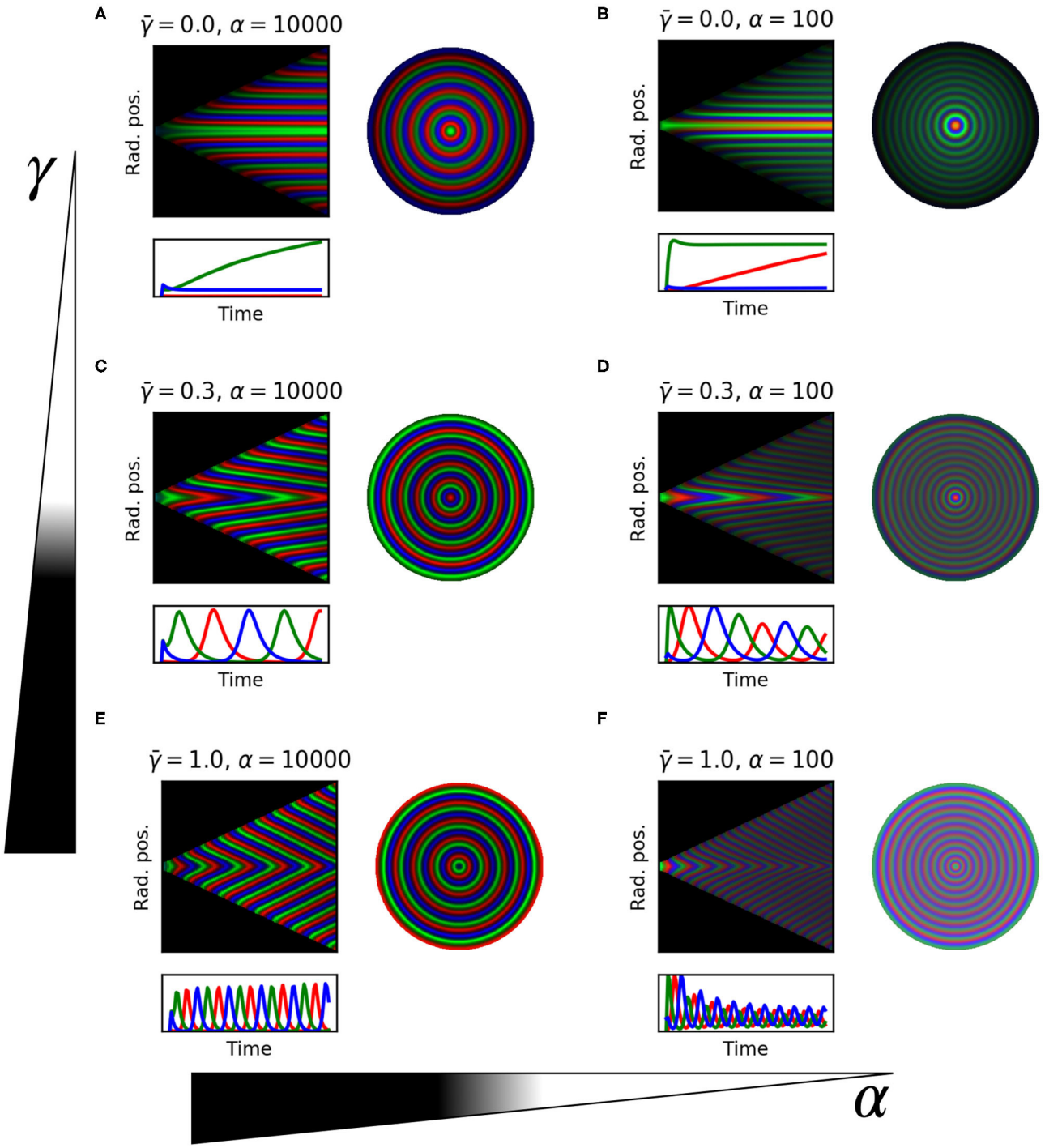


FIGURE 6 | 1D model simulations for selected parameters. Each panel shows: kymograph of the emerged pattern, where stripes show wave trajectories; trace of the three repressor concentrations over time in the center of the colony; the whole colony showing the final ring pattern. **(A)** $\bar{\gamma} = 0.0$, α = 10,000. **(B)** $\bar{\gamma} = 0.0$, α = 100. **(C)** $\bar{\gamma} = 0.3$, α = 10,000. **(D)** $\bar{\gamma} = 0.3$, α = 100. **(E)** $\bar{\gamma} = 1.0$, α = 10,000. **(F)** $\bar{\gamma} = 1.0$, α = 100.

traveling waves. The history of the shape of wavefront is therefore retained in the pattern. Finally, we demonstrate that growth rate heterogeneity in three dimensional colonies also generates traveling waves as layers (**Figure 9C**), showing that the spatio-temporal pattern is not specific to monolayers (see **Supplementary Material**, **Video 7** and **Video 8**).

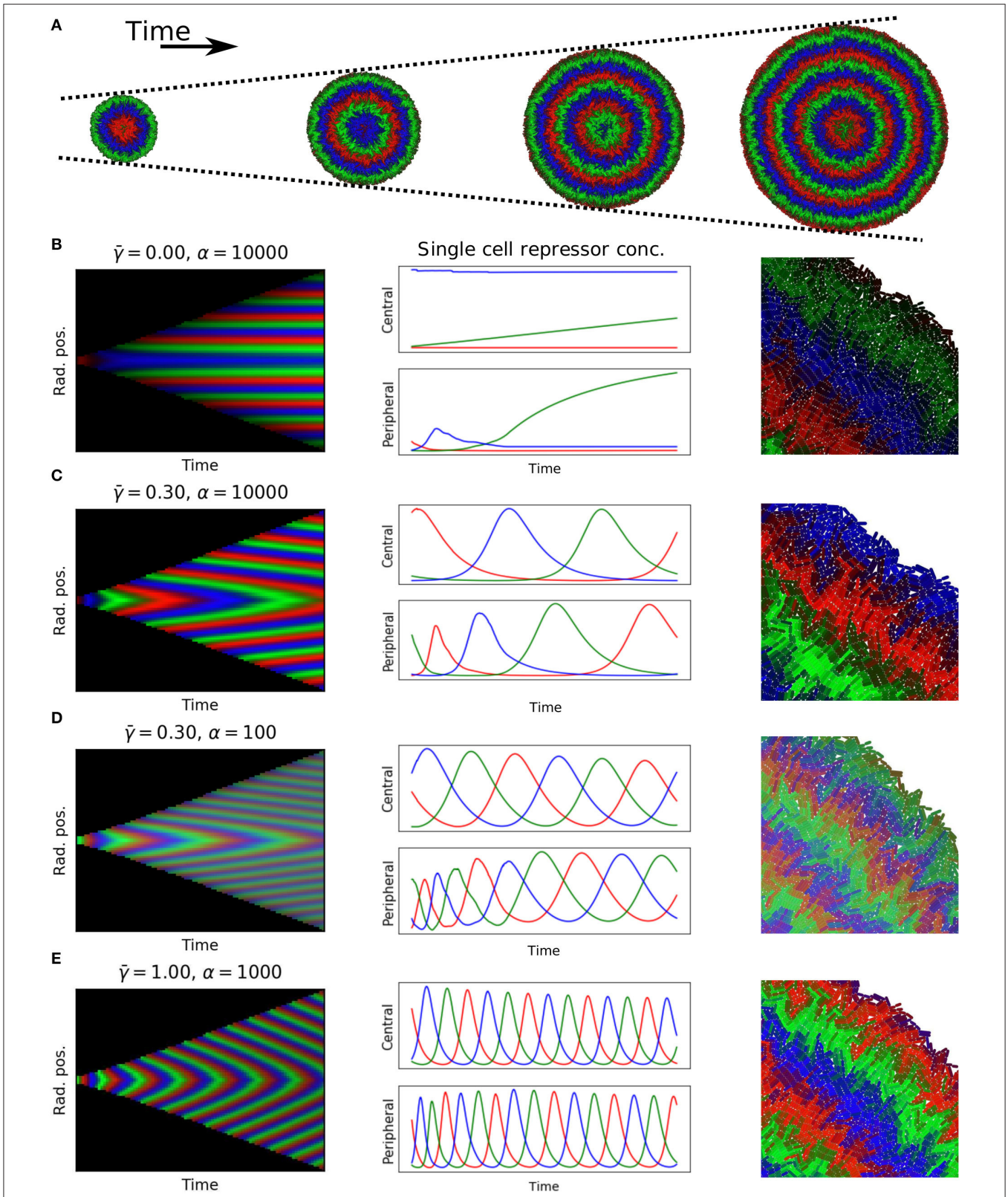


FIGURE 7 | Simulations of growing colonies. **(A)** Colonies with 5,000, 20,000, 35,000, and 50,000 cells, equally spaced 9.3 doubling times apart, with $\bar{\gamma} = 0.3$, α=10.000. **(B–E)** Each panel shows: kymograph of repressor concentrations (51 doublings, approximately 60,000 cells); time dynamics for central cell and peripheral cell in colony; close up of edge of colony at end of experiment. Parameters: **(B)** $\bar{\gamma} = 0$, $\alpha = 10,000$. **(C)** $\bar{\gamma} = 0.3$, $\alpha = 10,000$. **(D)** $\bar{\gamma} = 0.3$, $\alpha = 100$. **(E)** $\bar{\gamma} = 1.0$, $\alpha = 1,000$.

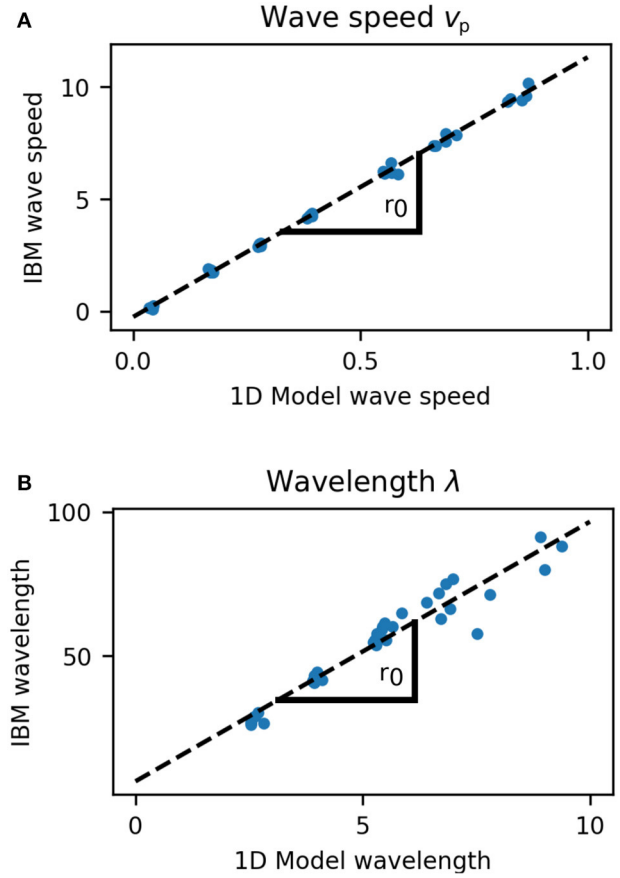


FIGURE 8 | Comparison of wave speed **(A)** and wavelength **(B)** between 1D model and growing cell colonies (IBM). The length scale of wave speed and wavelength is set by r_0 and the slope of these plots gives estimates for r_0 as 11.6 (wave speed) and 9.01 (wavelength) cell diameters. Speed and wavelength of traveling waves were closely correlated between the models, with Pearson's correlation coefficient 0.999 for wave speed, and 0.983 for wavelength.

3. DISCUSSION

Here we demonstrated how biophysical constraints on growth can induce spatio-temporal pattern formation from a simple genetic circuit. By coupling the repressilator (Potvin-Trottier et al., 2016) to a heterogeneous growth rate pattern via protein dilution we generated emergent traveling waves of gene expression. These traveling waves can be characterized by two properties; the wavelength and the wave speed. These properties are determined by two simple parameters that are feasible to control experimentally; the protein degradation rate, which controls the wave speed, and the maximal protein expression rate, which controls the wavelength. Our results make quantitative and qualitative predictions about the spatio-temporal patterns produced by the repressilator in growing cell colonies.

Our analysis predicts that traveling waves will be observed if the ratio of protein degradation to growth rate $\bar{\gamma} = \gamma/\mu_0$, is sufficiently high for the waves to form in the time of the experiment. For $\bar{\gamma} = 0$ we predict the formation of static rings of gene expression as observed in experiments (Potvin-Trottier et al., 2016), however we show here that this pattern could be generated purely by protein dilution and does not require cells to transition into stationary phase. We show that increasing $\bar{\gamma}$, which means increasing protein degradation rate or decreasing growth rate, will increase the speed of the waves. This could be achieved by choosing one of several protein degradation tag sequences that target the proteins for proteolysis (Purcell et al., 2012). Further, our model suggests that increasing maximum protein expression rate α, for example by choosing a more efficient ribosome binding site (RBS) (Salis et al., 2009) will increase the wavelength of the pattern. We parameterize simple empirical models for the effects of each of these genetic design modifications; $\log_{10}(\alpha) = \lambda - 1$ and $v_p = \bar{\gamma}/2$. Thus, we have effectively generated a quantitative datasheet for the repressilator gene circuit topology operating in a simple microcolony chassis.

We derive a simple model of coupling genetic circuits to growth rate via protein dilution, and show that it accurately predicts traveling wave patterns in growing cell colonies, their speed, and wavelength. The model also accurately predicts the colony front velocity. The mathematical and computational approach outlined here is not specific to the repressilator nor to bacterial colonies and could make predictions about spatial patterns produced by other circuit topologies and chassis. Here we did not consider gene circuits that affect growth rate, for example by regulation of metabolism, which may produce more complex spatial patterns (Nuñez et al., 2017), however it could be included in our framework leading to a more complex set of coupled differential equations. Thus, this analysis approach implements the rational design of spatio-temporal patterns of gene expression, enabling the design stage of the design-build-test-learn cycle.

Oscillators are important in regulation of multicellular systems and many studies have reproduced oscillations in synthetic genetic circuits by assembling different devices combining modular parts (Liu et al., 2015; Niederholtmeyer et al., 2015; Perez-Carrasco et al., 2018; Riglar et al., 2019). Studying and engineering synthetic oscillators can direct us to understand complex multicellular behaviors at multiple scales, in particular here the emergence of traveling waves of gene expression in populations of cells. There are a wide range of phenomena in which a key element to a developmental process is the appearance of a traveling wave of chemical concentration, mechanical deformation (Espeso et al., 2016), electrical or other type of signal (Murray, 2002). Two examples are the chemical and mechanical waves which propagate on the surface of many vertebrate eggs (Deneke and Di Talia, 2018). A developing embryo presents a large number of wave like events that appear after fertilization (Kimmel et al., 1995). Thus, one importance of this work is that we were able to rationally design and manipulate *in silico* genetic circuits to recapitulate such patterns with tunable wavelength and wave speed.

Noise is known to affect oscillators in various ways including stochastic coherence which makes the oscillations more consistent (Hilborn and Erwin, 2008), and may therefore

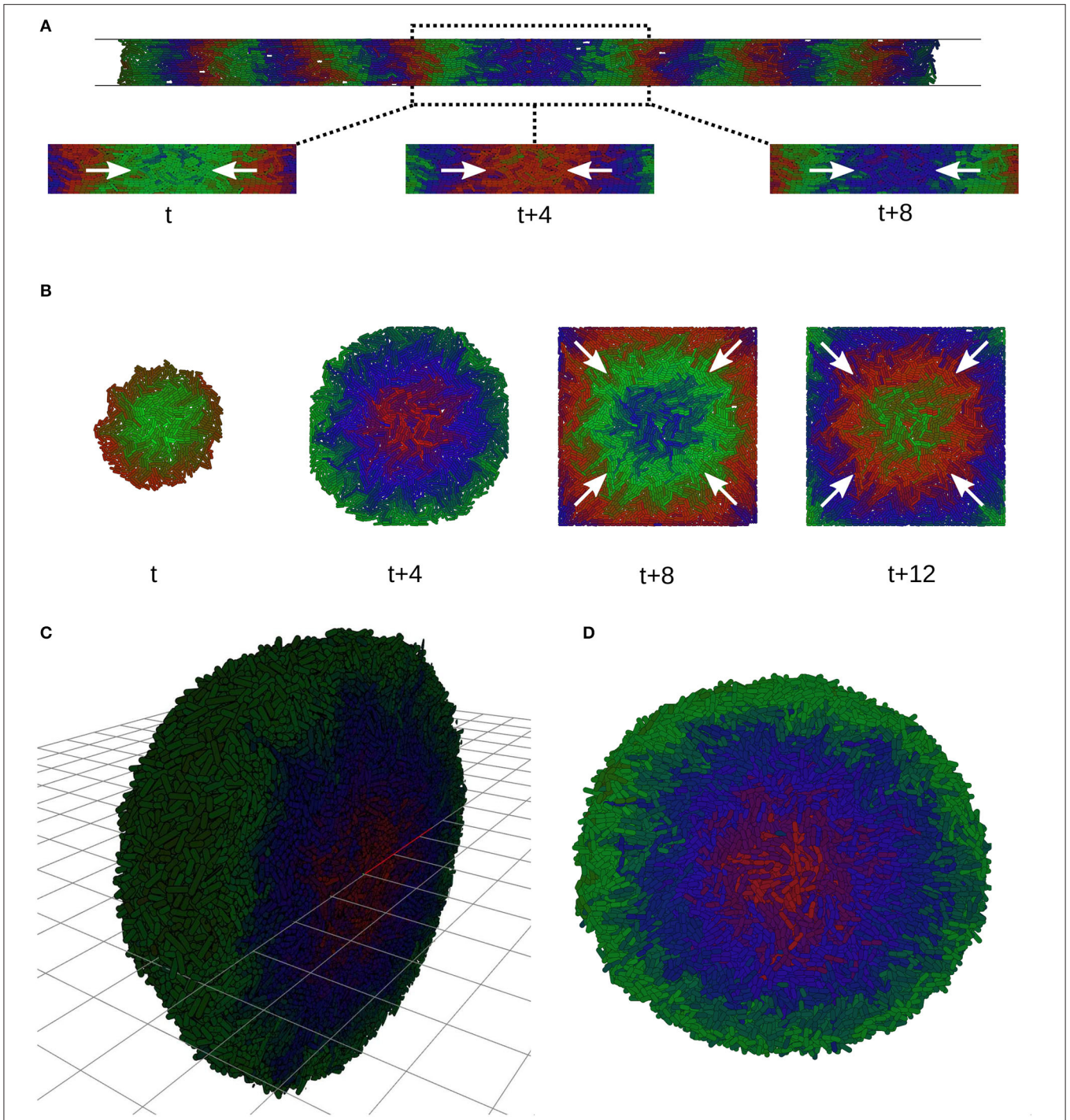


FIGURE 9 | Traveling waves in microfluidic devices. **(A)** Simulations of a monolayer of cells in a narrow infinite channel 20 cell diameters wide and one cell diameter tall. Traveling waves start at the sides and merge in the center (white arrows). **(B)** Growing colony in a square simulated microfluidic chamber. Before the colony reaches the constraints it grows with radial symmetry and initiates traveling waves (t, t+4). At t+12 the colony has used all the space and is constrained by chamber, growth arrest occurs, but the traveling waves continue (white arrows). **(C)** Spherical colonies grow when not constrained to a plane, forming traveling layers of gene expression. Image shows a cutaway of half the colony. **(D)** Transversal section of **(C)**.

stabilize spatio-temporal patterns to stochastic fluctuations in gene expression. We do not consider the role of noise in this study because at the parameter values we explore, the stochastic behavior approximates the continuous model, with regular and sustained synchronous oscillations (Woods et al., 2016) (**Figure S2**). We also note that since (Potvin-Trottier et al.,

2016) observed synchronous long-term oscillations that form ring patterns in growing colonies, cells must be synchronized on average over long length and time scales, and so noise is not likely to be important in the pattern formation process described here. However, it would be interesting to consider the role of noise in the generation of spatio-temporal patterns (Sagués et al., 2007; Zhou et al., 2008) due to lower gene copy or other circuit properties (Vilar et al., 2002; Lestas et al., 2010).

We reason that the traveling waves described here are generated by phase and frequency changes induced by reduction in growth rate as cells become more distant from the edge of the colony, but maintained by protein degradation. In the absence of protein degradation no traveling waves but simple static rings form (Potvin-Trottier et al., 2016). The phase differences are locked in as growth rate decays to zero, such that even after total growth arrest the traveling waves continue (**Figure 9B**). The scale of the waves, their speed and their wavelength are determined by r_0, the characteristic length scale of decay in growth rate. However, the radius of the colony also scales with r_0 and so the overall pattern is in a sense scale invariant, showing the same relative wavelength and speed for any exponential growth rate profile.

Our results show that the speed of traveling waves in growing bacterial colonies is approximately 10 cell diameters per doubling time (approximately $10\mu m$ per hour for *E. coli*) toward the colony center, but the colony border grows at only around 4 cell diameters per doubling. Hence gene expression information can be transferred faster via a traveling wave than by the physical transmission of cells. The ability to tune the wavelength λ and wave speed v_p of these patterns could enable design of novel cell-cell communication systems based on oscillatory signals. Further, coupling the oscillator to production of pulses of diffusing chemicals such as acyl-homoserine lactones (AHLs) could be used to enhance information transmission (Hopfield, 1974; Mangan et al., 2003). We note also that in a sense the traveling wave pattern, its speed and wavelength encode the history of the shape of the wavefront as the colony expands, which may be useful for example for information storage.

A fundamental result of this work is to demonstrate that mechanical constraint gives rise to higher order gene expression patterns in cell colonies, and provide such a system for analysis. There are a vast number of experimental conditions which could be created to induce different spatio-temporal patterns in such microcolonies. Microfluidics has shown to be of particular help to control mechanical constraints (Ruprecht et al., 2017), substrate stiffness (Wang et al., 2018), nutrients (Alnahhas et al., 2019), chemical inducers (Danino et al., 2010), cell-cell signaling (Alnahhas et al., 2019), and pattern formation (Kantsler et al., 2020). As we showed in **Figure 9**, controlling biophysical constraints using different channel layouts and mechanical properties of the substrate could produce different patterns of growth rate that give rise to structures that mimic different stages of the development of organisms (Johnson et al., 2017; Toda et al., 2018). A simple example is the one dimensional channel (**Figure 9A**) which mimics in a simple way an embryo growing along its axis and sending back waves of gene expression from the front of the cell population.

In summary we report here novel traveling wave spatio-temporal patterns resulting from the growth rate dependent dynamics of a repressilator genetic oscillator circuit. We developed an analytical framework to predict the spatio-temporal behavior of such genetic circuits in growing colonies. This framework allows multi-scale rational design of spatio-temporal patterns from genetic circuits and makes testable predictions for the synthetic biology design-build-test-learn cycle.

4. MATERIALS AND METHODS

Computation and analysis in this work were performed in Python (Van Rossum and Drake, 2009) with the use of the packages NumPy (Oliphant, 2006), SciPy (Virtanen et al., 2020), Pandas (McKinney, 2010), Jupyter (Pérez and Granger, 2007), Matplotlib (Hunter, 2007), Seaborn (Waskom et al., 2017), and NetworkX (Hagberg et al., 2018).

4.1. Individual Based Model

We grew colonies from 1 up to 60,000 cells, simulated using CellModeller (Rudge et al., 2012) with parameters $\Gamma = 10$ and $\Delta t = 0.05$. $\Gamma = \gamma_{cell}/\gamma_s$ is the ratio of cell stiffness to substrate stiffness, which we estimate order of magnitude 10 (see **Supplementary Material**). Briefly, CellModeller simulates cells as rods extending along their axis but otherwise rigid. As cells expand the resulting constraint energy is minimized to find the new arrangement of cells. Cells experience viscous drag with the substrate (γ_s) and along their length axis (γ_{cell}), and divide when they reach a target length set to $l_0 = 3.5$ cell diameters. At division the cell is divided into two equal sized rods, which are randomly perturbed slightly in their axis orientation. Cells were constrained to lie in a plane, except in **Figure 9C** in which cells grew in three dimensions.

Colonies were grown for approximately 48 doubling times and the final radius of the colonies was approximately ∼230 cell diameters. Since these simulations correspond to *E. Coli* cells, these units represents approximately $\sim 230\mu m$. Simulations were stored in a file for each time step. This file contains information about the state of each cell present in the colony, including the position, protein concentrations, growth rate, volume, length, among other variables.

4.2. Colony Growth Analysis

Growth rate μ and radial position R were obtained for each cell from 3 growing colonies from 5,000 up to 60,000 cells. At each time point the colony radius $R_{max}(t)$ was calculated and divided into n bins of size $\Delta r = 5$ cell diameters from the edge b_0 to the center b_n. The growth rates μ of all cells with $r \in [b_n, b_{n+1})$ were averaged to get $\langle \mu \rangle$. An exponential of the form $e^{-r(t)k}$ was fitted to $\langle \mu \rangle$ at each time, obtaining k at colonies with different R_{max}. A linear model was fitted to the colony radius $R_{max}(t)$ when $t > 10$ to compute the front velocity v_{front} using numpy.polyfit.

4.3. Kymograph Construction for 1D Model

To obtain values of $p_i(r, t)$ we integrate Equation (11). Starting from some initial colony radius R_0 we initialize $p_i(r, 0)$ for r a regularly spaced lattice on $(0, R_0)$. We use $p_2(r, 0) = 5$ for all r, that is homogeneous expression of only p_2. At each step of an

explicit Euler integration scheme we find the new cell positions and construct a new regularly spaced lattice in $(0, R(t)) = (0, R_0 + t/2)$ and interpolate $p_i(r, t + \Delta t)$ onto this grid before taking the next integration step. The algorithm is as follows:

1. Initialize $r(0)$ as a regular lattice on $(0, R_0)$ and $p_i^*(r, 0)$ to some values.
2. Compute $p_i(r, t + \Delta t)$ by an explicit Euler step of Equation (11) , and $r(t + \Delta t)$ using Equation (6).
3. Compute a new regular mesh $r'(t + \Delta t)$ on $(0, R(t + \Delta t))$.
4. Interpolate the protein concentrations to get $p_i(r', t + \Delta t)$.
5. Set $t \to t + \Delta t$, and $r(t + \Delta t) \to r'(t + \Delta t)$, and repeat from step 2.

At the end of this procedure we have constructed a set of samples $p_i(r, t)$ which we then interpolate to form the kymograph.

4.4. Dynamical Simulations of Gene Expression

Using the file stored for each simulation in the IBM, we have a representation of the biophysical model decoupled from the genetic circuit. Using these data we performed simulations of the gene expression model derived in Equation (11). In order to keep updating the state of the cells, which is affected by cell division, we constructed a graph of parent-child relations. Thus, we integrate Equation (11) forward using explicit Euler method between each state of the biophysical model. One assumption made is that when a cell divides the children inherit the value of the protein concentration his parent. We assume the number of proteins divides equally between the two cells, as does the volume of the cell, keeping protein concentration constant. Resultant simulations then serialized to a JSON file. These files were later used to perform analysis and create visual representations.

4.5. Kymograph Construction for Individual Based Simulations

Using the JSON file obtained in the temporal simulation of gene expression with the biophysical model, we generated positions and growth rates of cells. Then we binned cells according to their radial position using bin size $\Delta R = 5$. Finally we take the average protein concentration in each bin and repeat for all time steps to get $p_i(R, t)$.

4.6. Wave Speed Estimation

First we take each row of the kymograph and identify the radial peaks (scipy.signal.find_peaks) in each protein concentration for each time step. Next the peaks are paired with the nearest peak in the previous time step, and the average distance between them used to calculate the wave speed as $v = \langle \Delta x \rangle / \delta t$, where $\langle \Delta x \rangle$ is the mean peak to peak distance and δt is the simulation time step.

4.7. Wavelength Estimation

In order to estimate the wavelength λ of the traveling waves we note that $\lambda = (v_p + v_{front})T$, where v_p is the wave speed, T is the period of oscillations, and $v_{front} = \frac{1}{2}$ is the velocity of the colony edge or wavefront. To estimate the oscillation period we take the leading edge of the colony and compute the peaks in its time varying protein concentration $p_i(r = 0, t)$. Then as above we estimate the period as the mean of the peak to peak times so that $T = \langle \Delta t \rangle$. The wave speed is taken from the calculations described above, and the resulting estimate for wavelength is $\lambda = (v_p + \frac{1}{2})T = (v_p + \frac{1}{2})\langle \Delta t \rangle$.

4.8. Analytical Estimates of Wavelength and Wave Speed

We used Equations (14)–(15) to estimate the wavelength and wave speed of traveling waves that the repressilator would produce with an exponential growth rate profile (Equation 1). First we numerically integrated Equation (11) with fixed growth rate $\bar{\mu}$. For each combination of parameters we simulated oscillations at the colony edge ($\bar{\mu} = 1$) and the colony interior ($\bar{\mu} = 0$), and measured the periods T_{ext} and T_{int} as described above. These values were then substituted into Equations (14)–(15) to compute the estimated wavelength and wave speed.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://www.github.com/synbiouc/spatialoscillator.

AUTHOR CONTRIBUTIONS

GY, GV, and TR designed the study and analyzed the data. MM, GY, GV, and TR wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

GV was supported by a scholarship from the Institute for Biological and Medical Engineering, Pontificia Universidad Católica de Chile. GY was supported by Beca Ayudante Doctorando scholarship from the Department of Chemical and Bioprocess Engineering, Pontificia Universidad Católica de Chile. TR, GV, GY, and MM acknowledge financial support from the National Agency for Research and Development (ANID)/PIA/ACT192015.

ACKNOWLEDGMENTS

We thank Gustavo Düring, Luca Ciandrini, Pascal Rogalla, and Ignacio Medina for helpful and stimulating discussions. We also thank the members of the Synthetic Biology Lab for their support and encouragement—Anibal Arce, Kevin Simpson, Tamara Matute, Isaac Nuñez, Fernán Federici, among others. A preprint of this work is available on bioRxiv (Yáñez Feliu et al., 2020).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.00893/full#supplementary-material

REFERENCES

Aboobaker, A. A., Tomancak, P., Patel, N., Rubin, G. M., and Lai, E. C. (2005). Drosophila microRNAs exhibit diverse spatial expression patterns during embryonic development. *Proc. Natl. Acad. Sci. U.S.A.* 102, 18017–18022. doi: 10.1073/pnas.0508823102

Alnahhas, R. N., Winkle, J. J., Hirning, A. J., Karamched, B., Ott, W., Josić, K., et al. (2019). Spatiotemporal dynamics of synthetic microbial consortia in microfluidic devices. *ACS Synth. Biol.* 8, 2051–2058. doi: 10.1021/acssynbio.9b00146

Andersen, K. B., and von Meyenburg, K. (1980). Are growth rates of *Escherichia coli* in batch cultures limited by respiration? *J. Bacteriol.* 144, 114–123. doi: 10.1128/JB.144.1.114-123.1980

Ausländer, D., Ausländer, S., Pierrat, X., Hellmann, L., Rachid, L., and Fussenegger, M. (2018). Programmable full-adder computations in communicating three-dimensional cell cultures. *Nat. Methods* 15:57. doi: 10.1038/nmeth.4505

Chan, C. J., Heisenberg, C. P., and Hiiragi, T. (2017). Coordination of morphogenesis and cell-fate specification in development. *Curr. Biol.* 27, R1024–R1035. doi: 10.1016/j.cub.2017.07.010

Cookson, N. A., Mather, W. H., Danino, T., Mondragõn-Palomino, O., Williams, R. J., Tsimring, L. S., et al. (2011). Queueing up for enzymatic processing: correlated signaling through coupled degradation. *Mol. Syst. Biol.* 7, 1–9. doi: 10.1038/msb.2011.94

Danino, T., Mondragón-Palomino, O., Tsimring, L., and Hasty, J. (2010). A synchronized quorum of genetic clocks. *Nature* 463, 326–330. doi: 10.1038/nature08753

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.* 9, 67–103. doi: 10.1089/10665270252833208

Deneke, V. E., and Di Talia, S. (2018). Chemical waves in cell and developmental biology. *J. Cell Biol.* 217, 1193–1204. doi: 10.1083/jcb.201701158

Elowitz, M. B., and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335–338. doi: 10.1038/35002125

Endy, D., and Brent, R. (2001). Modelling cellular behaviour. *Nature* 409, 391–395. doi: 10.1038/35053181

Espeso, D. R., Martínez-García, E., de Lorenzo, V., and Goñi-Moreno, Á. (2016). Physical forces shape group identity of swimming *Pseudomonas putida* cells. *Front. Microbiol.* 7:1437. doi: 10.3389/fmicb.2016.01437

Farrell, F., Hallatschek, O., Marenduzzo, D., and Waclaw, B. (2013). Mechanically driven growth of quasi-two-dimensional microbial colonies. *Phys. Rev. Lett.* 111, 1–5. doi: 10.1103/PhysRevLett.111.168101

Fraleigh, S. P., and Bungay, H. R. (1986). Modelling of nutrient gradients in a bacterial colony. *Microbiology* 132, 2057–2060. doi: 10.1099/00221287-132-7-2057

Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403, 339–342. doi: 10.1038/35002131

Gjorevski, N., and Nelson, C. M. (2010). Endogenous patterns of mechanical stress are required for branching morphogenesis. *Integr. Biol.* 2, 424–434. doi: 10.1039/c0ib00040j

Gorochowski, T. E. (2016). Agent-based modelling in synthetic biology. *Essays Biochem.* 60, 325–336. doi: 10.1042/EBC20160037

Grant, M. A., Wacław, B., Allen, R. J., and Cicuta, P. (2014). The role of mechanical forces in the planar-to-bulk transition in growing *Escherichia coli* microcolonies. *J. R. Soc. Interface* 11:20140400. doi: 10.1098/rsif.2014.0400

Green, A. A., Kim, J., Ma, D., Silver, P. A., Collins, J. J., and Yin, P. (2017). Complex cellular logic computation using ribocomputing devices. *Nature* 548, 117–121. doi: 10.1038/nature23271

Guélon, T., Mathias, J.-D., and Deffuant, G. (2012). Influence of spatial structure on effective nutrient diffusion in bacterial biofilms. *J. Biol. Phys.* 38, 573–588. doi: 10.1007/s10867-012-9272-x

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2018). "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, eds T. V. Gäel Varoquaux and J. Millman (Pasadena, CA), 11–15.

Healy, C. P., and Deans, T. L. (2019). Genetic circuits to engineer tissues with alternative functions. *J. Biol. Eng.* 13, 1–7. doi: 10.1186/s13036-019-0170-7

Hilborn, R. C., and Erwin, J. D. (2008). Stochastic coherence in an oscillatory gene circuit model. *J. Theor. Biol.* 253, 349–354. doi: 10.1016/j.jtbi.2008.03.012

Hopfield, J. J. (1974). Kinetic proofreading: a new mechanism for reducing errors in biosynthetic processes requiring high specificity. *Proc. Natl. Acad. Sci. U.S.A.* 71, 4135–4139. doi: 10.1073/pnas.71.10.4135

Hunter, J. D. (2007). Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* 9, 90–95. doi: 10.1109/MCSE.2007.55

Johnson, M. B., March, A. R., and Morsut, L. (2017). Engineering multicellular systems: using synthetic biology to control tissue self-organization. *Curr. Opin. Biomed. Eng.* 4, 163–173. doi: 10.1016/j.cobme.2017.10.008

Kantsler, V., McDonald, E. O., Kuey, C., Ghanshyam, M. J., and Asally, M. (2020). Pattern engineering of living bacterial colonies using meniscus-driven fluidic channels. 9, 1277–1283. doi: 10.1021/acssynbio.0c00146

Karig, D., Martini, K. M., Lu, T., DeLateur, N. A., Goldenfeld, N., and Weiss, R. (2018). Stochastic Turing patterns in a synthetic bacterial population. *Proc. Natl. Acad. Sci. U.S.A.* 115, 6572–6577. doi: 10.1073/pnas.1720770115

Khain, E., and Sander, L. M. (2006). Dynamics and pattern formation in invasive tumor growth. *Phys. Rev. Lett.* 96:188103. doi: 10.1103/PhysRevLett.96.188103

Kim, J., Yin, P., and Green, A. A. (2018). Ribocomputing: cellular logic computation using RNA devices. *Biochemistry* 57, 883–885. doi: 10.1021/acs.biochem.7b01072

Kimmel, C. B., Ballard, W. W., Kimmel, S. R., Ullmann, B., and Schilling, T. F. (1995). Stages of embryonic development of the zebrafish. *Dev. Dyn.* 203, 253–310. doi: 10.1002/aja.1002030302

Klumpp, S. (2011). Growth-rate dependence reveals design principles of plasmid copy number control. *PLoS ONE* 6:e20403. doi: 10.1371/journal.pone.0020403

Klumpp, S., Zhang, Z., and Hwa, T. (2009). Growth rate-dependent global effects on gene expression in bacteria. *Cell* 139, 1366–1375. doi: 10.1016/j.cell.2009.12.001

Lestas, I., Vinnicombe, G., and Paulsson, J. (2010). Fundamental limits on the suppression of molecular fluctuations. *Nature* 467, 174–178. doi: 10.1038/nature09333

Liu, J., Prindle, A., Humphries, J., Gabalda-Sagarra, M., Asally, M., Lee, D. Y. D., et al. (2015). Metabolic co-dependence gives rise to collective oscillations within biofilms. *Nature* 523, 550–554. doi: 10.1038/nature14660

Long, Z., Nugent, E., Javer, A., Cicuta, P., Sclavi, B., Lagomarsino, M. C., et al. (2013). Microfluidic chemostat for measuring single cell dynamics in bacteria. *Lab Chip* 13, 947–954. doi: 10.1039/c2lc41196b

Luo, N., Wang, S., and You, L. (2019). Synthetic pattern formation. *Biochemistry* 58, 1478–1483. doi: 10.1021/acs.biochem.8b01242

Mangan, S., Zaslaver, A., and Alon, U. (2003). The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. *J. Mol. Biol.* 334, 197–204. doi: 10.1016/j.jmb.2003.09.049

Matson, J., and Characklis, W. G. (1976). Diffusion into microbial aggregates. *Water Res.* 10, 877–885. doi: 10.1016/0043-1354(76)90022-1

Matsushita, M., and Fujikawa, H. (1990). Diffusion-limited growth in bacterial colony formation. *Phys. A* 168, 498–506. doi: 10.1016/0378-4371(90)90402-E

McKinney, W. (2010). "Data structures for statistical computing in Python," in *Proceedings of the 9th Python in Science Conference*, eds S. van der Walt and J. Millman (Austin, TX), 56–61.

Murray, J. D. (2002). *Mathematical Biology I. An Introduction, Vol. 17 of Interdisciplinary Applied Mathematics, 3 Edn.* New York, NY: Springer.

Neubauer, P., Lin, H., and Mathiszik, B. (2003). Metabolic load of recombinant protein production: inhibition of cellular capacities for glucose uptake and respiration after induction of a heterologous gene in *Escherichia coli*. *Biotechnol. Bioeng.* 83, 53–64. doi: 10.1002/bit.10645

Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., et al. (2015). Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* 4, 1–18. doi: 10.7554/eLife.09771

Nielsen, A. A., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., et al. (2016). Genetic circuit design automation. *Science* 352:aac7341. doi: 10.1126/science.aac7341

Nuñez, I. N., Matute, T. F., Del Valle, I. D., Kan, A., Choksi, A., Endy, D., et al. (2017). Artificial symmetry-breaking for morphogenetic engineering bacterial colonies. *ACS Synth. Biol.* 6, 256–265. doi: 10.1021/acssynbio.6b00149

Oliphant, T. E. (2006). *A Guide to NumPy, Vol. 1*. Cambridge, MA: Trelgol Publishing.

Osella, M., and Lagomarsino, M. C. (2013). Growth-rate-dependent dynamics of a bacterial genetic oscillator. *Phys. Rev. E* 87:012726. doi: 10.1103/PhysRevE.87.012726

Pérez, F., and Granger, B. E. (2007). IPython: a system for interactive scientific computing. *Comput. Sci. Eng.* 9, 21–29. doi: 10.1109/MCSE.2007.53

Perez-Carrasco, R., Barnes, C. P., Schaerli, Y., Isalan, M., Briscoe, J., and Page, K. M. (2018). Combining a toggle switch and a repressilator within the AC-DC circuit generates distinct dynamical behaviors. *Cell Syst.* 6, 521–530.e3. doi: 10.1016/j.cels.2018.02.008

Potvin-Trottier, L., Lord, N. D., Vinnicombe, G., and Paulsson, J. (2016). Synchronous long-term oscillations in a synthetic gene circuit. *Nature* 538, 514–517. doi: 10.1038/nature19841

Purcell, O., Grierson, C. S., Di Bernardo, M., and Savery, N. J. (2012). Temperature dependence of ssrA-tag mediated protein degradation. *J. Biol. Eng.* 6:10. doi: 10.1186/1754-1611-6-10

Riglar, D. T., Richmond, D. L., Potvin-Trottier, L., Verdegaal, A. A., Naydich, A. D., Bakshi, S., et al. (2019). Bacterial variability in the mammalian gut captured by a single-cell synthetic oscillator. *Nat. Commun.* 10, 1–12. doi: 10.1038/s41467-019-12638-z

Rudge, T. J., Federici, F., Steiner, P. J., Kan, A., and Haseloff, J. (2013). Cell polarity-driven instability generates self-organized, fractal patterning of cell layers. *ACS Synth. Biol.* 2, 705–714. doi: 10.1021/sb400030p

Rudge, T. J., Steiner, P. J., Phillips, A., and Haseloff, J. (2012). Computational Modeling of Synthetic Microbial Biofilms. *ACS Synth. Biol.* 1, 345–352. doi: 10.1021/sb300031n

Ruprecht, V., Monzo, P., Ravasio, A., Yue, Z., Makhija, E., Strale, P. O., et al. (2017). How cells respond to environmental cues - insights from bio-functionalized substrates. *J. Cell Sci.* 130, 51–61. doi: 10.1242/jcs.196162

Sagués, F., Sancho, J. M., and García-Ojalvo, J. (2007). Spatiotemporal order out of noise. *Rev. Modern Phys.* 79, 829–882. doi: 10.1103/RevModPhys.79.829

Salis, H. M., Mirsky, E. A., and Voigt, C. A. (2009). Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol.* 27, 946–950. doi: 10.1038/nbt.1568

Santos-Moreno, J., and Schaerli, Y. (2019). Using synthetic biology to engineer spatial patterns. *Adv. Biosyst.* 3, 1–15. doi: 10.1002/adbi.201800280

Sickle, J. J., Ni, C., Shen, D., Wang, Z., Jin, M., and Lu, T. (2020). Integrative circuit-host modeling of a genetic switch in varying environments. *Sci. Rep.* 10:8383. doi: 10.1038/s41598-020-64921-5

Smith, W. P., Davit, Y., Osborne, J. M., Kim, W., Foster, K. R., and Pitt-Francis, J. M. (2017). Cell morphology drives spatial patterning in microbial communities. *Proc. Natl. Acad. Sci. U.S.A.* 114, E280–E286. doi: 10.1073/pnas.1613007114

Steiner, P. J., Williams, R. J., Hasty, J., and Tsimring, L. S. (2016). Criticality and adaptivity in enzymatic networks. *Biophys. J.* 111, 1078–1087. doi: 10.1016/j.bpj.2016.07.036

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S., and Hasty, J. (2008). A fast, robust and tunable synthetic gene oscillator. *Nature* 456, 516–519. doi: 10.1038/nature07389

Tamsir, A., Tabor, J. J., and Voigt, C. A. (2011). Robust multicellular computing using genetically encoded nor gates and chemical "wires." *Nature* 469, 212–215. doi: 10.1038/nature09565

Toda, S., Blauch, L. R., Tang, S. K., Morsut, L., and Lim, W. A. (2018). Programming self-organizing multicellular structures with synthetic cell-cell signaling. *Science* 361, 156–162. doi: 10.1126/science.aat0271

Tuson, H. H., Auer, G. K., Renner, L. D., Hasebe, M., Tropini, C., Salick, M., et al. (2012). Measuring the stiffness of bacterial cells from growth rates in hydrogels of tunable elasticity. *Mol. Microbiol.* 84, 874–891. doi: 10.1111/j.1365-2958.2012.08063.x

Van Rossum, G., and Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Velardo, M. J., Burger, C., Williams, P. R., Baker, H. V., López, M. C., Mareci, T. H., et al. (2004). Patterns of gene expression reveal a temporally orchestrated wound healing response in the injured spinal cord. *J. Neurosci.* 24, 8562–8576. doi: 10.1523/JNEUROSCI.3316-04.2004

Vicsek, T., Cserző, M., and Horváth, V. K. (1990). Self-affine growth of bacterial colonies. *Phys. A* 167, 315–321. doi: 10.1016/0378-4371(90)90116-A

Vilar, J. M., Kueh, H. Y., Barkai, N., and Leibler, S. (2002). Mechanisms of noise-resistance in genetic oscillators. *Proc. Natl. Acad. Sci. U.S.A.* 99, 5988–5992. doi: 10.1073/pnas.092133899

Vining, K. H., and Mooney, D. J. (2017). Mechanical forces direct stem cell behaviour in development and regeneration. *Nat. Rev. Mol. Cell Biol.* 18, 728–742. doi: 10.1038/nrm.2017.108

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Walt, S. J. V. D., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods.* 17, 261–272. doi: 10.1038/s41592-019-0686-2

Wang, W., Li, L., Ding, M., Luo, G., and Liang, Q. (2018). A microfluidic hydrogel chip with orthogonal dual gradients of matrix stiffness and oxygen for cytotoxicity test. *Biochip J.* 12, 93–101. doi: 10.1007/s13206-017-2202-z

Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., et al. (2017). *Seaborn: v0.8.1*. Stanford, CA.

Winkle, J. J., Igoshin, O. A., Bennett, M. R., Josić, K., and Ott, W. (2018). Modeling mechanical interactions in growing populations of rod-shaped bacteria. *Phys. Biol.* 14:e110742. doi: 10.1101/110742

Wong, A., Wang, H., Poh, C. L., and Kitney, R. I. (2015). Layering genetic circuits to build a single cell, bacterial half adder. *BMC Biol.* 13:40. doi: 10.1186/s12915-015-0146-0

Woods, M. L., Leon, M., Perez-Carrasco, R., and Barnes, C. P. (2016). A statistical approach reveals designs for the most robust stochastic gene oscillators. *ACS Synth. Biol.* 5, 459–470. doi: 10.1021/acssynbio.5b00179

Xie, M., and Fussenegger, M. (2018). Designing cell function: assembly of synthetic gene circuits for cell biology applications. *Nat. Rev. Mol. Cell Biol.* 19, 507–525. doi: 10.1038/s41580-018-0024-z

Yáñez Feliu, G. A., Vidal Peña, G., Mũñoz Silva, M. A., and Rudge, T. J. (2020). Novel tunable spatio-temporal patterns from a simple genetic oscillator circuit. *bioRxiv [Preprint]*. doi: 10.1101/2020.07.06.190199

Yeung, E., Dy, A. J., Martin, K. B., Ng, A. H., Del Vecchio, D., Beck, J. L., et al. (2017). Biophysical constraints arising from compositional context in synthetic gene networks. *Cell Syst.* 5, 11–24.e12. doi: 10.1016/j.cels.2017.06.001

Zhou, T., Zhang, J., Yuan, Z., and Chen, L. (2008). Synchronization of genetic oscillators. *Chaos* 18:037126. doi: 10.1063/1.2978183

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 11 September 2020
doi: 10.3389/fbioe.2020.01009

The Synthetic Biology Open Language (SBOL) Version 3: Simplified Data Exchange for Bioengineering

James Alastair McLaughlin[1], Jacob Beal[2], Göksel Mısırlı[3], Raik Grünberg[4], Bryan A. Bartley[2], James Scott-Brown[5], Prashant Vaidyanathan[6], Pedro Fontanarrosa[7], Ernst Oberortner[8], Anil Wipat[1], Thomas E. Gorochowski[9] and Chris J. Myers[10]**

[1] School of Computing, Newcastle University, Newcastle-upon-Tyne, United Kingdom, [2] Raytheon BBN Technologies, Cambridge, MA, United States, [3] School of Mathematics and Computing, Keele University, Keele, United Kingdom, [4] Computational Bioscience Research Center, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, [5] Nuffield Department of Population Health, University of Oxford, Oxford, United Kingdom, [6] Microsoft Research, Cambridge, United Kingdom, [7] Department of Biomedical Engineering, University of Utah, Salt Lake City, UT, United States, [8] Lawrence Berkeley National Laboratory, DOE Joint Genome Institute, Berkeley, CA, United States, [9] School of Biological Sciences, University of Bristol, Bristol, United Kingdom, [10] Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, CO, United States

OPEN ACCESS

Edited by:
Pablo Carbonell,
Universitat Politècnica de València, Spain

Reviewed by:
Mario Andrea Marchisio,
Tianjin University, China
Matthew Wook Chang,
National University of Singapore, Singapore
Chris Barnes,
University College London, United Kingdom

***Correspondence:**
Thomas E. Gorochowski
thomas.gorochowski@bristol.ac.uk
Chris J. Myers
myers@ece.utah.edu

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: 29 May 2020
Accepted: 31 July 2020
Published: 11 September 2020

Citation:
McLaughlin JA, Beal J, Mısırlı G, Grünberg R, Bartley BA, Scott-Brown J, Vaidyanathan P, Fontanarrosa P, Oberortner E, Wipat A, Gorochowski TE and Myers CJ (2020) The Synthetic Biology Open Language (SBOL) Version 3: Simplified Data Exchange for Bioengineering. Front. Bioeng. Biotechnol. 8:1009. doi: 10.3389/fbioe.2020.01009

The Synthetic Biology Open Language (SBOL) is a community-developed data standard that allows knowledge about biological designs to be captured using a machine-tractable, ontology-backed representation that is built using Semantic Web technologies. While early versions of SBOL focused only on the description of DNA-based components and their sub-components, SBOL can now be used to represent knowledge across multiple scales and throughout the entire synthetic biology workflow, from the specification of a single molecule or DNA fragment through to multicellular systems containing multiple interacting genetic circuits. The third major iteration of the SBOL standard, SBOL3, is an effort to streamline and simplify the underlying data model with a focus on real-world applications, based on experience from the deployment of SBOL in a variety of scientific and industrial settings. Here, we introduce the SBOL3 specification both in comparison to previous versions of SBOL and through practical examples of its use.

Keywords: synthetic biology, data standards, data exchange, knowledge representation, SBOL

1. INTRODUCTION

Synthetic biology builds upon advances in genetics, molecular biology, metabolic engineering, and other related disciplines by applying principles such as modularization, standardization, and a design-build-test-learn workflow to enable the engineering of biological systems, just as software engineering does to the design of computer programs (Endy, 2005). The design-build-test-learn workflow is heavily dependant on data exchange. A standardized knowledge representation, or data standard, for exchanging information is critical from the initial stage of knowledge gathering—where data about existing biological parts and systems must be integrated into a common model—through to the entire design-build-test-learn lifecycle. Data standards are also crucial for the effective dissemination of final products or the publication of novel designs to ensure precise and unambiguous details of a system are accessible for oversight, management, and potential future re-use.

The unique requirements of synthetic biology present a major barrier to the development of such standards. Biological designs often involve engineering activities across a wide range of scales, from single molecules to genes, pathways, strains, and complex multi-cellular systems. Consequently, synthetic biologists need to exchange a wide variety of information, including the intended behavior of the system and actual experimental measurements. Information being exchanged also often covers multiple aspects of a design, including nucleic acid sequences (e.g., the sequence that encodes an enzyme or transcription factor), molecular interactions that a designer intends to result from the introduction of a chosen sequence (e.g., chemical modification of metabolites or regulation of gene expression), as well as details regarding the construction of the final engineered strain (e.g., nucleic acid synthesis, assembly, and the transformation of a chosen cell type) and associated experiments and data. All of these diverse perspectives need to be effectively integrated to facilitate the effective engineering of biological systems.

While there already exist many computational representations of biological entities, these are almost all designed for the annotation of natural systems and therefore struggle to describe the specifics of engineered designs. For example, simple formats for representing sequences such as FASTA (Pearson, 1990) are focused purely at the scale of nucleic or amino acid sequences and cannot capture higher-level aspects of a design (e.g., a sequence composition from constituent sub-sequences/parts). More sophisticated formats such as GenBank (Benson et al., 2013) or GFF (Stein, 2013) provide a flat representation of sequence features that is well-suited to describing natural systems, but again are fundamentally focused on annotation at the nucleic or amino acid level and are therefore unable to effectively represent functional relationships between regions of a sequence (e.g., description of protein-protein interactions) and localization (e.g., intracellular transport, cell-to-cell communication), not to mention engineering concepts such as interfaces and specifications or information capturing the intent of the designer.

The *Synthetic Biology Open Language* (SBOL) has been developed to address these challenges. SBOL is a standard to support the specification and exchange of biological design information in synthetic biology (Galdzicki et al., 2014), following an open community process involving both "wet" bench scientists and "dry" scientific modelers and software developers across academia, industry, and other institutions (see Methods). One of the primary aims motivating the development of SBOL is the need to make the knowledge involved in the synthetic biology lifecycle computationally tractable and therefore amenable to process automation. The research question of how domain knowledge can be decomposed into a form accessible to computational methods is long-established in computer science. The Resource Description Framework (RDF) (W3C, 2014) is a data model formalized by the World Wide Web Consortium (W3C) to describe named properties and their values that is already widely used by the bioinformatics community, with some of the largest biological datasets such as UniProt and PubChem publishing official RDF versions (Redaschi and UniProt Consortium, 2009; Fu et al., 2015). SBOL is built upon RDF, and is also backed by a formally defined ontology (Misirli et al., 2019), allowing design data to be machine-navigable as a knowledge graph.

Since its initial publication in 2011, SBOL has become the recommended format for engineered nucleic acid constructs in ACS Synthetic Biology (Hillson et al., 2016), and is supported by many biological design tools. For instance, Eugene (Bilitchenko et al., 2011; Oberortner et al., 2014; Oberortner and Densmore, 2015), GEC (Pedersen and Phillips, 2009; Dalchau et al., 2019), Cello (Vaidyanathan et al., 2015; Nielsen et al., 2016), GenoCAD (Czar et al., 2009), ShortBOL (Crowther et al., 2020), and GeneTech (Baig and Madsen, 2017) provide computational frameworks for combinatorial design space exploration, where users can specify structural, functional, and performance constraints. The outputs generated by these tools in SBOL can then be directly used by DNA assembly planning software tools such as BOOST (Oberortner et al., 2017), Raven (Appleton et al., 2014), j5 (Hillson et al., 2012), and DeviceEditor (Chen et al., 2012) to automate the process of physically building DNA constructs. Tools such as iBioSim (Myers et al., 2009; Watanabe et al., 2018), MoSeC (Misirli et al., 2011), and SBOLDesigner (Zhang et al., 2017) support the same SBOL data format and support the modeling, analysis, and simulation of biosystems. There are also a number of data repositories, registries, and databases that support and store data in the SBOL format, such as SynBioHub (McLaughlin et al., 2018), SBOLme (Kuwahara et al., 2017), JBEI-ICE (Ham et al., 2012), and the Virtual Parts Repository (VPR) (Cooling et al., 2010; Hallinan et al., 2014; Misirli et al., 2014). The SBOL community has also developed a graphical language for the visualization of biological designs (Quinn et al., 2015; Beal et al., 2019), which has been used in combination with the data standard in tools such as Pigeon (Bhatia and Densmore, 2013), DNAPlotlib (Der et al., 2017; Bartoli et al., 2018), VisBOL (McLaughlin et al., 2016), Constellation, and SBOLCanvas. These tools help to visualize constructs in the computational synthetic biology space such as genetic circuits, biochemical components, and possible design spaces based on structural or functional constraints. There are many other examples that highlight the utility of the SBOL data exchange format to connect and integrate data to create a seamless computational workflow. For instance, Cello (Nielsen et al., 2016) adopted the concept of a User Constraint File (UCF) used in digital logic design to specify the library of genetic gates and the associated properties and meta-data required to synthesize combinational Boolean logic circuits. In addition to this UCF file, the same library is also available in SBOL format, which allows the data from Cello to be used in other tools and workflows as highlighted in a recent effort to use the Cello library and Virtual Parts Repository API to build computational models encoded in the *Systems Biology Markup Language* (SBML) (Hucka et al., 2003) that could be simulated using iBioSim (Misirli et al., 2018).

The first version of SBOL (Galdzicki et al., 2011) defined a simple data model for the description of engineered DNA components and their sequences. Since then, SBOL has evolved to support the capture of information at many different levels

of representation across entire synthetic biology workflows (**Figure 1**). In particular, the previous major revision, SBOL2 (Bartley et al., 2015; Roehner et al., 2016), generalized the data model to allow for designs to include not only DNA components, but also other molecular species such as RNAs, proteins, larger components of a system such as whole cells, and links to models encoded using complementary standards such as SBML (Hucka et al., 2003). The standard was also incrementally expanded with several minor revisions (Beal et al., 2016; Cox et al., 2018; Madsen et al., 2019b) to capture information about combinatorial design libraries, external file attachments, sequence construction, experimental tests, and measurements. Furthermore, by leveraging the Provenance Ontology (PROV-O) (Lebo et al., 2013), SBOL2 can capture provenance information to link and trace information and processes throughout the entire design-build-test-learn cycle.

The incremental expansion in the scope of SBOL2 over the past few years has resulted in a significant increase in the complexity of the SBOL data model and has revealed aspects of the representation that limited future developments. While SBOL Enhancement Proposals (SEPs) to address this complexity had been accepted by the community, they were considered too major for a 2.x release, and therefore the need for a new major iteration of SBOL became apparent.

Here, we present SBOL version 3 (SBOL3), a substantially simplified standard that addresses these limitations, building upon the experience of the SBOL community applying SBOL across scientific and industrial settings. This new version (Baig et al., 2020) provides for a more direct and elegant expression of the diverse types of biological design information in use today, while at the same time reducing the complexity of the data model, which helps simplify the development of supporting libraries and data exchange with compatible tools. SBOL3 is an attempt to learn from the application of the previous SBOL standards, take stock of new developments and directions in the field, and establish a strong foundation for improved data exchange and computational-accessibility across synthetic biology.

2. RESULTS

SBOL3 contains ten main top-level classes to support the various aspects of the design-build-test-learn workflow (**Figure 2**). In particular, designs can be expressed using the `Component`, `Sequence` and `CombinatorialDerivation` classes. The `Component` class is intended to be widely applicable across all scales of biodesign, and can be used to describe not only genetic designs, but also the design of other biological entities such as proteins, functional RNAs, strains, multicellular systems, media, and experimental samples. For those `Component`s that have a defined primary structure, such as nucleic acids and proteins, an instance of the `Sequence` class can be assigned. A `CombinatorialDerivation` allows one to specify a design pattern where individual `SubComponent`s can be selected from a set of variants.

Beyond design, the `Implementation` class corresponds to the build stage of the synthetic biology lifecycle and is used to represent physical entities such as a sample of plasmid, a stab of transformed bacteria, or an aliquot of liquid culture. The `Experiment` and `ExperimentalData` classes support the test stage, allowing for the linking of data generated during an experiment. The `Model` class associates learned information with a design. All of this information can be linked together using the `Activity` class from PROV-O (Lebo et al., 2013). For example, a *design* `Activity` may describe how a `Component` is designed from a `Model` description. A *build* `Activity` describes how an `Implementation` is constructed to the specification of a `Component` description. A *test* `Activity` describes how an `Experiment` is conducted using an `Implementation` artifact. Finally, a *learn* `Activity` may describe how a `Model` is updated using information from an `Experiment`. The `Collection` class has members which can be of any of these types or even `Collection`s themselves. Finally, all of these objects can refer to objects of the `Attachment` class, which is used for links to external data (images, spreadsheets, textual documents, experimental instrument outputs, etc.).

2.1. SBOL3 Components

The main design entity in SBOL3 is the `Component` class. **Figure 3** provides an overview of the classes used by or linked to by the `Component` class. The "structural" classes have existed in various forms since the original SBOL1 specification. SBOL2 introduced the "functional" classes of `Interaction`, `Participation`, and `Model`. When SBOL2 introduced these classes, they were intentionally kept separated from structural information in a parallel "module" class hierarchy, with the aim of allowing a simpler core "component" hierarchy to focus on the construction of nucleic acid sequences and to be largely shared with SBOL1. As SBOL has been applied to an expanding range of designs, engineering scales, and workflows, however, it has become clear that this dichotomy often tended to create additional complexity by separating elements of a design that would more naturally exist in the same scope. A summary of the changes to the "component" hierarchy is provided in **Table 1**.

For example, consider a simple auto-regulatory device: a transcriptional unit comprising a promoter, ribosome binding site (RBS), coding sequence (CDS), and terminator, where a transcription factor encoded by the CDS represses the activity of the promoter (**Figure 4**). In SBOL1 (or an annotation format such as GenBank or GFF), only the genetic structure of the transcriptional unit can be represented, omitting the regulatory relationship. An SBOL2 representation begins similarly, with a `ComponentDefinition` to represent the transcriptional unit as a whole, with its parts each a `Component` instantiations of the `ComponentDefinition` for the respective constituent promoter, RBS, CDS, and terminator parts, with these functions identified using terms from the *Sequence Ontology* (SO) (Eilbeck et al., 2005). The auto-regulatory interaction must then be expressed separately in a `ModuleDefinition` which, like the `ComponentDefinition`, describes the transcriptional unit, but this time, from a functional perspective. To do this, the transcriptional unit must be instantiated in the `ModuleDefinition` using a

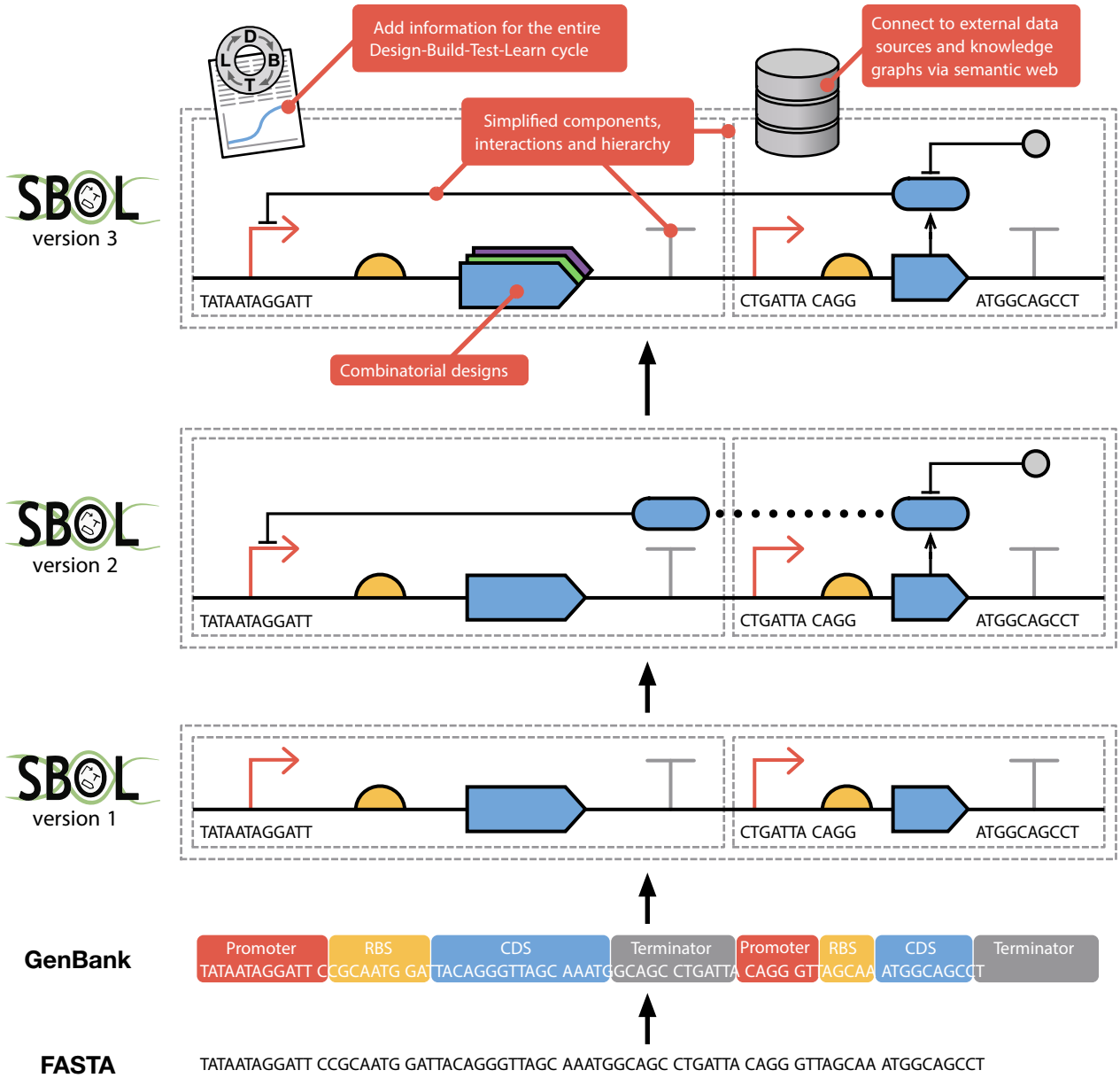


FIGURE 1 | The evolution of SBOL from earlier FASTA and GenBank formats. FASTA was developed to capture pure sequence information. GenBank extends upon this allowing sequences to have annotations, thereby capturing some structural and functional information. SBOL1 adds the ability to use hierarchical composition when describing a design as well as only partially specifying sequences. The complementary SBOL Visual standard (Beal et al., 2019; Madsen et al., 2019a) enables the visual representation of biological design information in an unambiguous way (SBOL version 1, 2, and 3 designs are all shown using SBOL Visual). SBOL2 added the ability to specify modules and functional interactions between parts. Finally, SBOL3 simplifies the SBOL2 data model and greatly improves interoperability with other computational tools through the use of a standardized knowledge graph representation.

`FunctionalComponent`, but its parts are still contained within the `ComponentDefinition` and are not exposed at the level of the `ModuleDefinition`. To document the interaction, therefore, it is also necessary to create promoter and CDS `FunctionalComponent` objects at the level of the `ModuleDefinition` and a `MapsTo` relation for each that identifies the promoter and CDS in the `ModuleDefinition` as being the same promoter and CDS in the `ComponentDefinition`. Finally, an `Interaction` can be created in the `ModuleDefinition` to indicate that the CDS has a regulatory effect on the promoter. While this representation does capture all of the information desired, synthetic biologists do not typically separate their thinking in this manner: the promoter and CDS are being composed as

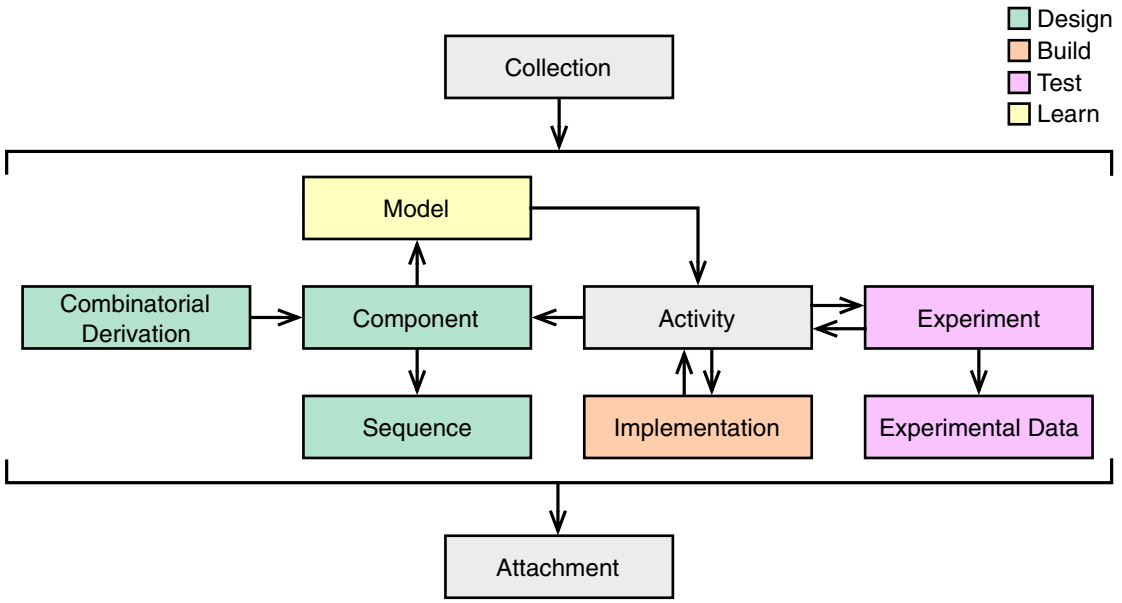


FIGURE 2 | Main top-level classes of SBOL3 and their relationships. The color of each class corresponds to its role in the design (green), build (orange), test (pink), and learn (yellow) stages of the synthetic biology workflow. Additional utility classes are represented by gray boxes.

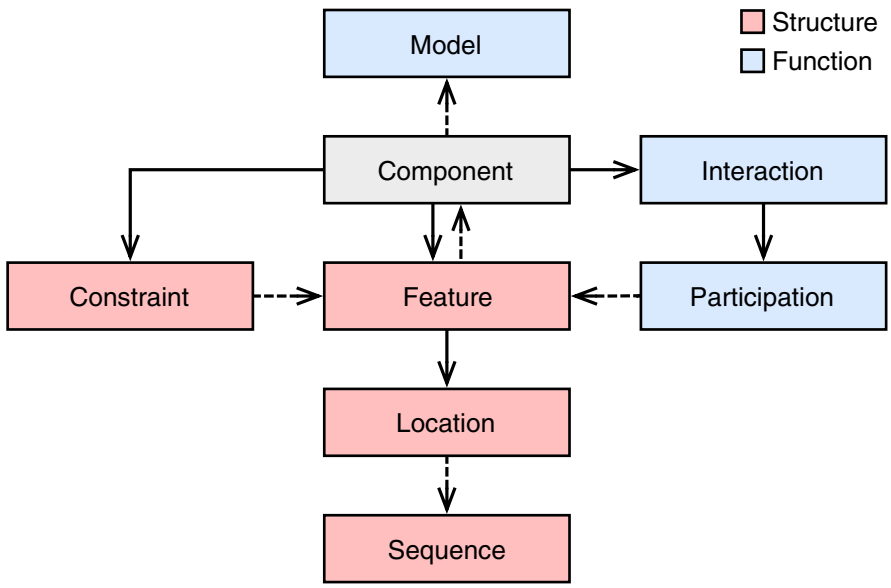


FIGURE 3 | The SBOL3 `Component` object and related objects. Solid arrows indicates ownership and dashed arrows represent a reference to an object of another class. Red and blue boxes represent structural and functional objects, respectively. To represent structural aspects, a `Component` can include `Feature`s, which may refer to `Location`s within a `Sequence`. A `Component` can also include `Constraint`s between these features. To represent functional aspects, a `Component` can include `Interaction`s that can refer to relationships between participating `Feature`s. A `Component` can also have its behavior described using a `Model`.

they are in the sequence structure precisely because of their expected interaction. As a result, rather than deriving advantage from the separation, SBOL tools instead tend to try to hide the distinction from the user, further increasing both complexity and opportunity for error.

In SBOL3, structural and functional aspects are both captured using a single `Component` class (**Figure 3**). Namely, to represent structural aspects, a `Component` can include `Feature`s, some of which may be at some `Location` within a `Sequence`, and which may have `Constraint`s expressing

TABLE 1 | Table of usage scenarios and their corresponding classes in SBOL version 1, 2, and 3.

	SBOL1	SBOL2	SBOL3
DNA part	`DnaComponent`	`ComponentDefinition`	`Component`
Non-DNA part	N/A	`ComponentDefinition`	`Component`
Part uses	`SequenceAnnotation`	`Component`	`SubComponent`
Functional groups	N/A	`ModuleDefinition`	`Component`
Func. group uses	N/A	`Module`	`SubComponent`
Sequence features	`SequenceAnnotation`	`SequenceAnnotation`	`SequenceFeature`
References	N/A	`MapsTo`	`ComponentReference`
External definitions	N/A	N/A	`ExternallyDefined`
Placeholders	N/A	N/A	`LocalSubComponent`

other relations in identity or space. To represent functional relationships a `Component` can include `Interaction`s that can refer to relationships between participating `Feature`s. Finally, a `Component` can refer to an externally defined model using the `Model` class. The SBOL3 representation in **Figure 4** shows how much simpler this unified approach can be, with the functional information added through a single `Interaction` rather than an entire parallel construct and set of identity mappings.

A more complex example illustrating the advantage of this approach is shown in **Figure 5** for the classic genetic toggle switch (Gardner et al., 2000). As with the auto-regulatory device, the SBOL2 representation has compact structural representations of each transcriptional unit, but the functional representation "explodes" these back into a collection of copies and identity mappings for all of the elements that participate in interactions. In SBOL3, on the other hand, the combination of structural and functional information into a single `Component` means that every element of the system appears precisely once and no identity mappings are necessary.

The generalization of `Component` in SBOL3 enables a single, unified hierarchy to capture designs comprising components across multiple scales of a design, from individual molecules to entire cells. For example, the system depicted in **Figure 6** illustrates how the SBOL3 `Component` class can be used to represent a multicellular system where a signaling molecule (AHL) is used for communication between "sender" and "receiver" cells. Moving to these larger scales is also enabled by expanding `Component` type information beyond the Sequence Ontology to additionally use appropriate classes of terms from the Systems Biology Ontology (SBO) (Courtot et al., 2011) and Gene Ontology (GO) (Harris et al., 2004). In this multicellular system, for example, each cell is assigned the role `SBO:0000290` (physical compartment) and type `GO:0005623` (cell), while the subsystems for the sender and receiver are each assigned the role `SBO:0000289` (functional compartment). Constraints are then used to express the spatial structure of the systems, with the sender cells acting to produce AHL molecules initially contained within those cells, the receiver cells responding to the AHL molecules contained within those cells, and the fact that AHL is being shared between the two types of cells is represented by an identity relation between the two instances of the molecule.

Finally, to better support the expanded range of design elements that can be represented, SBOL3 also changes the ontology used for specifying the type of a `Component`. Previous versions of SBOL used the BioPAX (Demir et al., 2010) definitions for molecular species, such as DNA and protein `ComponentDefinition` instances. However, this set of species is restricted, making it difficult to describe designs across different molecular scales. The Systems Biology Ontology (SBO) (Courtot et al., 2011) provides a much richer and more extensible set of terms, already used by SBOL2 in the `Interaction` and `Participation` classes and by SBOL Visual. SBOL3 standardizes the definition of molecular species on SBO in order to have a more expressive and consistent specification of component types. For example, a DNA `Component` can be labeled using the SBO term `SBO:0000251` (Deoxyribonucleic acid), while a complex can be labeled using `SBO:0000254` (Non-covalent complex). A `Component` used to represent primarily functional rather than structural relationships, on the other hand, such as a metabolic synthesis pathway spanning multiple integration sites, uses the `SBO:0000241` (Functional entity) term.

2.2. Features

In SBOL3, the `Feature` class is used to specify elements of interest within a `Component`. SBOL3 introduces several other classes of `Feature` to enable simpler representation of synthetic biology designs.

2.2.1. SubComponents and SequenceFeatures

The original SBOL1 and SBOL2 structural representations focused on the hierarchical composition of parts, such as the inclusion of the *pBAD* promoter in the design of an arabinose sensor. This was accomplished in SBOL2 using a `Component` (now a subclass of `Feature` called `SubComponent` in SBOL3) to refer to a definition of the included part, while its location or locations on the sequence (if known) were expressed using a `SequenceAnnotation`.

However, there are many simpler features (such as a restriction site or -35 region) which are useful to annotate but do not have any meaningful separate hierarchical existence

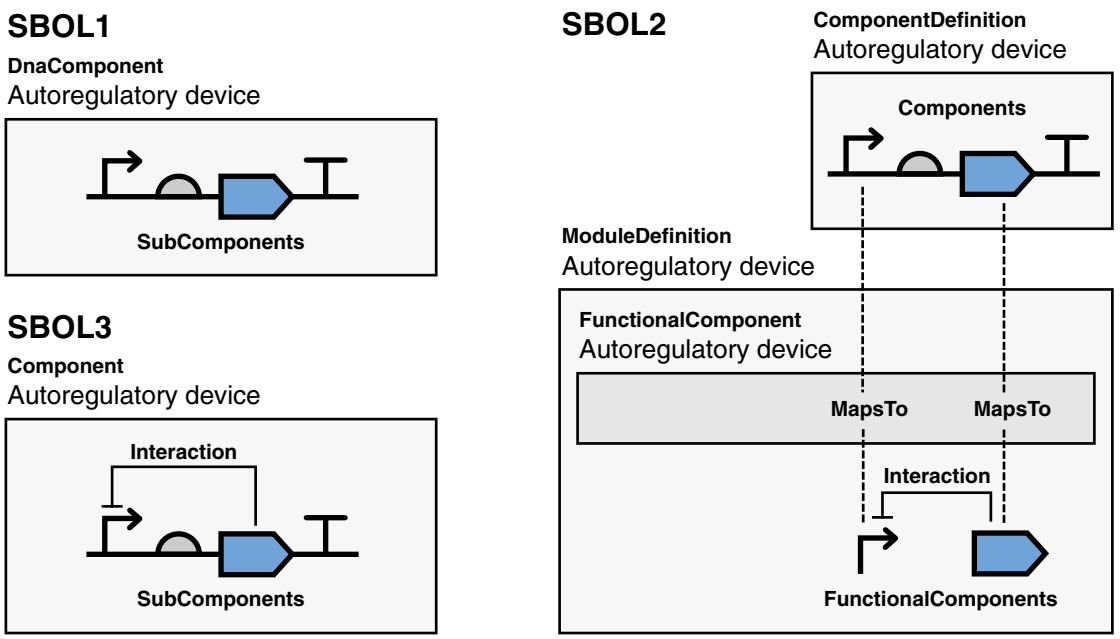


FIGURE 4 | A simple auto-regulatory device represented using SBOL versions 1, 2, and 3. In the SBOL1 example, the structure of the unit is captured, but the regulatory function is not. In the SBOL2 example, the structure is captured using a `ComponentDefinition`, the function is captured in a separate `ModuleDefinition`, and the two objects are connected using `MapsTo` relations. In the SBOL3 example, both the structure and function are captured by a single `Component`, under which the `SubComponent` and `Interaction` objects can co-exist. Diagrams are drawn using SBOL Visual notation (Beal et al., 2019).

within a design. As SBOL2 evolved, such annotations were simplified by allowing a `SequenceAnnotation` to provide feature information about a sequence directly, without the need to link the annotation to a `Component`, but the two classes could not be separated fully without breaking backward compatibility.

In SBOL3, sub-components and feature annotation are now fully refactored into two separate subclasses of `Feature`. The `SubComponent` subclass describes a hierarchical part-subpart relationship, with the option to directly specify its location on a sequence if known and relevant, while the `SequenceFeature` subclass describes a feature that must be associated with a location, but does not indicate a part-subpart relationship.

2.2.2. Local and External Design Elements

SBOL3 also simplifies the handling of two other common cases where defining a full `Component` is not useful. First, similar to `SequenceAnnotation`, a `LocalSubComponent` is used to represent components whose only purpose is to be local placeholders or composites that only really make sense within the context of their parent `Component`, being defined in terms of their relationships with other `Feature`s. For example, a `LocalSubComponent` may be used to specify a variable in a template for a combinatorial library, with the local subcomponent indicating information such as "put a promoter in this location" and "put a barcode in that location." In another example, a `LocalSubComponent` can be used to specify a plasmid assembled from several `SubComponent`s, which then goes on to be transformed into a cell strain.

Another important case is when an established collection of knowledge is better kept outside of SBOL entirely. For example, knowledge about small molecules or proteins is already thoroughly encoded in a standard format in databases such as ChEBI (Degtyarenko et al., 2007) or UniProt (UniProt Consortium, 2007). In SBOL3, an `ExternallyDefined` feature allows such elements to be included in a design by pointing to the canonical non-SBOL definition, while still giving sufficient information to reason about its use within a design via `type` and `role` properties from ontologies such as SBO and GO. In SBOL2, by contrast, such elements were required to be mirrored in "empty" `ComponentDefinition` objects that still essentially just served as a link to the definition while tending to obfuscate the sharing of common design elements.

2.2.3. Simplified References

Finally, SBOL3 also introduces a `ComponentReference` class that allows a `Feature` within a `SubComponent` to be used directly in an `Interaction` or `Constraint` relationship. For example, a `ComponentReference` can be used in an `Interaction` indicating that the TetR protein represses the pTet promoter on a plasmid that is included in a design as a `SubComponent`.

This greatly simplifies such representations relative to SBOL2. In SBOL2, such a reference was constructed by importing a copy of the element as an immediate child of the object where the relationship was expressed and then linking this copy to the original with a `MapsTo` identity relation. The

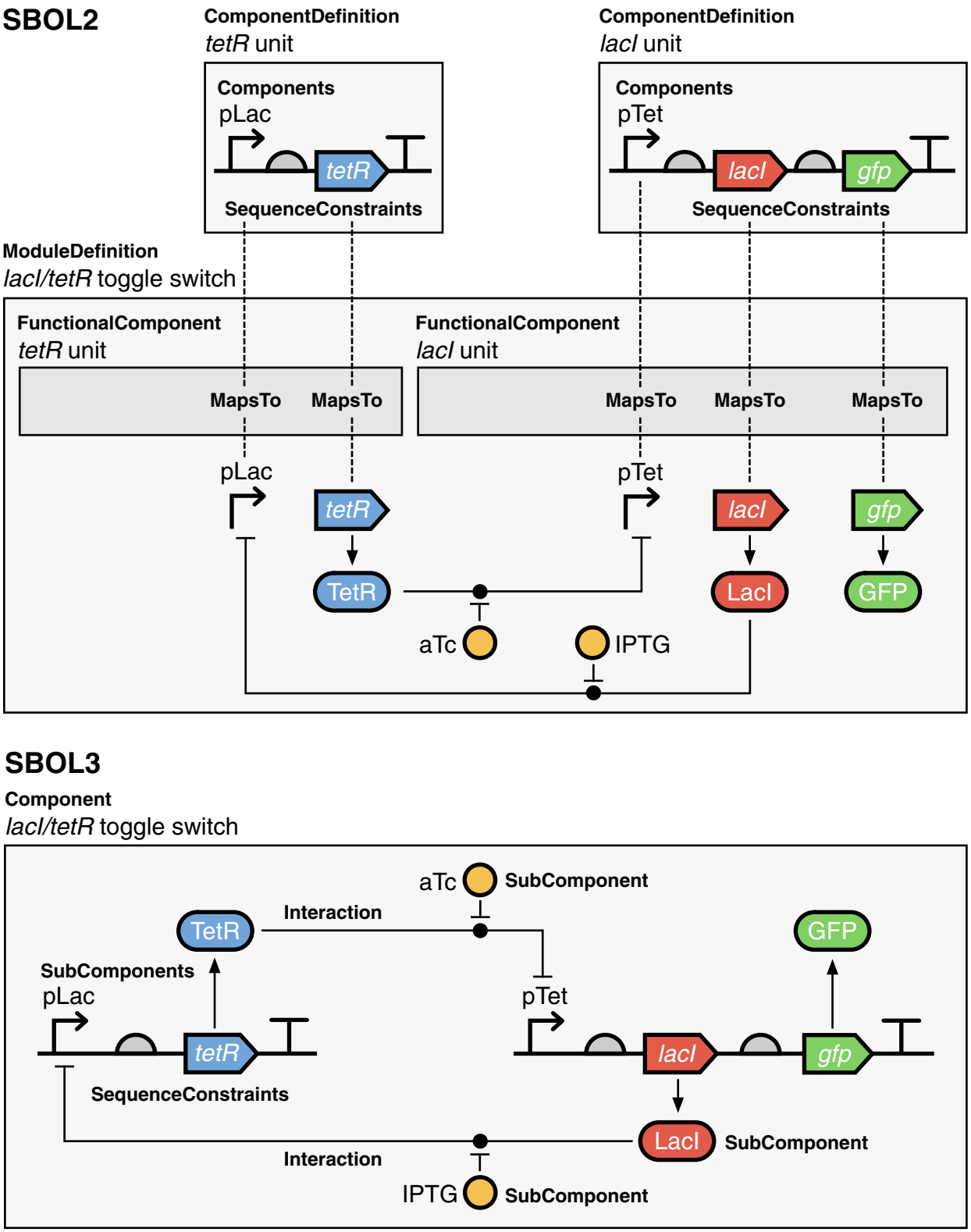


FIGURE 5 | Gardner et al. (2000) toggle switch represented using SBOL versions 2 and 3. In the SBOL2 representation, the structure of the *lacI* and *tetR* units are defined in `ComponentDefinitions` and instantiated in a `ModuleDefinition` as `FunctionalComponents`. Their function-linked sub-structures are then mapped into the `ModuleDefinition` using `MapsTo` to assign corresponding `FunctionalComponents` for the promoter and CDS components, which can then

(Continued)

FIGURE 5 | be used as participants in `Interactions`. In the SBOL3 representation, the creation of a `ModuleDefinition` and `MapsTo` relations as in the SBOL2 example is no longer necessary as sequence information and interactions can co-exist in the same parent `Component` object. Diagrams are drawn using SBOL Visual notation (Beal et al., 2019). A serialized SBOL3 representation of this construct is available on GitHub at https://github.com/SynBioDex/SBOLTestSuite/tree/master/SBOL3/toggle_switch.

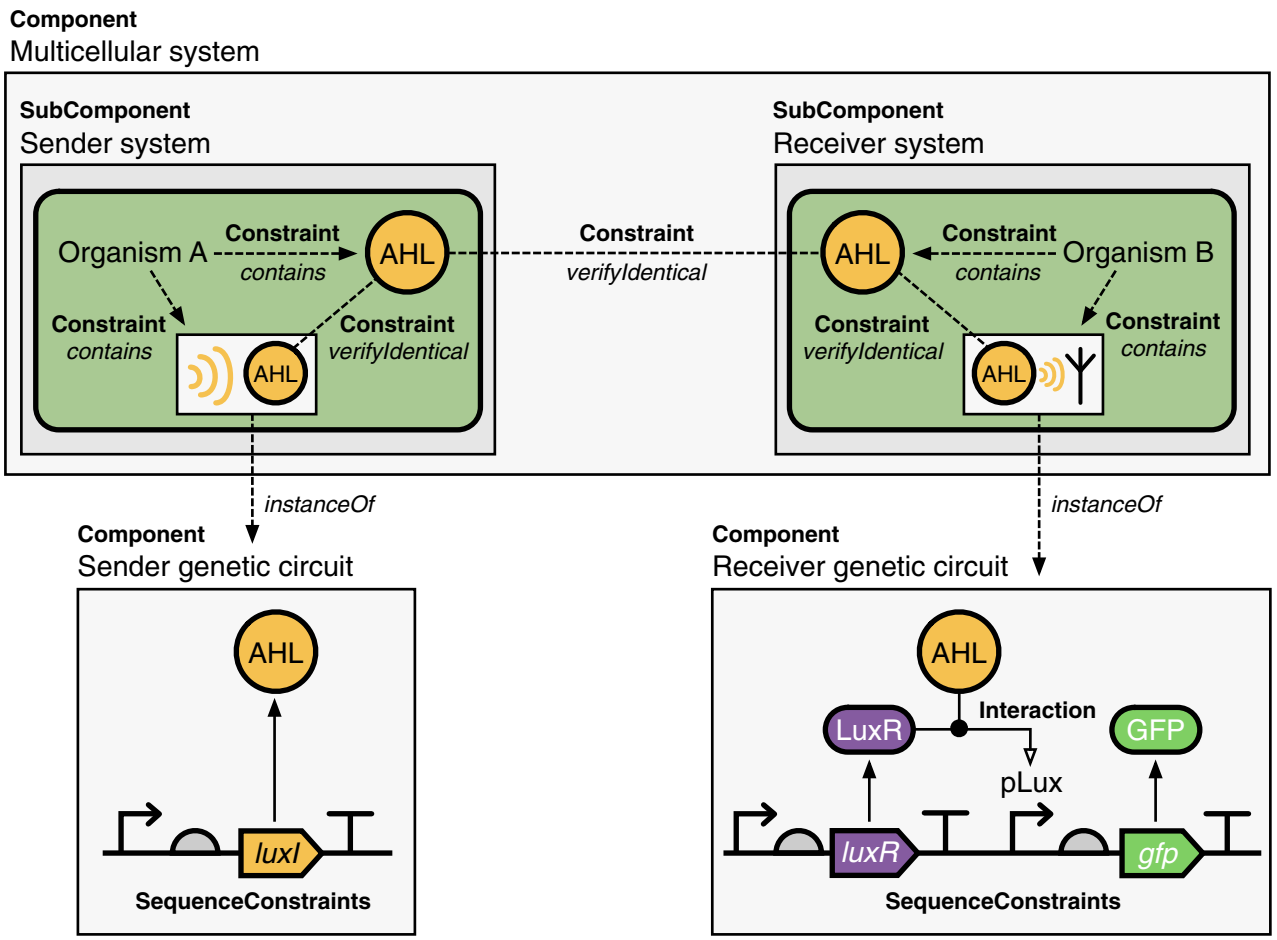


FIGURE 6 | A multicellular communication system represented using SBOL3. Two different organisms implement a sender and receiver system, which uses a small molecule (AHL) as a signal. The sender and receiver systems are represented by `Component`s and use constraints to show that each of these cell types contains AHL (not shown are details of the genetic system and its interactions with the molecule). These sender and receiver systems are `SubComponent`s of the overall multicellular system, which is also represented by a `Component`. The fact that AHL is shared between the two systems is captured using an identity constraint. A serialized SBOL3 representation of this construct is available on GitHub at https://github.com/SynBioDex/SBOLTestSuite/tree/master/SBOL3/multicellular.

`ComponentReference` approach also enables multi-layer references, which were not possible in SBOL2 without also modifying the description of the intermediate layer designs.

2.3. Generalized Constraints

While the `Interaction` class can be used to express functional relationships between biological components, it is also often useful to be able to express information about the non-functional design relationship between components. Such relationships include identity (e.g., replacing a placeholder in a template with a complete definition), relative positions in a sequence (e.g., "pLac precedes *tetR*"), and general spatial relations (e.g., containment of a plasmid in the chassis strain it transforms). The incremental growth of SBOL2 resulted in this information being expressed in a limited manner across a mixture of different classes: identity relationships were expressed using a mix of `MapsTo` and `SequenceConstraint` objects, while spatial relationships were expressed with a mix of `SequenceConstraint` and `Interaction` objects. SBOL3 combines and generalizes these into a unified `Constraint` class, in which two components (a subject and an object) are linked using a restriction to express their relationship.

In SBOL3, identity relationships between components are expressed with the `verifyIdentical`, `differentFrom`,

and `replaces` relationships. The SBOL2 relationships for expressing relative positions in a sequence—`precedes`, `sameOrientationAs`, and `oppositeOrientationAs`—are expanded with additional restrictions that cover the full range of sequential relationships (Allen, 1983): `strictlyPrecedes`, `meets`, `overlaps`, `contains`, `strictlyContains`, `equals`, `finishes`, and `starts`.

Likewise, the set of constraints is further expanded to deal with the spatial relationships of physical objects in general, rather than just the special case of directional linear sequence. In particular, these relations are based on the set of all topological relationships between two spatial regions without holes (Egenhofer and Herring, 1991), including common unions and omitting symmetric relations that can be expressed by swapping subject and object. These new topological restrictions include:

- `isDisjointFrom` – subject and object do not overlap in space. Example: a plasmid is disjoint from a chromosome.
- `strictlyContains` – subject entirely contains object: they do not share a boundary. Example: a cell contains a plasmid.
- `contains` – subject contains object and they might or might not share a boundary. Example: a cell contains a protein that may or may not bind to its membrane.
- `meets` – subject and object are connected at a shared boundary. Example: two strains of adherent cells meet at their membranes.
- `covers` – subject contains object but also shares a boundary. Example: a bacterial cell encloses its transmembrane proteins.
- `overlaps` – subject and object overlap in space, but portions of each are outside of the other. Example: a transmembrane protein overlaps the cell membrane.

Taken all together, these three sets of relationships provide a much simpler and more expressive system for expressing design constraints in SBOL3 than existed in SBOL2.

2.4. Interfaces

In SBOL2, information about the recommended interface for a component/module was dispersed into the "access" field of `ComponentInstance` and the "direction" field of `FunctionalComponent`. This makes the interfaces implicit rather than explicit, scatters the information, and forced premature definition of information about interfaces. As SBOL is now being used to build designs that comprise more complex devices on a larger scale, a clear specification of how components work together is highly important.

In SBOL3, this information is instead collected into an explicit `Interface` object with `input`, `output`, and `non-directional` properties. Each of these properties refers to a set of `Feature` objects in the same `Component` that owns the `Interface`. Specifying any `Interface` is optional, however, so this information only need be added to systems where it makes sense and at an appropriate stage of engineering. For example, a NOR gate from (Gander et al., 2017) could be described as an SBOL3 `Component` with four `SubComponent`s: two gRNA inputs, the DNA component that they regulate (comprising two binding sites, a promoter, and a gRNA coding sequence), and the gRNA output. It would then be assigned an `Interface` with two input relations (to the input gRNA SubComponents) and one output relation (to the output gRNA SubComponent).

2.5. Relationship With RDF and the Semantic Web

All versions of SBOL have used RDF as a serialization format. However, the relationship between SBOL and its underlying Semantic Web representation has previously been unclear. SBOL3 addresses these issues by following Semantic Web related best practices where possible, enabling better integration with existing Semantic Web tools.

2.5.1. Consistent Property Names

SBOL uses many terms from existing ontologies, such as Dublin Core and PROV-O. The SBOL1 and SBOL2 specifications were written in a manner such that those terms were given a new "SBOL alias" that was sometimes, but not always, distinct from the name assigned to them by the ontology. For example, instead of defining the concept of a "title" or "description," the SBOL2 specification used the `dcterms:title` and `dcterms:description` properties from the Dublin Core ontology. However, the `dcterms:title` property is first introduced as the "SBOL alias" of `name`, and then later "mapped" to an ontology term in the serialization section of the specification.

This makes serialized SBOL confusing to read, because the ontologically-defined names used in the serialization do not always match the specification-defined names used by SBOL libraries. For example, SBOL2 renames the prov:wasDerivedFrom property to `wasDerivedFroms` for consistency with other aliases used in the specification. This also meant that integrating terms from other ontologies into SBOL2 required a two-step process of writing their description as SBOL "aliases" and then writing their "serialization."

In SBOL3, the use of external ontologies has been made explicit and consistent throughout the specification. For example, `dcterms:title` has been replaced with an `sbol:name` property, and all of the diagrams in the specification have been updated to display the singular, prefixed form of property names (e.g., `prov:wasDerivedFrom`) rather than an "SBOL-adjusted" version (`wasDerivedFroms`).

2.5.2. Differentiating SBOL Entities (Concepts) and Properties

The SBOL2 data model has several labels that are both used to refer to entities and property names. In SBOL2, they were differentiated by using the uppercase letter when referring to entities and using the lowercase letter when referring to property names. However, not all RDF tools are case-sensitive. Moreover, referring to the data model makes it more difficult to explain in papers. In SBOL3, this ambiguity is removed and the labels are made as unique as possible. Additionally, prefixing and suffixing is applied to property names, e.g., "has..." or "is...Of," as is the Semantic Web convention. For example, the SBOL2 `interaction` property is now `hasInteraction` in SBOL3. Additionally, all entities that are represented as RDF

resources now begin with an uppercase letter, again following RDF convention. For example, the `public` specifier in SBOL2 is now `Public` in SBOL3.

2.5.3. Serialization

Before SBOL3, the standard specified a bespoke file format used for data exchange. This file format required the development of libraries specifically for serializing and parsing SBOL data. In contrast, SBOL3 no longer specifies a particular file format for data exchange. Rather, it specifies how SBOL data structures map to an RDF graph representation. This graph may then be easily serialized to and parsed from a number of file formats, such as XML, Turtle, N-Triples, and JSON, using standard software packages. In addition to simplifying the underlying software implementation, different serialization formats may provide advantages for certain users. For example, Turtle increases human-readability of SBOL documents, and even allows them to be edited manually, while JSON is particularly convenient when developing web applications using JavaScript, and N-Triples is better for minimal difference detection version control systems.

2.6. Namespaces and Identifiers

Finally, one of the important considerations for enabling design data interoperability is the need for consistent and compatible identifiers. As SBOL is built upon RDF, it inherits the World Wide Web concept of a *Uniform Resource Identifier* (URI), a superset of the *Uniform Resource Locator* (URL) standard. Consequently, most SBOL resources, whether in a local SBOL file or in an online repository, have an identifier that resembles a Web address.

In SBOL1, the format of these URIs was left unspecified, meaning there is little consistency in the URIs created by different SBOL1-enabled tools. SBOL2 introduced the concept of "compliant URIs," which comply with a set of optional best practice rules. Broadly, compliant URIs take the form `<URI prefix>/<displayId>/<version>`, where the URI prefix of child objects must be prefixed with the persistent identity of their parent.

While SBOL2 compliant URIs are an improvement over the lack of specification in SBOL1, they also suffer from several practical issues. First, the positioning of the version at the end of the URI is contrary to the established RDF convention of positioning the identifier at the end, meaning existing RDF tooling often displays the version of SBOL2 resources in place of the identifier. Second, URI-suffix versioning is too granular (at object level, when changes are often made across many objects in a design) but also too contagious (changing an object version requires making duplicate copies of everything that points to it as well). Finally, these rules remain optional, meaning there is no guarantee that SBOL2 data has compliant URIs, and it is unclear when implementing tooling how to handle the case of mixed compliant and non-compliant URIs.

SBOL3 addresses these issues by replacing the best practice of compliant URIs with a required SBOL3 URI structure of the form `<URI prefix>/<displayId>`, leaving the handling of versioning and placement (if any) of the version up to the tooling. For example, the version could become part of the prefix (e.g., http://example.com/toggleswitch/1/lacI, part of the displayId (e.g., http://example.com/toggleswitch/lacI_1, or even omitted entirely (e.g., handled instead via git versioning).

Another challenge in SBOL2 was determining which portion of a URI to rewrite when moving it from one namespace to another. This often occurs when an SBOL document is migrated from hosting on one server to a new location on a different server, due to the dual role of a URI as both identifier and Web locator. SBOL3 addresses this by introducing a `Namespace` class that can be used to explicitly encode which portion of a set of URIs should change and which should be retained.

3. DISCUSSION

SBOL supports the representation of abstraction hierarchies across multiple scales of bioengineering, from individual molecules to multi-cellular compositions and complete synthetic genomes (Bartley et al., 2020). The SBOL data model supports a wide variety of important use cases for synthetic biology and bioengineering, including visualization (McLaughlin et al., 2016), sequence design automation (Zhang et al., 2017), sharing of genetic design information (McLaughlin et al., 2018), metabolic engineering (Kuwahara et al., 2017), and generation of dynamical models from sequence representations (Misirli et al., 2018). Additionally, SBOL can be used to capture information about the workflows used to engineer biological systems, supporting reproducibility and automation of these processes.

As described in this paper, the SBOL community has drawn upon several years of experience with the real-world use of SBOL in scientific and industrial settings to produce a specification for SBOL3 that is simultaneously simpler and more expressive. Improvements to the standard in SBOL3 generally fall into one of two categories: simplification of the data model, or closer conformance with Semantic Web best practices. Major simplifications in the data model include the unification of structural and functional compositions into a single component hierarchy; simplification of the description of sub-components and sequence features; and simplifying connections between inputs and outputs across modular interfaces (e.g., transcriptional logic gates).

The other category of improvements in SBOL3 adjust the standard to take better advantage of Semantic Web technologies. By embracing existing developments, this shift will enable more rapid development of SBOL tools and libraries and simplify their maintenance. It will also enable users of SBOL to more easily integrate biological knowledge in the context of their tools through the use of ontologies, which are already widely used in the life sciences to explicitly define biological entities and their relationships. In addition to building upon existing ontologies wherever possible such as the Sequence Ontology (Eilbeck et al., 2005) and the Systems Biology Ontology (Courtot et al., 2011), SBOL itself is now represented as a machine-readable ontology, SBOL-OWL (Misirli et al., 2019). Similar to how ontologies are built upon the RDF layer to provide the meaning of RDF graphs, SBOL-OWL defines data model entities that are used to build SBOL graphs. Formal representation of the data model as an ontology opens up the possibility of using different Semantic

Web tools, such as using existing reasoners to infer information, or validating SBOL data against a schema. Logical axioms are then used to constrain how different SBOL entities can be used together. SBOL-OWL is also embedded into the SBOL Visual Ontology (Misirli et al., 2020), which has been developed as a machine-accessible catalog of glyphs. This integration further facilitates searching for standard SBOL glyphs using ontological terms, and a web service layer enables accessing these glyphs via the Internet.

Overall, these improvement produce a new version of SBOL that provides for a more direct and elegant expression of a broad range of bioengineering information, while at the same time reducing the number of complex classes and rules to a functional minimum, thus providing a significantly improved means of data exchange. These improvements will thus facilitate easier adoption by new users and more rapid development of software tools and datasets that make use of the standard.

3.1. Future Work

The dramatic expansion in scope from the simple DNA components of SBOL1 to the complex systems across multiple scales captured by SBOL3 was driven by the needs of the synthetic biology community, as the field of synthetic biology matured and its applications became both more widespread and more complex. The SEP process by which SBOL3 was developed ensures the standard can continually adapt to the changing requirements of an evolving discipline, while ensuring that proposed changes are ratified by the community. For example, proposals for SBOL 3.0.1 have already been made to improve internationalization by adopting a file encoding and replacing Uniform Resource Identifiers (URIs) with Internationalized Resource Identifiers (IRIs).

While the nature of future requirements can only be speculated, there are many aspects of the synthetic biology lifecycle which remain largely unspecified by SBOL. For example, while SBOL recommends the use of the `prov:Plan` class, it does not yet recommend any domain-specific properties for its annotation. Equally, while the concept of an experiment can be captured in SBOL, it does not yet standardize metadata about the experiment or experimental data. Future revisions of the SBOL standard will therefore undoubtedly concern not only its expressiveness in describing design elements, but also its ability to capture and formalize the synthetic biology lifecycle as a whole.

4. METHODS

Since its inception, the SBOL Standard has been developed as a community effort by the SBOL Development Group, which is open to any interested person. However, the development process was largely informal until the SBOL Enhancement Proposal (SEP) mechanism was introduced in 2015 (Grünberg and Bartley, 2015), shortly after the finalization of the SBOL2 specification. Development of SBOL3 has been driven by this formal process of documenting user experiences, developing proposals, and constructively debating the merits of these proposals.

Under this process, any SBOL user can propose a change by drafting a document with a specific format (an SEP), which is then discussed by the community on the mailing list and in GitHub issues associated with the SEP. Once the current elected editors of the standard judge that an SEP has been discussed sufficiently and an approximate consensus achieved, a voting form is posted, and any member of the SBOL Developers Group can vote for or against it. The SEP is immediately accepted if at least a two-thirds majority of votes cast are in favor. Otherwise, there is a further period of discussion, during which the SEP can be modified or withdrawn by its original author(s), followed by a second vote in which only a simple majority is required for acceptance.

Since the publication of SBOL2 in 2015, 46 SEPs have been opened, as community experience in deployment of SBOL revealed some of the practical challenges and opportunities for enhancement. Of these SEPs, twelve were implemented as incremental updates to SBOL2, resulting in significant milestones in SBOL version 2.1.0 (Beal et al., 2016), which introduced feature annotation and the encoding of provenance information to trace the history of designs; SBOL version 2.2.0 (Cox et al., 2018), which introduced support for combinatorial designs; and SBOL version 2.3.0 (Madsen et al., 2019b), which introduced extensions to support measurements, parameters, and the organization and attachment of experimental data.

Other SEPs were deemed too major to be integrated into a 2.x release of SBOL, since they would create backwards compatibility problems. Therefore, they were scheduled for SBOL version 3. After a series of community votes, a working group met to assemble the SBOL3 specification at the HARMONY 2020 Workshop at EMBL-EBI in Cambridge, UK. The resolution of conflicts between these SEPs resulted in a final SEP summarizing all changes in the SBOL3 data model. After voted acceptance of this SEP by the community, the SBOL3 specification was finalized.

Though there are not yet any complete software implementations of SBOL3, the SBOL community has established an SBOL3 implementation working group comprising many of the developers of libraries for previous SBOL versions and other interested parties. The first software libraries are expected to be released within the coming months for Java, Python, and JavaScript. Preliminary support for SBOL3 has been implemented in ShortBOL (Crowther et al., 2020), a tool for composing SBOL using a shorthand syntax.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. The specifications described in this article are freely available from https://sbolstandard.org/data-model-specification/.

AUTHOR CONTRIBUTIONS

JM, JB, GM, RG, and CM authored the SEP proposals from which the SBOL3 standard was developed. JM, JB, GM, RG, BB, JS-B, TG, PV, PF, EO, AW, and CM developed the SBOL3 specification

and authored this manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

JM was supported by FUJIFILM DioSynth Biotechnologies. JB was supported in part by the NSF Expeditions in Computing Program Award #1522074 as part of the Living Computing Project. JB and BB were supported in part by the Air Force Research Laboratory (AFRL) and DARPA under contract FA875017CO184. JS-B was supported by the EPSRC & BBSRC Centre for Doctoral Training in Synthetic Biology (grant EP/L016494/1) and by DSTL. TG was supported by BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre (grant BB/L01386X/1) and a Royal Society University Research Fellowship (grant UF160357). The work of EO was part of the DOE Joint Genome Institute (https://jgi.doe.gov) supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U.S. Department of Energy. AW was supported by The Engineering and Physical Sciences Research Council grants EP/J02175X/1, EP/R003629/1, EP/N031962/1 (JM and AW), and EP/R019002/1. CM and PF were supported by the National Science Foundation under grants CCF-1748200 and 1939892 and DARPA grant FA8750-17-C-0229. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

ACKNOWLEDGMENTS

In addition to the listed authors, the development of SBOL3 has benefited greatly from discussions with many stakeholders throughout the user and developer community, including the SBOL developers mailing list, users and developers of the SBOL libraries and SBOL-enabled software tools, and the SBOL Industrial Consortium. Valuable support and guidance was also provided by members of the SBOL Editors and SBOL Steering Committee.

REFERENCES

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 832–843.

Appleton, E., Tao, J., Haddock, T., and Densmore, D. (2014). Interactive assembly algorithms for molecular cloning. *Nat. Methods* 11:657. doi: 10.1038/nmeth.2939

Baig, H., Fontanarrosa, P., Kulkarni, V., McLaughlin, J. A., Vaidyanathan, P., Bartley, B., et al. (2020). Synthetic Biology Open Language (SBOL) version 3.0. 0. *J. Integr. Bioinform*. 1. doi: 10.1515/jib-2020-0017

Baig, H., and Madsen, J. (2017). "A top-down approach to genetic circuit synthesis and optimized technology mapping," in *9th International Workshop on Bio-Design Automation* (Pittsburgh, PA), 1–2.

Bartley, B., Beal, J., Clancy, K., Misirli, G., Roehner, N., Oberortner, E., et al. (2015). Synthetic Biology Open Language (SBOL) version 2.0.0. *J. Integr. Bioinform*. 12, 902–991. doi: 10.2390/biecoll-jib-2015-272

Bartley, B. A., Beal, J., Karr, J. R., and Strychalski, E. A. (2020). Organizing genome engineering for the gigabase scale. *Nat. Commun*. 11, 1–9. doi: 10.1038/s41467-020-14314-z

Bartoli, V., Dixon, D. O. R., and Gorochowski, T. E. (2018). *Automated Visualization of Genetic Designs Using DNAplotlib*. New York, NY: Springer. doi: 10.1007/978-1-4939-7795-6_22

Beal, J., Cox, R. S., Grünberg, R., McLaughlin, J., Nguyen, T., Bartley, B., et al. (2016). Synthetic Biology Open Language (SBOL) version 2.1.0. *J. Integr. Bioinform*. 13, 30–132. doi: 10.1515/jib-2016-291

Beal, J., Nguyen, T., Gorochowski, T. E., Goñi-Moreno, A., Scott-Brown, J., McLaughlin, J. A., et al. (2019). Communicating structure and function in synthetic biology diagrams. *ACS Synthet. Biol*. 8, 1818–1825. doi: 10.1021/acssynbio.9b00139

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., et al. (2013). Genbank. *Nucleic Acids Res*. 41, D36–D42. doi: 10.1093/nar/gks1195

Bhatia, S., and Densmore, D. (2013). Pigeon: a design visualizer for synthetic biology. *ACS Synthet. Biol*. 2, 348–350. doi: 10.1021/sb400024s

Bilitchenko, L., Liu, A., Cheung, S., Weeding, E., Xia, B., Leguia, M., et al. (2011). Eugene–a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS ONE* 6:e18882. doi: 10.1371/journal.pone.0018882

Chen, J., Densmore, D., Ham, T. S., Keasling, J. D., and Hillson, N. J. (2012). Deviceeditor visual biological cad canvas. *J. Biol. Eng*. 6:1. doi: 10.1186/1754-1611-6-1

Cooling, M. T., Rouilly, V., Misirli, G., Lawson, J., Yu, T., Hallinan, J., et al. (2010). Standard virtual biological parts: a repository of modular modeling components for synthetic biology. *Bioinformatics* 26, 925–931. doi: 10.1093/bioinformatics/btq063

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., et al. (2011). Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol*. 7:543. doi: 10.1038/msb.2011.77

Cox, R. S., Madsen, C., McLaughlin, J. A., Nguyen, T., Roehner, N., Bartley, B., et al. (2018). Synthetic Biology Open Language (SBOL) version 2.2.0. *J. Integr. Bioinform*. 15. doi: 10.1515/jib-2018-0001

Crowther, M., Grozinger, L., Pocock, M., Taylor, C. P., McLaughlin, J. A., Mısırlı, G., et al. (2020). ShortBOL: a language for scripting designs for engineered biological systems using Synthetic Biology Open Language (SBOL). *ACS Synthet. Biol*. 9, 962–966. doi: 10.1021/acssynbio.9b00470

Czar, M. J., Cai, Y., and Peccoud, J. (2009). Writing DNA with genoCAD. *Nucleic Acids Res.* 37(suppl_2), W40–W47. doi: 10.1093/nar/gkp361

Dalchau, N., Grant, P. K., Vaidyanathan, P., Spaccasassi, C., Gravill, C., and Phillips, A. (2019). Scalable dynamic characterization of synthetic gene circuits. *bioRxiv [Preprint]* 635672. doi: 10.1101/635672

Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., et al. (2007). Chebi: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res.* 36(suppl_1), D344–D350. doi: 10.1093/nar/gkm791

Demir, E., Cary, M. P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., et al. (2010). The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.* 28:935. doi: 10.1038/nbt.1666

Der, B. S., Glassey, E., Bartley, B. A., Enghuus, C., Goodman, D. B., Gordon, D. B., et al. (2017). DNAplotlib: programmable visualization of genetic designs and associated data. *ACS Synthet. Biol.* 6, 1115–1119. doi: 10.1021/acssynbio.6b00252

Egenhofer, M. J., and Herring, J. (1991). *Categorizing Binary Topological Relations between Regions, Lines, and Points in Geographic Databases*. Technical report, University of Maine. doi: 10.1007/3-540-54414-3_36

Eilbeck, K., Lewis, S. E., Mungall, C. J., Yandell, M., Stein, L., Durbin, R., et al. (2005). The sequence ontology: a tool for the unification of genome annotations. *Genome Biol.* 6:R44. doi: 10.1186/gb-2005-6-5-r44

Endy, D. (2005). Foundations for engineering biology. *Nature* 438:449. doi: 10.1038/nature04342

Fu, G., Batchelor, C., Dumontier, M., Hastings, J., Willighagen, E., and Bolton, E. (2015). Pubchemrdf: towards the semantic annotation of pubchem compound and substance databases. *J. Cheminform.* 7:34. doi: 10.1186/s13321-015-0084-4

Galdzicki, M., Clancy, K. P., Oberortner, E., Pocock, M., Quinn, J. Y., Rodriguez, C. A., et al. (2014). The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat. Biotechnol.* 32, 545–550. doi: 10.1038/nbt.2891

Galdzicki, M., Wilson, M. L., Rodriguez, C. A., Adam, L., Adler, A., Anderson, J. C., et al. (2011). *BBF RFC 84: Synthetic Biology Open Language (SBOL) Version 1.0.0*. Technical report, BioBricks Foundation.

Gander, M. W., Vrana, J. D., Voje, W. E., Carothers, J. M., and Klavins, E. (2017). Digital logic circuits in yeast with crispr-dcas9 nor gates. *Nat. Commun.* 8, 1–11. doi: 10.1038/ncomms15459

Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403, 339–342. doi: 10.1038/35002131

Grünberg, R., and Bartley, B. (2015). *SEP 001: SBOL Enhancement Proposals*. Available online at: https://github.com/SynBioDex/SEPs/issues/1.

Hallinan, J., Gilfellon, O., Misirli, G., and Wipat, A. (2014). "Tuning receiver characteristics in bacterial quorum communication: an evolutionary approach using standard virtual biological parts," in *2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology* (Honolulu, HI: IEEE), 1–8. doi: 10.1109/CIBCB.2014.6845520

Ham, T. S., Dmytriv, Z., Plahar, H., Chen, J., Hillson, N. J., and Keasling, J. D. (2012). Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. *Nucleic Acids Res.* 40, e141–e141. doi: 10.1093/nar/gks531

Harris, M., Deegan, J., Ireland, A., Lomax, J., Ashburner, M., Foulger, R, et al. (2004). The gene ontology (GO) database and informatics resource. *Nucleic Acids Res.* 32(suppl_1), D258–D261. doi: 10.1093/nar/gkh036

Hillson, N. J., Plahar, H. A., Beal, J., and Prithviraj, R. (2016). Improving synthetic biology communication: recommended practices for visual depiction and digital submission of genetic designs. *ACS Synthet. Biol.* 5, 449–451. doi: 10.1021/acssynbio.6b00146

Hillson, N. J., Rosengarten, R. D., and Keasling, J. D. (2012). J5 DNA assembly design automation software. *ACS Synthet. Biol.* 1, 14–21. doi: 10.1021/sb2000116

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531. doi: 10.1093/bioinformatics/btg015

Kuwahara, H., Cui, X., Umarov, R., Grünberg, R., Myers, C. J., and Gao, X. (2017). SBOLme: a repository of SBOL parts for metabolic engineering. *ACS Synthet. Biol.* 6, 732–736. doi: 10.1021/acssynbio.6b00278

Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., et al. (2013). *Prov-O: The Prov Ontology. W3C Recommendation, 30*.

Madsen, C., Moreno, A. G., Palchick, Z., P, U., Roehner, N., Bartley, B., Bhatia, S., et al. (2019a). Synthetic Biology Open Language Visual (SBOL Visual) Version 2.1. *J. Integr. Bioinform.* 16:20180101. doi: 10.1515/jib-2018-0101

Madsen, C., Moreno, A. G., Umesh, P., Palchick, Z., Roehner, N., Atallah, C., et al. (2019b). Synthetic Biology Open Language (SBOL) version 2.3. *J. Integr. Bioinform.* 16. doi: 10.1515/jib-2019-0025

McLaughlin, J. A., Myers, C. J., Zundel, Z., Mısırlı, G., Zhang, M., Ofiteru, I. D., et al. (2018). Synbiohub: a standards-enabled design repository for synthetic biology. *ACS Synthet. Biol.* 7, 682–688. doi: 10.1021/acssynbio.7b00403

McLaughlin, J. A., Pocock, M., Mısırlı, G., Madsen, C., and Wipat, A. (2016). VisBOL: web-based tools for synthetic biology design visualization. *ACS Synthet. Biol.* 5, 874–876. doi: 10.1021/acssynbio.5b00244

Misirli, G., Beal, J., Gorochowski, T. E., Stan, G.-B., Wipat, A., and Myers, C. J. (2020). SBOL visual 2 ontology. *ACS Synthet. Biol.* 9, 972–977. doi: 10.1021/acssynbio.0c00046

Misirli, G., Hallinan, J., and Wipat, A. (2014). Composable modular models for synthetic biology. *ACM J. Emerg. Technol. Comput. Syst.* 11, 1–19. doi: 10.1145/2631921

Misirli, G., Hallinan, J. S., Yu, T., Lawson, J. R., Wimalaratne, S. M., Cooling, M. T., et al. (2011). Model annotation for synthetic biology: automating model to nucleotide sequence conversion. *Bioinformatics* 27, 973–979. doi: 10.1093/bioinformatics/btr048

Misirli, G., Nguyen, T., McLaughlin, J. A., Vaidyanathan, P., Jones, T. S., Densmore, D., et al. (2018). A computational workflow for the automated generation of models of genetic designs. *ACS Synthet. Biol.* 8, 1548–1559. doi: 10.1021/acssynbio.7b00459

Misirli, G., Taylor, R., Goni-Moreno, A., McLaughlin, J. A., Myers, C., Gennari, J. H., et al. (2019). SBOL-OWL: An ontological approach for formal and semantic representation of synthetic biology information. *ACS Synthet. Biol.* 8, 1498–1514. doi: 10.1021/acssynbio.8b00532

Myers, C. J., Barker, N., Jones, K., Kuwahara, H., Madsen, C., and Nguyen, N.-P. D. (2009). ibiosim: a tool for the analysis and design of genetic circuits. *Bioinformatics* 25, 2848–2849. doi: 10.1093/bioinformatics/btp457

Nielsen, A. A., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., et al. (2016). Genetic circuit design automation. *Science* 352:aac7341. doi: 10.1126/science.aac7341

Oberortner, E., Bhatia, S., Lindgren, E., and Densmore, D. (2014). A rule-based design specification language for synthetic biology. *ACM J. Emerg. Technol. Comput. Syst.* 11, 1–19. doi: 10.1145/2641571

Oberortner, E., Cheng, J.-F., Hillson, N. J., and Deutsch, S. (2017). Streamlining the design-to-build transition with build-optimization software tools. *ACS Synthet. Biol.* 6, 485–496. doi: 10.1021/acssynbio.6b00200

Oberortner, E., and Densmore, D. (2015). Web-based software tool for constraint-based design specification of synthetic biological systems. *ACS Synthet. Biol.* 4, 757–760. doi: 10.1021/sb500352b

Pearson, W. R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.* 183, 63–98. doi: 10.1016/0076-6879(90)83007-V

Pedersen, M., and Phillips, A. (2009). Towards programming languages for genetic engineering of living cells. *J. R. Soc. Interface* 6(suppl_4), S437–S450. doi: 10.1098/rsif.2008.0516.focus

Quinn, J. Y., Cox, R. S. III, Adler, A., Beal, J., Bhatia, S., Cai, Y., et al. (2015). SBOL visual: a graphical language for genetic designs. *PLoS Biol.* 13:e1002310. doi: 10.1371/journal.pbio.1002310

Redaschi, N., and UniProt Consortium (2009). UniProt in RDF: tackling data integration and distributed annotation with the semantic web. *Nat. Prec.* doi: 10.1038/npre.2009.3193.1

Roehner, N., Beal, J., Clancy, K., Bartley, B., Misirli, G., Grunberg, R., et al. (2016). Sharing structure and function in biological design with SBOL 2.0. *ACS Synthet. Biol.* 5, 498–506. doi: 10.1021/acssynbio.5b00215

Stein, L. (2013). *Generic Feature Format Version 3 (GFF3)*. Available online at: https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md

UniProt Consortium (2007). The universal protein resource (uniprot). *Nucleic Acids Res.* 36(suppl_1), D190–D195. doi: 10.1093/nar/gkm895

Vaidyanathan, P., Der, B. S., Bhatia, S., Roehner, N., Silva, R., Voigt, C. A., et al. (2015). A framework for genetic logic synthesis. *Proc. IEEE* 103, 2196–2207. doi: 10.1109/JPROC.2015.2443832

W3C (2014). *RDF 1.1 Concepts and Abstract Syntax*. Available online at: https://www.w3.org/TR/rdf11-concepts

Watanabe, L., Nguyen, T., Zhang, M., Zundel, Z., Zhang, Z., Madsen, C., et al. (2018). ibiosim 3: a tool for model-based genetic circuit design. *ACS Synthet. Biol.* 8, 1560–1563. doi: 10.1021/acssynbio.8b00078

Zhang, M., McLaughlin, J. A., Wipat, A., and Myers, C. J. (2017). SBOLDesigner 2: an intuitive tool for structural genetic design. *ACS Synthet. Biol.* 6, 1150–1160. doi: 10.1021/acssynbio.6b00275

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor declared a past co-authorship with one of the authors TG.

frontiers in Bioengineering and Biotechnology

METHODS
published: 08 October 2020
doi: 10.3389/fbioe.2020.558247

Prediction of Protein Mutational Free Energy: Benchmark and Sampling Improvements Increase Classification Accuracy

*Brandon Frenz[1], Steven M. Lewis[1], Indigo King[1], Frank DiMaio[2], Hahnbeom Park[2] and Yifan Song[1]**

[1] Cyrus Biotechnology, Seattle, WA, United States, [2] Department of Biochemistry, University of Washington, Seattle, WA, United States

Software to predict the change in protein stability upon point mutation is a valuable tool for a number of biotechnological and scientific problems. To facilitate the development of such software and provide easy access to the available experimental data, the ProTherm database was created. Biases in the methods and types of information collected has led to disparity in the types of mutations for which experimental data is available. For example, mutations to alanine are hugely overrepresented whereas those involving charged residues, especially from one charged residue to another, are underrepresented. ProTherm subsets created as benchmark sets that do not account for this often underrepresent tense certain mutational types. This issue introduces systematic biases into previously published protocols' ability to accurately predict the change in folding energy on these classes of mutations. To resolve this issue, we have generated a new benchmark set with these problems corrected. We have then used the benchmark set to test a number of improvements to the point mutation energetics tools in the Rosetta software suite.

Keywords: mutation, protein, mutation free energy, protein design and engineering, thermodynamics

OPEN ACCESS

Edited by:
Fabio Parmeggiani, University of Bristol, United Kingdom

Reviewed by:
Pietro Sormanni, University of Cambridge, United Kingdom
James T. MacDonald, Imperial College London, United Kingdom

***Correspondence:**
Yifan Song
yifan@cyrusbio.com

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *05 May 2020*
Accepted: *16 September 2020*
Published: *08 October 2020*

Citation:
Frenz B, Lewis SM, King I, DiMaio F, Park H and Song Y (2020) Prediction of Protein Mutational Free Energy: Benchmark and Sampling Improvements Increase Classification Accuracy. Front. Bioeng. Biotechnol. 8:558247. doi: 10.3389/fbioe.2020.558247

INTRODUCTION

The ability to accurately predict the stability of a protein upon mutation is important for numerous problems in protein engineering and medicine including stabilization and activity optimization of biologic drugs. To perform this task a number of strategies and force fields have been developed, including those that perform exclusively on sequence (Casadio et al., 1995; Capriotti et al., 2005; Kumar et al., 2009) as well as those that involve sophisticated physical force fields both knowledge based (Sippl, 1995; Gilis and Rooman, 1996; Potapov et al., 2009), physical models (Pitera and Kollman, 2000; Pokala and Handel, 2005; Benedix et al., 2009), and hybrids (Pitera and Kollman, 2000; Guerois et al., 2002; Kellogg et al., 2011; Jia et al., 2015; Park et al., 2016; Quan et al., 2016).

To facilitate the development of these methodologies and provide easy access to the available experimental information the ProTherm database (Uedaira et al., 2002) was developed. This database collects thermodynamic information on a large number of protein mutations and makes it available in an easy to access format. At the time of this writing it contains 26,045 entries.

Due to its ease of access the ProTherm has served as the starting point for a number of benchmark sets used to validate different stability prediction software packages, including those in the Rosetta software suite. However significant biases exist in the representation of different types and classes of mutations in the ProTherm, as it is derived from the existing literature across many types of proteins and mutations. The most obvious example of this is the large number of entries involving a mutation from a native residue to alanine as making this type of mutation is a common technique used to find residues important for protein function. Therefore a large number of the benchmark sets derived from the ProTherm, which did not account for this bias, have significantly under or overrepresented these classes of mutations. These findings suggest previous reports on the accuracy of stability prediction software does not accurately reflect these tools' ability to predict stability changes across all classes of mutations.

To address this issue we have generated a novel benchmark subset which accounts for this bias in the ProTherm database (**Supplementary Table 1**). We then used this benchmark set to validate and improve upon an existing free energy of mutation tool within the Rosetta software suite, "Cartesian $\Delta\Delta G$," first described in Park et al. (2016).

RESULTS

In order to benchmark our Rosetta-based stability prediction tools we classified the possible mutations into 17 individual categories as well as reported results on four aggregate categories. We analyzed five previously published benchmark sets to determine their coverage across the different classes of mutations and found them inadequate in a number of categories, especially involving charged residues (**Figures 1A–E**). For example, the number of data points for mutational types ranged from 0 to 24 for negative to positive, 0 to 50 for positive to negative, 3 to 28 for hydrophobic to negative, and 3 to 44 for hydrophobic to positive entries across the benchmark sets tested. Mutations to and/or from hydrophobic residues dominated the benchmark sets ranging from 75 to 92% of the total entries.

To compare the composition of these benchmark sets to that of the database we examined the curated ProTherm (ProTherm*) provided by Ó Conchúir et al. (2015)[1] which is a selection of entries containing only mutations which occur on a single chain and provide experimental $\Delta\Delta G$ values (**Supplementary Table 2**). We find that significant biases still exist here, with several categories having fewer than 50 unique mutations. These include: positive to negative, 42; hydrophobic to negative, 43; and non-charged polar to positive, 47. Mutations involving hydrophobic residues are still overrepresented, with 62.2% of all mutations in the database being mutations to hydrophobic residues, compared to the expected 39.8% if mutations from the starting structures were chosen randomly (**Figures 1F–G**).

We also analyzed the benchmark sets with respect to the number of buried vs. exposed residues in the data sets. No large biases were observed. All benchmark sets were within 6% of what would be expected if mutations were random (data not shown).

To sample more broadly across all types of mutations and remove sources of bias in our algorithm development we created a new benchmark set of single mutations that are more balanced across mutational types and avoid other biases. To generate this set we performed the following operations:

(1) Removed any entries from the curated ProTherm* that occur on the interface of a protein complex or interact with ligands—the energetics of these mutations would include intra-protein and inter-molecular interactions that would alter the desired intra-protein energetics of a free energy calculation.
(2) We removed entries of identical mutation on similar-sequence (>60%) backbones. For mutations occurring at the same position in similar sequences, if the mutation is identical (e.g., L → I) and the sequence identity >60%, then that mutation is included only once in the database; if the mutation is not identical (L → I in one protein and L → Q in another) then the mutation is included.
(3) We populated each mutation category, excluding small to large, large to small, buried, and surface, with 50 entries except for the cases where insufficient experimental data points exist. Statistics on the excluded categories were derived from data points that were already present in the other categories.
(4) When multiple experimental values (including identical mutations as identified in point 2 above) were available we chose the $\Delta\Delta G$ value taken at the pH closest to 7.

The resulting benchmark set contains 767 entries across a range of different types and classes of mutations (**Figure 2**). This constitutes a reduction from the 2,971 total entries in the curated ProTherm*, with mutations to hydrophobics being the most frequently being eliminated. This reduction does eliminate potentially useful data, and introduces a slight bias toward solvent exposed residues: 66% of the mutations being on residues with greater than 20% solvent exposed surface area compared to 54% if chosen randomly. This change is useful to reduce bias toward favoring hydrophobic mutations and has been controlled for by checking our algorithm's performance when residues are classified by burial.

We tested Protocol 3 described in Kellogg et al. (2011) on this benchmark set (**Table 1**). To assess method quality, we analyzed prediction power by a number of different methods including Pearson's R, Predictive Index (Pearlman and Charifson, 2001), and Matthews Correlation Coefficient (MCC) (Matthews, 1975). We also analyzed classification errors instead of correlation. A mutation is classified as stabilizing if the change in free energy is $\leq$-1 kcal/mol, it is classified as destabilizing if the change is $\geq$1 kcal/mol, and neutral if it falls between these values. Each mutation is assigned a value of 0 for destabilizing, 1 for neutral, and 2 for stabilizing. We then scored each entry by taking the absolute value of the difference between the value for the experiment and the prediction. A value of 0 indicates the prediction was correct, 1 indicates the prediction

[1]https://guybrush.ucsf.edu/benchmarks/benchmarks/DDG

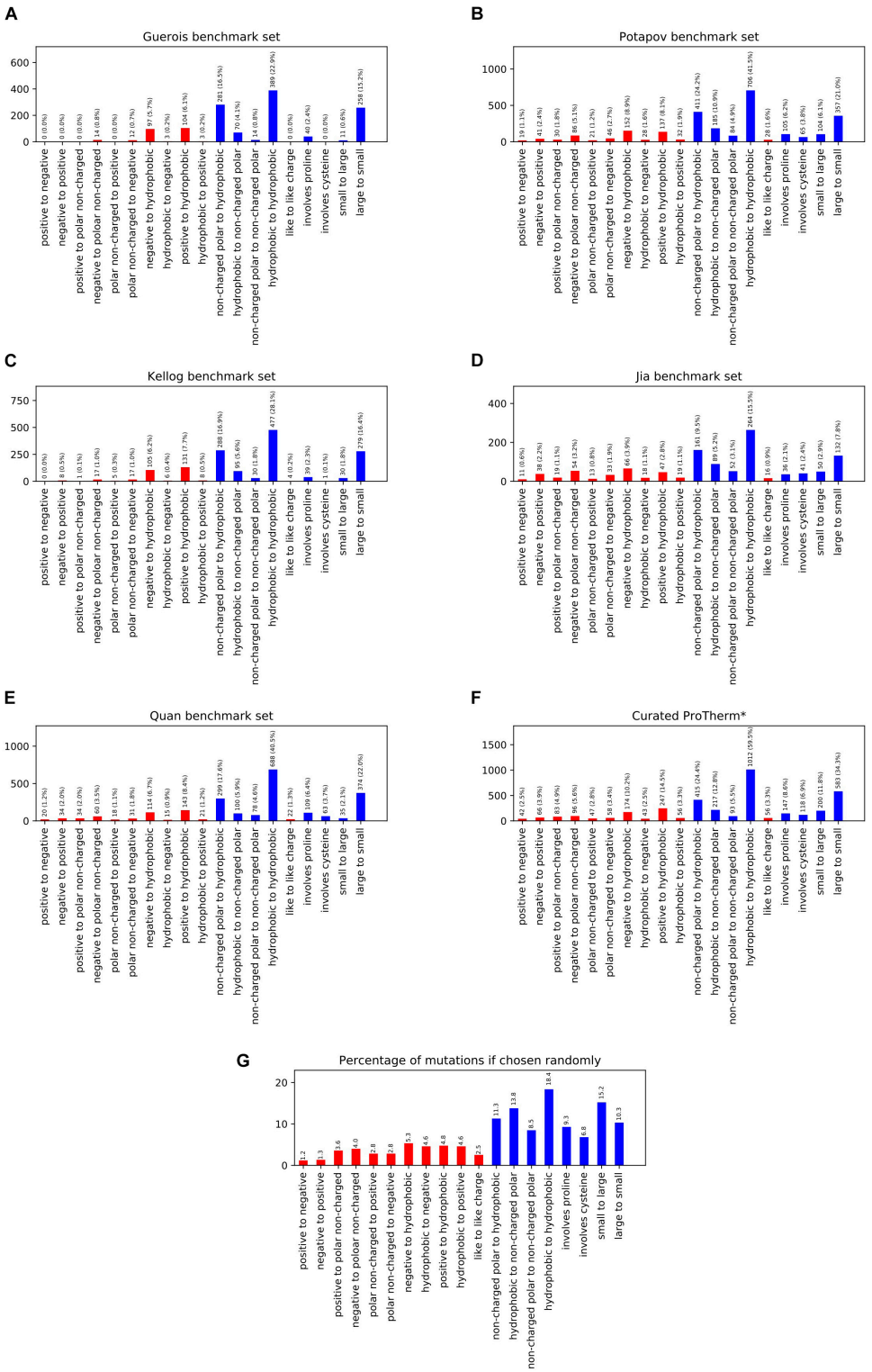


FIGURE 1 | This figure shows the population of different mutation classes used to benchmark a number of methods ability to predict the change in free energy upon mutation. The citations for these benchmark sets are as follows: **(A)** Guerois et al. (2002), **(B)** Potapov et al. (2009), **(C)** Kellogg et al. (2011), **(D)** Jia et al. (2015), **(E)** Quan et al. (2016), curated ProTherm* **(F)** Ó Conchúir et al. (2015). The probability of these classifications occurring given the amino acid composition of the structures in the Curated ProTherm* database are shown in **(G)**. Classes involving charged residues are colored in red. All data sets are significantly biased in their types of mutations present, especially when it comes to mutations to hydrophobics. All data sets contain greater than 27% hydrophobic to hydrophobic mutations vs. the expected 18.4% **(G)**.

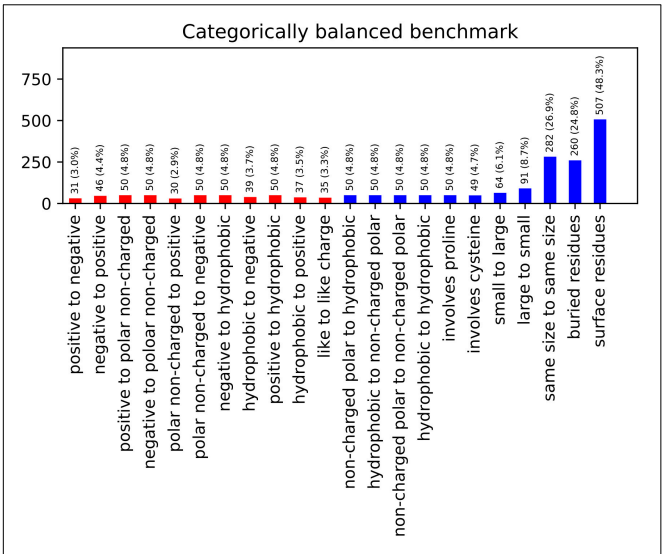


FIGURE 2 | Categorically Balanced Benchmark mutational category statistics. This figure shows the metrics of our new benchmark set selected to provide a more balanced representation of different mutation classifications. Classes involving charged residues are shown in red.

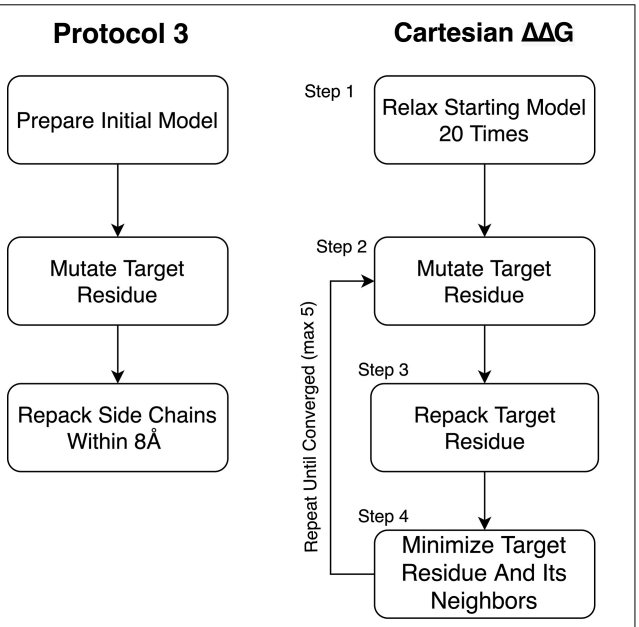


FIGURE 3 | Diagram of Protocol 3 and Cartesian ΔΔG. This figure diagrams the steps involved in the older Protocol 3 as well as the Cartesian ΔΔG protocol. Novel changes described in this paper include the removal of constraints during step 1 of the Cartesian ΔΔG protocol, the addition of Step 2.1 for mutations involving proline as well as the choice to repeat testing until the protocol converges on a lower energy score instead of a fixed number (3) of times.

was moderately incorrect, i.e., the mutation is destabilizing and the prediction was neutral, and 2 indicates the prediction was egregiously wrong.

To address some metrics on which Protocol 3 performs poorly, we were interested in using a more modern Rosetta ΔΔG protocol, Cartesian ΔΔG, first briefly described in Park et al. (2016). We refactored the Cartesian ΔΔG code to utilize the Mover framework described in Leaver-Fay et al. (2011), keeping the underlying science the same (other than changes highlighted here) while eliminating bugs and improving efficiency and modifiability (**Supplementary Table 3**).

We tested changes to the preparation phase of the protocol relative to what was used in Park et al. (2016; **Figure 3**). To improve the preparation step (step 1), we tested Cartesian Relax (as opposed to traditional torsion space Relax, used in Kellogg's Protocol 3) (Kellogg et al., 2011) both with and without all atom constraints and found that Pearson's R correlations were worse when models were prepared without constraints but the Predictive Index and MCC improved. The variability between runs also dropped when models were prepared without constraints. The biggest impact was on the classification of mutations, however, with the number of egregiously wrong predictions falling from 34.00 ± 2.0 to 24.67 ± 0.6 (**Supplementary Table 2**). This likely has to do with the use of Cartesian minimization during step 4, and the importance of preparing a structure with similar sampling methods to those used during mutational energy evaluation. We consider the MCC and Predictive Index improvements more valuable and thus recommend model preparation without all atom constraints.

We also examined a potential runtime improvement for Cartesian ΔΔG. In Park et al. (2016), the final energy for a mutation is the average of three replicates. We examined a multi-run convergence criterion, described in further detail below, and settled on the convergence criterion method due to its equivalent accuracy with reduced run time.

Finally, we tested adding increased backbone sampling around residues that are being mutated to or from proline, which had no impact on the Pearson's R, Predictive Index, and MCC, but reduced the number of egregious errors slightly from 25.33 ± 0.6 to 24.67 ± 0.6 (**Supplementary Table 3**).

This updated Cartesian ΔΔG algorithm has improved performance overall when compared to Protocol 3, especially in the ability to accurately classify mutations including the large reduction of egregious errors in classification (**Tables 1**, **2**). For example, the number of mutations predicted as stabilizing when they are destabilizing or vice versa fell from an average of 53 with Protocol 3 to an average of 24.6 across three replicates. "Off by 1" errors are also lower (317.3 vs. 289.3) (**Supplementary Table 2**). This trend is much stronger than the improvement in correlations, and more importantly reflects the practical value of correctly classifying mutational categories. For example in protein engineering, a protein designer's practical interest is whether any given mutation is stabilizing at all, more than which of two mutations is more stabilizing.

The overall level of accurate classification predictions increases from 51.7 to 59.1% from the Protocol 3 to Cartesian

TABLE 1 | Correlations and Predictive Index for Protocol 3 and our improved Cartesian ΔΔG across different mutation categories.

	Protocol 3				Cartesian ΔΔG			
Mutation Type	**Pearson's R**	**Pearson's R Filtered**	**Predictive Index**	**MCC**	**Pearson's R**	**Pearson's R Filtered**	**Predictive Index**	**MCC**
Small to large	0.54 ± 0.000	0.68 ± 0.000	0.57 ± 0.001	0.36 ± 0	0.48 ± 0.0041	0.66 ± 0.006	0.55 ± 0.0096	0.55 ± 0.0221
Large to small	0.57 ± 0.000	0.76 ± 0.000	0.59 ± 0.000	0.37 ± 0	0.62 ± 0.0199	0.8 ± 0.0015	0.71 ± 0.0239	0.46 ± 0.0176
Positive to negative	0.40 ± 0.000	0.79 ± 0.000	0.28 ± 0.000	0.00 ± 0	0.65 ± 0.0024	0.88 ± 0.0072	0.74 ± 0.0054	0.5 ± 0.0891
Negative to positive	0.34 ± 0.000	0.61 ± 0.000	0.26 ± 0.000	0.19 ± 0	0.36 ± 0.0128	0.57 ± 0.0176	0.48 ± 0.0312	0.53 ± 0.0302
Negative to hydrophobic	0.27 ± 0.000	0.55 ± 0.000	0.27 ± 0.000	0.15 ± 0	0.58 ± 0.0064	0.71 ± 0.0124	0.64 ± 0.0068	0.43 ± 0.0228
Hydrophobic to negative	0.83 ± 0.000	0.87 ± 0.000	0.84 ± 0.000	0.27 ± 0	0.73 ± 0.0554	0.81 ± 0.0477	0.8 ± 0.0867	0.5 ± 0.0551
Positive to hydrophobic	0.06 ± 0.000	0.23 ± 0.000	0.01 ± 0.000	0.35 ± 0	0.46 ± 0.0266	0.62 ± 0.0326	0.51 ± 0.0283	0.66 ± 0.0016
Hydrophobic to positive	0.57 ± 0.000	0.73 ± 0.000	0.63 ± 0.002	0.37 ± 0	0.51 ± 0.0112	0.7 ± 0.0031	0.67 ± 0.0074	0.44 ± 0.0858
Non-charged polar to positive	0.40 ± 0.000	0.67 ± 0.000	0.39 ± 0.004	0.43 ± 0	0.28 ± 0.0075	0.78 ± 0.0148	0.4 ± 0.0196	0.28 ± 0.0
Positive to non-charged polar	0.32 ± 0.000	0.67 ± 0.000	0.52 ± 0.000	0.26 ± 0	0.43 ± 0.0042	0.8 ± 0.0178	0.72 ± 0.0084	0.55 ± 0.0602
Non-charged polar to negative	0.64 ± 0.000	0.73 ± 0.000	0.69 ± 0.000	0.00 ± 0	0.62 ± 0.0196	0.83 ± 0.0042	0.66 ± 0.0153	0.67 ± 0.0
Negative to non-charged polar	0.13 ± 0.000	0.44 ± 0.000	-0.07 ± 0.000	0.22 ± 0	0.37 ± 0.0076	0.69 ± 0.0138	0.44 ± 0.0135	0.53 ± 0.028
Non-charged polar to hydrophobic	0.70 ± 0.000	0.70 ± 0.000	0.64 ± 0.001	0.38 ± 0	0.74 ± 0.0014	0.78 ± 0.0012	0.66 ± 0.0027	0.38 ± 0.0
Hydrophobic to non-charged polar	0.41 ± 0.000	0.66 ± 0.000	0.39 ± 0.000	0.47 ± 0	0.57 ± 0.0105	0.75 ± 0.022	0.58 ± 0.0027	0.11 ± 0.022
Non-charged polar to non-charged polar	0.76 ± 0.000	0.76 ± 0.000	0.66 ± 0.002	0.15 ± 0	0.52 ± 0.0049	0.84 ± 0.0038	0.79 ± 0.0021	0.49 ± 0.0313
Hydrophobic to hydrophobic	0.67 ± 0.000	0.74 ± 0.000	0.72 ± 0.000	0.57 ± 0	0.61 ± 0.0051	0.75 ± 0.0068	0.68 ± 0.0111	0.28 ± 0.0282
charge to charge	0.31 ± 0.000	0.73 ± 0.000	0.36 ± 0.000	0.26 ± 0	0.32 ± 0.0067	0.7 ± 0.0143	0.35 ± 0.0055	0.44 ± 0.0397
Involves cysteine	0.25 ± 0.000	0.63 ± 0.000	0.27 ± 0.000	0.26 ± 0	0.07 ± 0.0428	0.49 ± 0.0946	0.16 ± 0.0498	0.08 ± 0.0708
Involves proline	0.02 ± 0.000	0.54 ± 0.000	0.33 ± 0.000	0.30 ± 0	0.51 ± 0.0277	0.76 ± 0.0264	0.54 ± 0.0271	0.51 ± 0.1401
Same size	0.36 ± 0.000	0.38 ± 0.000	0.37 ± 0.000	0.22 ± 0	0.45 ± 0.0035	0.45 ± 0.0035	0.51 ± 0.0091	0.31 ± 0.0251
Buried	0.20 ± 0.000	0.54 ± 0.000	0.55 ± 0.000	0.35 ± 0	0.43 ± 0.0022	0.43 ± 0.0022	0.54 ± 0.0104	0.26 ± 0.0056
Surface	0.31 ± 0.000	0.34 ± 0.000	0.35 ± 0.000	0.23 ± 0	0.47 ± 0.006	0.5 ± 0.0064	0.6 ± 0.0076	0.42 ± 0.0165
Everything	0.25 ± 0.000	0.47 ± 0.000	0.48 ± 0.000	0.28 ± 0	0.49 ± 0.0025	0.49 ± 0.0025	0.61 ± 0.0062	0.41 ± 0.0127

This table contains the Pearson's R correlations for each class of mutations in our benchmark set for both Protocol 3 and Cartesian ΔΔG. Each is repeated three times using the same inputs and the average and standard deviation are shown. Given the sensitivity to outliers of Pearson's R we also report it as Pearson's R filtered after removing up to five outliers from each set. An outlier is defined as any single entry which, when removed, changes the correlation coefficient by 0.025 or greater. We also report the Predictive Index and the Matthews Correlation Coefficient which are less sensitive to the absolute free energy of a prediction but rather whether it can be correctly classified. Cartesian ΔΔG significantly outperforms Protocol 3 both in the unfiltered Pearson's R, Predictive Index, and Matthews Correlation Coefficient. In each analysis metric the higher value indicates greater accuracy.

ΔΔG Rosetta methods. We also note that over all charged residues the category predictions accuracy was 47.9% for protocol 3 and increases to over 60.5% with Cartesian ΔΔG. The Cartesian ΔΔG algorithm is more broadly useful across any type of protein mutation, while Protocol 3 had uneven applicability.

DISCUSSION

Here we describe a number of issues in previous benchmark sets used to assess the quality of protein stability prediction software. In particular we have found a lack of adequate experimental data being included for mutations involving charged residues.

Using these updated benchmarks we show that protein stability prediction tools in Rosetta vary widely across different types of mutation classes. In addition, given that this problem is pervasive throughout the field, it is likely that the reported accuracy of many methods for stability prediction may not reflect the diversity of possible mutation types. We encourage other developers to analyze the performance of their tools across different types of mutations using our benchmark set or one which has appropriately accounted for the biases that exist within the databases (**Supplementary Table 1**). The reduced size of this data set may also be useful for rapid training or situations with computational limitations.

Last, we have refactored the Cartesian ΔΔG protocol code to improve consistency and modifiability, and have also made minor modifications to the structure preparation and analysis step as well as to how mutations involving proline are sampled. By analyzing these algorithms with the new benchmark set, and

TABLE 2 | The ability of Protocol 3 and Cartesian ΔΔG to correctly classify mutations.

Mutation class	Protocol 3			Cartesian ΔΔG			Total entries
	Same class (%)	Off by one (%)	Off by two (%)	Same class (%)	Off by one (%)	Off by two (%)	
Small to large	55.7	42.7	1.6	67.7	29.2	3.1	64
Large to small	60.4	34.1	5.5	53.5	42.9	3.7	91
Positive to negative	32.3	61.3	6.5	67.7	32.3	0.0	31
Negative to positive	26.1	56.5	17.4	55.1	35.5	9.4	46
Negative to hydrophobic	32.0	50.0	18.0	56.0	38.0	6.0	50
Hydrophobic to negative	53.8	43.6	2.6	70.9	29.1	0.0	39
Positive to hydrophobic	50.0	42.0	8.0	47.3	50.0	2.7	50
Hydrophobic to positive	73.0	18.9	8.1	79.3	18.0	2.7	37
Non-charged polar to positive	50.0	50.0	0.0	61.1	35.6	3.3	30
Positive to non-charged polar	58.0	38.0	4.0	61.3	36.7	2.0	50
Non-charged polar to negative	48.0	48.0	4.0	58.0	40.0	2.0	50
Negative to non-charged polar	44.0	36.0	20.0	50.7	47.3	2.0	50
Non-charged polar to hydrophobic	48.0	48.0	4.0	48.7	49.3	2.0	50
Hydrophobic to non-charged polar	56.0	42.0	2.0	66.0	28.7	5.3	50
Non-charged polar to non-charged polar	57.3	38.7	4.0	67.3	30.7	2.0	50
Hydrophobic to hydrophobic	76.0	22.0	2.0	72.0	26.0	2.0	50
Charge to charge	65.7	34.3	0.0	51.4	48.6	0.0	35
Involves cysteine	55.1	40.8	4.1	49.3	44.0	6.7	49
Involves proline	54.0	38.0	8.0	52.0	44.0	4.0	50
Buried	65.4	26.5	8.1	65.0	29.4	5.6	260
Surface	44.7	49.0	6.3	56.1	41.9	2.0	507
Everything	51.7	41.4	6.9	59.1	37.7	3.2	767

This table shows the ability of Protocol 3 and Cartesian ΔΔG to correctly classify a mutation. Mutations are assigned a value of 0 for destabilizing, 1 for neutral, and 2 for stabilizing. The absolute value of the difference between the predicted class and the experimental class represents no error, mild error (off by one class), or egregious error (off by two classes). Performance across the benchmark is reported here as a percentage of mutations in each class. Cartesian ΔΔG correctly classifies more entries (59.1 vs. 51.7%), and produces fewer catastrophic off by two errors (3.2 vs. 6.9%).

focusing on previously underrepresented categories of mutations (e.g., uncharged to charged), we are able to demonstrate the Cartesian ΔΔG algorithm has improved correlation to experimental values and improved ability to correctly classify (stabilizing/destabilizing/neutral) a mutation relative to the older Protocol 3 methodology. These results show the importance of diverse datasets in algorithm benchmarking, and the need to look beyond the surface when analyzing the results of these algorithms.

METHODS

Benchmark Set Pruning

To create our benchmark set, we began by making a copy of the curated ProTherm* database (Ó Conchúir et al., 2015) and began removing entries that were unsuitable. Because we wished to develop a point mutation algorithm without the complexities of multiple mutation interactions, we excluded any entry which did not represent a single mutation. Because the algorithm is intended to represent ΔΔG of monomer folding and not binding interactions, we also removed entries on the interface of a protein-protein complex, or interacting with a non-water ligand. Interactions were defined as any atom in the mutated residue within 5 Å of an atom not on the same chain. To increase experimental diversity, we wished to remove duplicate mutations. To identify duplicates, we performed an all to all sequence

TABLE 3 | Residue category assignments and category combinations.

Type and 1 letter codes	Combination categories	
Small GAVSTC	Positive to negative	Non-charged polar to hydrophobic
Large FYWKRHQE	Negative to positive	Hydrophobic to non-charged polar
Negative DE	Positive to non-charged polar	Non- charged polar to non-charged polar
Positive RK	Negative to non-charged polar	Hydrophobic to hydrophobic
Polar YTSHKREDQN	Non-charged polar to positive	Like to like charge
Non-charged polar YTSNQH	Non-charged polar to negative	Involves proline
Hydrophobic FILVAGMW	Negative to hydrophobic	Involves cysteine
Cysteine C	Hydrophobic to negative	Small to large
Proline P	Positive to hydrophobic	Large to small
	Hydrophobic to positive	

On the left, we list the nine residue groupings considered in this benchmark and annotate which residues go in each class. At center and right, we list the 19 mutational types considered by combining these classes.

alignment to find parent backbones with ≥60% sequence identity. Within these clusters of sequences, any entries in which the same native residue is mutated to the same target were treated as identical. When multiple experimental ΔΔG values were available for an identical mutation we chose the value taken at closer to neutral pH.

Benchmark Category Population

We identified 21 categories of mutation type by combinations from nine residue type classifications (**Table 3**). We then populated each narrowly defined category (e.g., polar non-charged to negative) with up to 50 entries. Some categories (large to small, small to large, buried, and surface) are supersets of the more narrowly defined categories and were sufficiently populated by the experiments selected from the other groups.

A few categories involving charged residues (positive to negative, negative to positive, non-charged polar to positive, hydrophobic to negative, hydrophobic to positive, and like to like charge) did not have enough data to hit 50 entries so every available unique experiment was added.

ΔΔG Prediction

To prepare models for ΔΔG calculations, structures were stripped to only the chain in which the mutation occurs. Rosetta local refinement, consisting of alternating cycles of side chain packing and all atom minimization ("Relax"), was then performed 20 times on the chain of interest and the model with the lowest Rosetta energy was selected as input. As noted in the Results, this was done without all atom constraints and in Cartesian space, not torsional space.

ΔΔG predictions were then performed using Protocol 3 described in Kellogg et al. (2011), the version of the Cartesian ΔΔG application described originally in Park et al. (2016), or the refactored and improved version of Cartesian ΔΔG elaborated upon here.

To provide context for our modifications, a brief description of the Cartesian ΔΔG protocol as presented in Park et al. (2016) is warranted. Cartesian ΔΔG calculates the change in folding energy upon mutation by taking the prepared starting structures, then mutating the target residue. This residue and its neighbors within 6 Å are then repacked. After repacking the mutated residue, the side chain atoms of residues within 6 Å of the target residue and the side chain and backbone atoms of sequence-adjacent residues are minimized in Cartesian space. The same optimization, without the change in sequence, is done on the starting structure to determine the baseline energy. The process is performed three times for both the mutant and the wild type sequence and the ΔΔG is calculated from the average of each. There is no particularly different handling of mutations involving proline.

Our modifications to the Cartesian ΔΔG tool include a change to the analysis and a change to proline handling (**Figure 2**). In the analysis step, we changed the number of mutant models generated using the following convergence criterion: the lowest energy 2 structures must converge to within 1 Rosetta Energy Unit, or take the best of 5 models, whichever comes first. In either case the lowest, not the average, energy is used. In order to address changes in the backbone resulting from mutations to and from proline we added additional fragment based sampling around mutations involving proline. By default 30 fragments of 5 residues in length, centered on the mutation, are sampled and the best scoring structure is carried forward for analysis. This uses the Cartesian Sampler system described in Wang et al. (2016). Command line flags and XML files can be found in **Supplementary Material**.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/**Supplementary Material**. The protocol and source code are freely available for academic use in the Rosetta software suite found at: https://www.rosettacommons.org/.

AUTHOR CONTRIBUTIONS

BF carried out the primary work and prepared the manuscript. SL analyzed data and edited the manuscript. IK analyzed data. FD and HP helped develop the initial code and provided guidance. YS supervised the work and provided assistance. All authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

This manuscript has been released as a pre-print at https://doi.org/10.1101/2020.03.18.989657 (Frenz et al., 2020). This work was supported in part by the US National Institutes of Health under award numbers R01GM123089 (FD).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.558247/full#supplementary-material

REFERENCES

Benedix, A., Becker, C. M., de Groot, B. L., Caflisch, A., and Böckmann, R. A. (2009). Predicting free energy changes using structural ensembles. *Nat. Methods* 6, 3–4. doi: 10.1038/nmeth0109-3

Capriotti, E., Fariselli, P., and Casadio, R. (2005). I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res.* 33, W306–W310.

Casadio, R., Compiani, M., Fariselli, P., and Vivarelli, F. (1995). Predicting free energy contributions to the conformational stability of folded proteins from the residue sequence with radial basis function networks. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 3, 81–88.

Frenz, B., Lewis, S., King, I., Park, H., DiMaio, F., and Song, Y. (2020). Prediction of protein mutational free energy: benchmark and sampling improvements increase classification accuracy. *bioRxiv*[Preprint] doi: 10.1101/2020.03.18.989657

Gilis, D., and Rooman, M. (1996). Stability changes upon mutation of solvent-accessible residues in proteins evaluated by database-derived potentials. *J. Mol. Biol.* 257, 1112–1126. doi: 10.1006/jmbi.1996.0226

Guerois, R., Nielsen, J. E., and Serrano, L. (2002). Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J. Mol. Biol.* 320, 369–387. doi: 10.1016/s0022-2836(02)00442-4

Jia, L., Yarlagadda, R., and Reed, C. C. (2015). Structure based thermostability prediction models for protein single point mutations with machine learning tools. *PLoS One* 10:e0138022. doi: 10.1371/journal.pone.0138022

Kellogg, E. H., Leaver-Fay, A., and Baker, D. (2011). Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins* 79, 830–838. doi: 10.1002/prot.22921

Kumar, P., Henikoff, S., and Ng, P. C. (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat. Protoc.* 4, 1073–1081. doi: 10.1038/nprot.2009.86

Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., et al. (2011). Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. *Methods Enzymol.* 487, 545–574.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* 405, 442–451. doi: 10.1016/0005-2795(75)90109-9

Ó Conchúir, S., Barlow, K. A., Pache, R. A., Ollikainen, N., Kundert, K., O'Meara, M. J., et al. (2015). A web resource for standardized benchmark datasets, metrics, and rosetta protocols for macromolecular modeling and design. *PLoS One* 10:e0130433. doi: 10.1371/journal.pone.0130433

Park, H., Bradley, P., Greisen, P. Jr., Liu, Y., Mulligan, V. K., Kim, D. E., et al. (2016). Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *J. Chem. Theory Comput.* 12, 6201–6212. doi: 10.1021/acs.jctc.6b00819

Pearlman, D. A., and Charifson, P. S. (2001). Are free energy calculations useful in practice? A comparison with rapid scoring functions for the p38 MAP kinase protein system†. *J. Med. Chem.* 44, 3417–3423. doi: 10.1021/jm0100279

Pitera, J. W., and Kollman, P. A. (2000). Exhaustive mutagenesis in silico: multicoordinate free energy calculations on proteins and peptides. *Proteins* 41, 385–397. doi: 10.1002/1097-0134(20001115)41:3<385::aid-prot100>3.0.co;2-r

Pokala, N., and Handel, T. M. (2005). Energy functions for protein design: adjustment with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *J. Mol. Biol.* 347, 203–227. doi: 10.1016/j.jmb.2004.12.019

Potapov, V., Cohen, M., and Schreiber, G. (2009). Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Eng. Des. Sel.* 22, 553–560. doi: 10.1093/protein/gzp030

Quan, L., Lv, Q., and Zhang, Y. (2016). STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics* 32, 2936–2946. doi: 10.1093/bioinformatics/btw361

Sippl, M. J. (1995). Knowledge-based potentials for proteins. *Curr. Opin. Struct. Biol.* 5, 229–235. doi: 10.1016/0959-440x(95)80081-6

Uedaira, H., Gromiha, M. M., Kitajima, K., and Sarai, A. (2002). ProTherm: thermodynamic database for proteins and mutants. *Seibutsu Butsuri Kagaku* 42, 276–278. doi: 10.2142/biophys.42.276

Wang, R. Y.-R., Song, Y., Barad, B. A., Cheng, Y., Fraser, J. S., and DiMaio, F. (2016). Automated structure refinement of macromolecular assemblies from cryo-EM maps using Rosetta. *eLife* 5:e17219. doi: 10.7554/eLife.17219

Conflict of Interest: Cyrus Biotechnology, Inc. funded and designed this study, was responsible for collection, analysis, and interpretation of the data as well as the writing of this article and decision to submit it for publication. They also provided software and infrastructure and employed BF, SL, IK, and YS. The product Cyrus Bench was currently marketed and uses software and methods developed in this study.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

METHODS
published: 23 October 2020
doi: 10.3389/fbioe.2020.568318

Elfin UI: A Graphical Interface for Protein Design With Modular Building Blocks

*Chun-Ting Yeh[1], Leon Obendorf[1,2] and Fabio Parmeggiani[1,3]**

[1] School of Chemistry and School of Biochemistry, University of Bristol, Bristol, United Kingdom, [2] Institute of Chemistry and Biochemistry, Freie Universität Berlin, Berlin, Germany, [3] Bristol Biodesign Institute and BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre, University of Bristol, Bristol, United Kingdom

Molecular models have enabled understanding of biological structures and functions and allowed design of novel macro-molecules. Graphical user interfaces (GUIs) in molecular modeling are generally focused on atomic representations, but, especially for proteins, do not usually address designs of complex and large architectures, from nanometers to microns. Therefore, we have developed Elfin UI as a Blender add-on for the interactive design of large protein architectures with custom shapes. Elfin UI relies on compatible building blocks to design single- and multiple-chain protein structures. The software can be used: (1) as an interactive environment to explore building blocks combinations; and (2) as a computer aided design (CAD) tool to define target shapes that guide automated design. Elfin UI allows users to rapidly build new protein shapes, without the need to focus on amino acid sequence, and aims to make design of proteins and protein-based materials intuitive and accessible to researchers and members of the general public with limited expertise in protein engineering.

Keywords: protein design, blender, GUI, repeat proteins, computational modeling

OPEN ACCESS

Edited by:
Jose Ruben Morones-Ramirez, Autonomous University of Nuevo León, Mexico

Reviewed by:
Mario Andrea Marchisio, Tianjin University, China
Thomas Dandekar, Julius Maximilian University of Würzburg, Germany
Jean Vanderdonckt, Catholic University of Louvain, Belgium

***Correspondence:**
Fabio Parmeggiani
fabio.parmeggiani@bristol.ac.uk

Specialty section:
This article was submitted to Synthetic Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *31 May 2020*
Accepted: *02 October 2020*
Published: *23 October 2020*

Citation:
Yeh C-T, Obendorf L and Parmeggiani F (2020) Elfin UI: A Graphical Interface for Protein Design With Modular Building Blocks. Front. Bioeng. Biotechnol. 8:568318. doi: 10.3389/fbioe.2020.568318

INTRODUCTION

Visualization and simulation of macromolecules have enabled our understanding of biological structures and have led to the development of a variety of tools for research, teaching and outreach, working at multiple scales (Johnson and Hertig, 2014).

Visualizing structures made also possible to design them, by taking into account the spatial relationship between different parts of the molecules. Dedicated software packages have emerged over the years for protein design, reviewed by Gainza et al. (2016), and popular viewers such as Chimera (Pettersen et al., 2004), PyMOL (The PyMOL Molecular Graphics System, Version 2.0 Schrödinger, LLC) (DeLano, 2002), and VMD (Humphrey et al., 1996) have now integrated design capabilities.

Protein design tools focus largely on atomic models and sequence design from a given backbone structure. Additionally, several approaches allow to build completely new structures by relying on secondary structure description and fragments assembly, like Rosetta remodel and blueprint builder (Huang et al., 2011; Koga et al., 2012), parametric design, as in Isambard (Wood et al., 2017), idealized secondary structures, e.g., CoCoPOD (Ljubetič et al., 2017) and TopoBuilder (Sesterhenn et al., 2020), or building blocks with super-secondary structures, as in SEWING (Jacobs et al., 2016) and Elfin (Yeh et al., 2018). Protein complexes have been successfully designed for symmetric systems, e.g., point group symmetry (Lai et al., 2012; King et al., 2014; Hsia et al., 2016) and lattices

(Lanci et al., 2012; Gonen et al., 2015), but large, precise and asymmetric assemblies are still a challenge. However, such scaffolds could prove particularly interesting in modulating cell surface receptor clustering and signaling via precise ligand organization and placement (Grochmal et al., 2013; Jost et al., 2013; Shaw et al., 2014; Mohan et al., 2019).

To address the challenge of building large structures, both symmetric and non-symmetric, DNA nanotechnology groups have led the way in developing Computer Aided Design (CAD) software, e.g., Tiamat (Williams et al., 2009), cadnano (Douglas et al., 2009), CanDo (Veneziano et al., 2016), vHelix (Benson et al., 2015), taking advantage of base pairing and regularity of DNA double helix structure.

Graphical User interfaces (GUI) have indeed a key role in making software accessible to a broad group of users, who are not necessarily expert, by enabling work on design principles, rather than biochemical details. While CAD tools for DNA nanostructures allow users to work purely on intuitive geometric concepts, e.g., shapes to achieve, protein design tools often require a more in-depth programming and biochemical knowledge. GUIs have been developed for the Rosetta modeling suite to improve usability (Adolf-Bryfogle and Dunbrack, 2013; Schenkelberg and Bystroff, 2015) and the protein folding game Foldit (Cooper et al., 2010) has successfully attracted a broad base of users from the general public. Its standalone interface (Kleffner et al., 2017) has become an instrument to interactively design new proteins, although designs are effectively limited to a few hundred amino acids, if systems are not symmetric.

Size is one of the major limitations in interactive protein design using atomic models, as the number of atoms quickly becomes the computational bottleneck. However, it is possible to take a more coarse-grained approach to design large and complex protein architectures, akin to DNA nanostructure designs.

In this work we have developed a user interface to allow design of protein structures using modular structural building blocks. Elfin user interface (Elfin UI) was developed as a graphical interface and an interactive editor to the Elfin software package (Yeh et al., 2018) for design of custom protein architectures (**Figure 1**). Elfin uses structural compatible building blocks (referred to as modules) derived from experimentally validated structures of repeat proteins to build large and complex architectures. The goals were to provide (1) a CAD-like environment for design of user-defined shapes, to which Elfin could find solutions in terms of protein sequence and structures, and (2) a sandbox framework to interactively explore potential protein architectures. We envision Elfin UI to be used in the design of protein origami, custom shaped nanoparticles and scaffolds for organization of enzymes and signaling molecules.

We have implemented Elfin UI as a Blender add-on. Blender is a popular free, open source and cross-platform 3D modeling application, which has been successfully extended with add-ons to integrate molecular viewers, like BlendMol (Durrant, 2019), BioBlender (Andrei et al., 2012), ePMV (Johnson et al., 2011), Pyrite (Rajendiran and Durrant, 2018).

By using modular compatible building blocks and a coarse-grained representation, we aim to provide a tool accessible to scientists, both expert and novice in protein design, and a new way to engage the public with the concepts of modular design and manufacturing using biological macromolecules.

METHODS

The Elfin software package is built around the Elfin solver, a genetic algorithm for the assembly of modular structures matching a user defined shape (Yeh et al., 2018), and contains an updated database with information about modular building blocks, a graphical user interface (Elfin UI) built as Blender add-on, and ancillary utility scripts (e.g., for installation, database preparation, file conversion). Code, documentation, installation scripts and tutorials are available on https://github.com/Parmeggiani-Lab/elfin.

Elfin UI's approach to protein design is similar to the idea of Model-Based UI Design (Calvary et al., 2003). In this framework, Elfin UI uses a database of individual proteins and termini compatibility matrix as the domain model. The task of protein design is undertaken by arranging and joining two or more protein modules to form the shape desired by the user. Each protein module is abstractly represented by attributes such as its center-of-mass, collision radius, and module pairwise transformation matrices. A design assembled by the user is converted into an atomic model by projecting atomic coordinates of each protein module onto their respective position and adding capping modules to each "free" termini to protect the otherwise exposed hydrophobic core and improve solubility (**Supplementary Figure S1**). Finally, if the designed protein's atomic structure passes third party verification (e.g., Rosetta, see **Supplementary Materials**), it is considered suitable to be produced and characterized experimentally.

Database

Elfin builds protein architectures using combinations of structural building blocks. Building blocks are stored as collection of atomic coordinates in The Protein Data Bank (PDB) format and used to precompute: (1) a JSON database, which includes, for each module, the center of mass and radius, a list of compatible modules and relative orientation of the pairs, expressed as rigid body transforms; (2) a Blender database that stores meshes of each module with cartoon representations of secondary structure elements.

Modules are classified as: core, when they are extracted from designed repeat proteins (Parmeggiani and Huang, 2017) and contain repeated super-secondary structure (e.g., helix-loop-helix-loop); junction, if they contain two contiguous and merged super-secondary structures typical of core modules (so that the module acts as a junction between core modules); or hub, if they are formed by multiple interacting chains. Core and junction modules are single chains that can be extended by adding a further module to the chain either at the N- or C- terminus. Some hubs' chains can be extended only at one terminus, if the other is involved in binding another chain.

Core modules have a specific name, like D4, proA, darp. Junctions include the name of the core modules that they bridge with a j (for junction) followed by a number, since there can be

multiple junctions between two core modules: e.g., D14_j1_D79, D14_j2_D79. The name indicates that they are compatible at the N-term with modules that possess a C-term interface of the same kind (anything ending in D14 in this case). Same for the C-term. Core modules are compatible, by definition, with themselves and with junctions with compatible ends. Hub names indicate the type of core module that they contain and eventually information about the number of subunits and type of symmetry, e.g., D4_C4 is a cyclic homo-tetramer of D4-derived units.

Modules form a continuous hydrophobic core that runs through each chain. As for repeat proteins (Parmeggiani and Huang, 2017), the core needs to be sealed off at the termini by modified repeating units, called capping repeats or caps, with the same structural unit of the last module: e.g., a D14 and a D49_j1_D14 module, placed at the C-term, require capping by Ccap_D14. Caps are added only at the final stage when a JSON file from Elfin UI or Elfin solver is converted into an atomic model in mmCIF format by the stitch.py script. mmCIF is the standard format for the Worldwide Protein Data Bank (wwPDB) and removes limitations on the number of atoms and chains present in the previous PDB format. Modules in the database are still stored as PDB

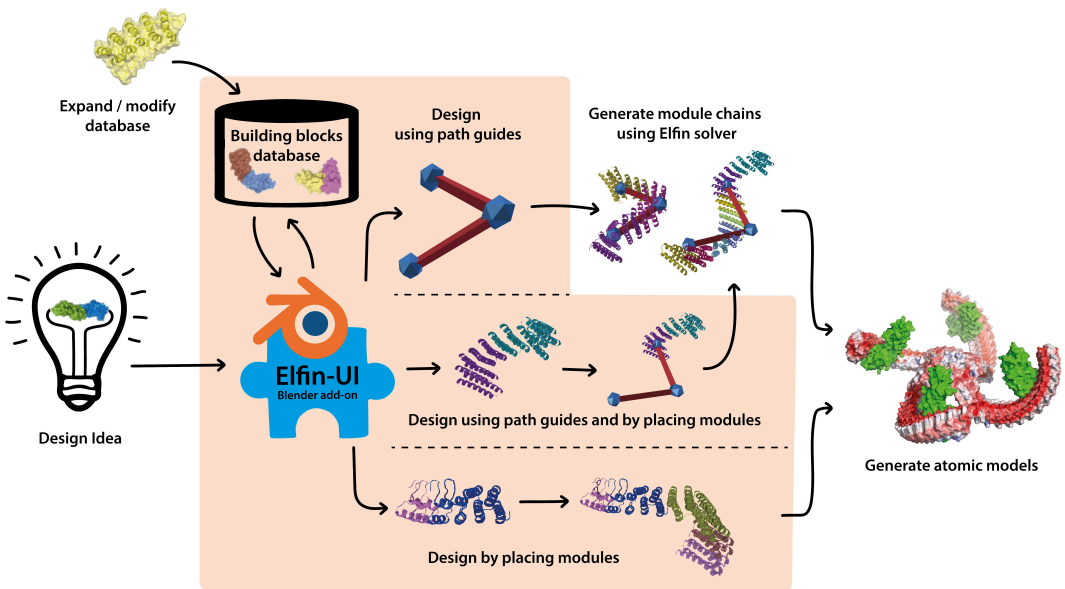


FIGURE 1 | Elfin UI is a Blender (blender.org) add-on that enables interactive coarse-grained design of proteins using combinations of pre-existing and validated building blocks. The shaded orange area indicates the functionalities of Elfin UI within the design process. Designs can be built by defining the desired shape and searching for matching building blocks combinations, by manually placing the building blocks, or by a combination of the two methods. Coarse grained representations are then converted to atomic model outputs in mmCIF format.

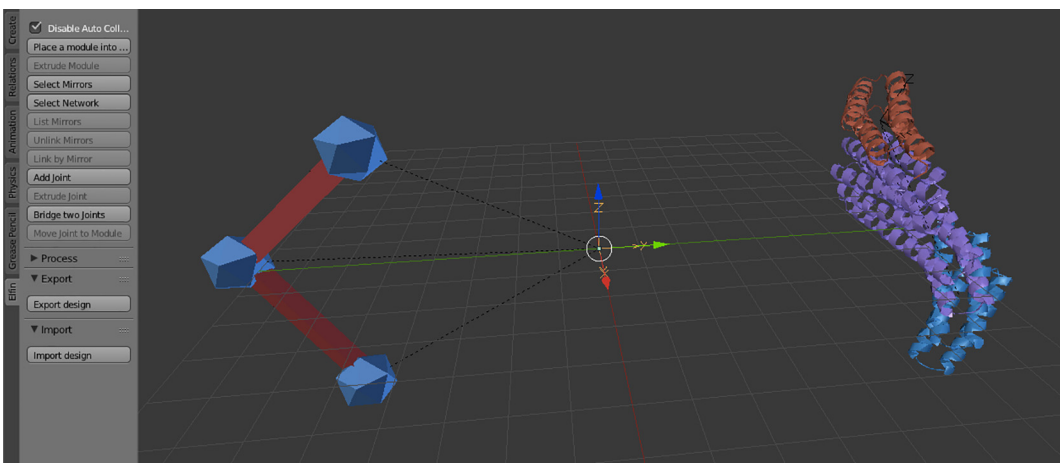


FIGURE 2 | The Elfin UI Blender add-on interface. The Elfin panel on the left shows the accessible operators. On the Blender scene, on the left is a path guide composed of three joints (blue icospheres) and two bridges (red), and on the right a protein formed by three modules, in different colors.

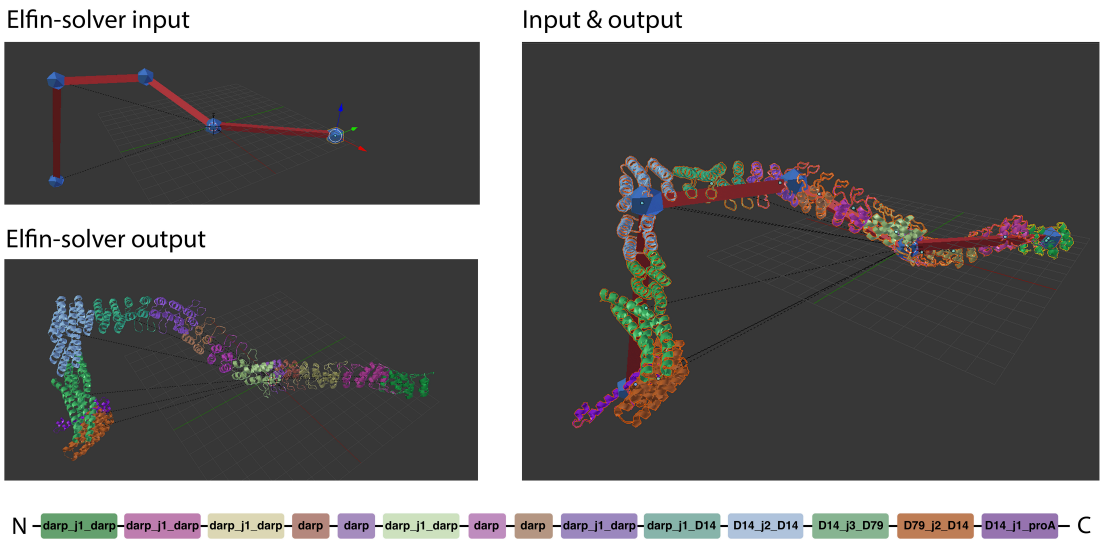


FIGURE 3 | Path guide building. Elfin UI allows to define a network of joints and bridges that can be used as input for Elfin solver. The designed output can be superimposed on the initial path guide. The colors indicate the different building blocks highlighted in the sequence at the bottom.

files, as the number of atoms is limited and within the capacity of the format.

Current modules are derived from published and experimentally verified structures. Core modules are extracted from designed helical repeats (DHRs) (Brunette et al., 2015), designed ankyrin repeats proteins (darpins) (Kramer et al., 2010) and protein A (Youn et al., 2017). Junctions were designed using either an helix fusion method (Wu et al., 2017; Youn et al., 2017) or *de novo* connecting helices (Brunette et al., 2020). Hubs were derived from oligomeric repeat proteins (Fallas et al., 2017). **Supplementary Table S1** contains a detailed list of modules and sources. Custom databases can be created using the scripts provided with the Elfin source code. The workflow is described in the **Supplementary Figure S2**.

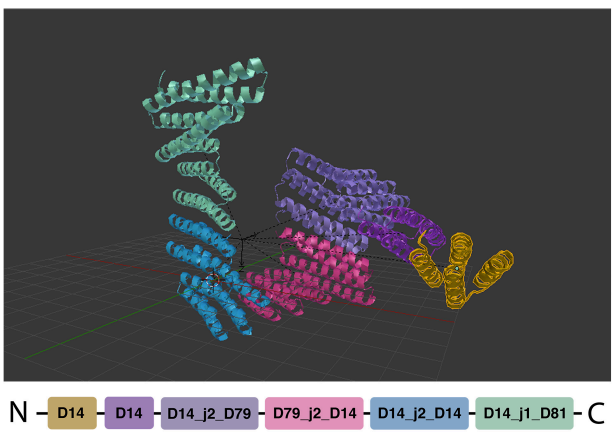


FIGURE 4 | Manual module placement. The single chain protein is built as a sequence of compatible modules, depicted in different colors.

Blender Add-On Implementation

Elfin UI was developed in python 2.7 as an add-on to blender 2.79. Currently it is not yet compatible with Blender 2.8. As Blender add-on, Elfin UI creates a context menu and adds sections in the side panel, but primarily interacts with objects in the scene by defining "operators" that apply some routine on selected objects. These operators can be invoked using shortcuts, by clicking context menu buttons, or looked up and called from the search menu. Elfin UI plugin defines many such operators to facilitate two main design processes: (1) path guide building, and (2) manual module placement (see results for description). Whenever objects (either protein modules or path guide components) are created through Elfin UI's operators, the object is spawned with a property group dedicated to storing Elfin's information. It stores the object type (module or path guide), link occupancy (who are the neighbors), and helper attributes such as a flag to indicate whether the object needs to be cleaned up by Elfin's object lifetime watcher. Other than object-specific information, data such as module compatibility and 3D models are loaded only once and stored in a singleton object until either Blender is closed, the add-on is reloaded, or when the user explicitly calls the reload operator.

Module compatibility is explicitly embedded in the prototype naming convention for module operators. Place Module and Extrude Module operators prompt the user with a filtered list of actionable module names (*filtered prototypes*). There could be many modules in a scene, but modules with the same name (e.g., D4.001, D4.002) are of the same prototype (D4). For extrusion, prototypes are filtered by compatibility and also terminus occupancy (i.e., is the N and/or C terminus already occupied?).

For Place Module, the name of each module is bounded by two period marks. These marks make it easy to search the exact module the user is looking for: e.g., a search for .D4 will return all modules with a name starting in D4.

For Extrude Module, names are in the form: : < chain1 > (< term1 >) - > (< term2 >) < chain2 > : < name2 > .

The chain1 and term1 are chain ID and terminus type of the module being extruded from. The term2, chain2, and name2 are corresponding attributes of the new module to be extruded into. For instance: when D49 is selected and extrusion on the N terminus is chosen, one of the items in the list could be:A(N) - > (C)A:D49_aC2_24. This means the terminus N of chain A of D49 can be extruded and connected to a yet-to-be-added D49_aC2_24 hub. In the latter, terminus C of chain A would be used for this connection. The first letter, if there is one, denotes the C-Terminus chain ID of the extrusion. This is needed because hub modules have more than one chain to extrude to and from. The last letter is therefore the N-Terminus chain ID in the to-be-extruded module.

Groups of modules or path guide primitives are organized in networks that keep track of which modules or path guides are "connected." Networks are displayed in Blender outliner view. While individual path guide "joints" can be freely rotated

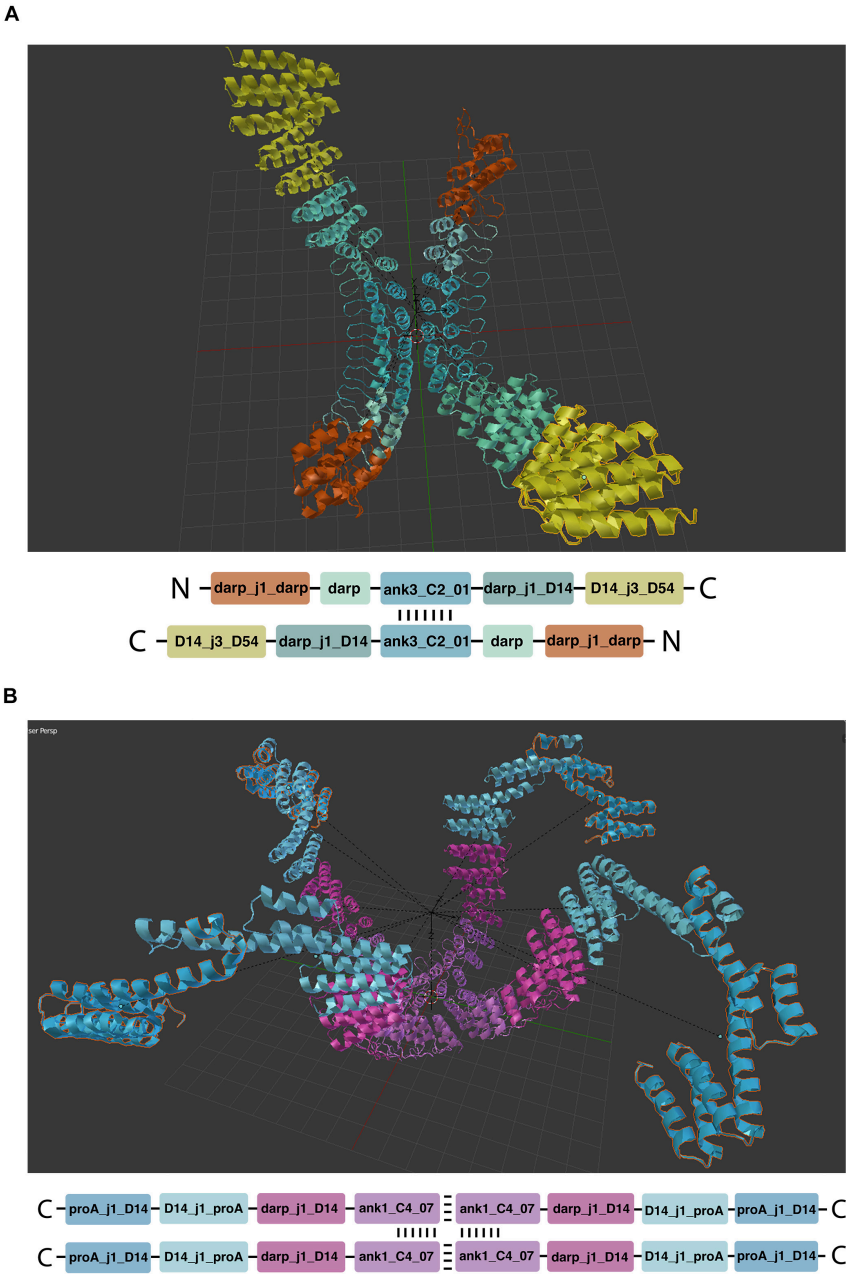


FIGURE 5 | Symmetric structures. **(A,B)** Show respectively two and four chain architectures. The oligomeric module (hub) is indicated by the repeated vertical and horizontal dashes.

and translated, Elfin UI locks individual modules. However, whole networks can be rotated and translated because they preserve the interface relationship of each connected group of modules. Creation and splitting of networks are automatic, and ease processing when exporting. Joining of two networks is also possible, subject to termini compatibility.

When designing using Elfin UI, live collision detection between protein modules can be turned on or off from the left side pane (default shortcut is T). When it is turned on, newly placed protein objects that result in collision will raise a clear warning on screen.

Since Elfin UI supports "partial design"—a design specification consisting of a network of path guide components overlapping manually placed modules, sanity checks such as overlap intention and link availability are conducted behind the scenes.

RESULTS

Elfin UI is part of the Elfin tool set that allows the user to design proteins with complex 3D shapes protein designs. In Elfin, a three-dimensional structure, defined as a network of nodes and edges, is translated into a protein structure using a combination of compatible structural building blocks, referred to as modules. Different module databases can be used and users can build their own, as described in the **Supplementary Materials.**

As an add-on, Elfin UI borrows Blender's graphical interface to enable the generation of 3D structures to facilitate two main design processes: (1) path guide building, and (2) manual module placement.

Path guides are 3D objects, formed by nodes and edges, that describe the geometry of a three-dimensional shape. Path guides can be exported to Elfin Solver (the core algorithm in Elfin), which generates a protein structure to fit, as close as possible, the defined 3D shape.

Alternatively, protein modules, which correspond to super-secondary structural elements (e.g., sets of alpha helices and beta sheets), can be manually placed. The protein chain can be then extended by adding compatible modules, allowing for a stepwise and interactive building of protein structures.

Elfin UI introduces a new panel of options in Blender and new import and export features that enable path guide building, manual module placement and hybrid designs.

Blender Interface

Elfin UI specific controls are located in an "elfin" panel in the Blender interface (**Figure 2**). The commands, called operators, allow paths guide building and module placement. Depending on current selected objects, only allowed operators can be used. Operators are also available in the search menu, accessible using the spacebar, in Blender 2.79. Every operator has a hashtag-three-letters-shortcut that, when entered in the search menu, immediately brings up that operator, speeding up the design process. E.g., the module extrusion operator is "#exm." Operators' detailed descriptions are available in the Elfin UI tutorial: https://github.com/Parmeggiani-Lab/elfin-ui/blob/master/resources/tutorial/README.md. Blender operators, like delete, work on these objects.

Modules are represented by meshes, derived from PyMol (DeLano, 2002) that depict protein secondary structures (helices, beta sheets and loops) and have been scaled appropriately: each square in the reference plane of the default Blender working space is 1 nm long. Interactions and relative positions are precomputed and stored in a database file, therefore, to preserve the relationships, module scaling is not allowed.

Elfin UI allows export of path guides and designed proteins as JSON files, which contain information about connectivity, type of modules (if present) and three-dimensional coordinates. Elfin solutions, produced as JSON files, contain a network of modules and can be imported in Elfin UI for visualization. JSON was chosen for its human-readability (which facilitates debugging and easy extension), ease to parse, and because there is not a large amount of data to justify size-efficient formats, such as binary formats.

Elfin UI is a module-centric interface and does not support atom or residue level views. Atomic models, in mmCIF format, are generated from json files by a script (stitch.py) in the Elfin tool set (see **Supplementary Figure S1** for details). Output files can be then visualized using molecular viewers (e.g., PyMol, Chimera, VMD) or loaded in any program that supports mmCIF files for energy minimization, molecular dynamics simulations or further design. After conversion from the modular coarse-grained representation to atomic coordinates, we perform energy minimization and relaxation in Rosetta (Leman et al., 2020) to ensure that the design shape is maintained (see **Supplementary Materials**).

Path Guide Building

Path guides are the objects that guide Elfin Solver to build a protein that most resembles the user's design intent. Path guides are not protein modules; they are simple geometry specifications expressed as "joints" and "bridges." These are synonymous to "nodes" and "edges" in mathematical terms, and in Blender, they are represented with premade icosphere and elongated cubes respectively (**Figure 2**).

The main path guide operators are:

- Add joint: place a new joint in space
- Extrude joint: create a new joint in the desired position connected to the current joint with a bridge
- Bridge two joints: create a new bridge between joints

When connecting between joints, bridges will stretch and contract visually according to the actual distance between the joints. Joints and bridges can be used to define complex networks. Since the distance between joints can be arbitrarily defined, there may not always be a solution in which protein modules can satisfy the path guide design, but Elfin Solver always tries to optimize.

After a design has been drawn out by the user, it can be exported into a JSON format that Elfin Solver reads and

processes. The optimized solution is saved into a JSON file that Elfin UI can read back into Blender and display as a 3D model (**Figure 3**).

Path guides are used to define arbitrary shapes that the user is interested in. If the goal is a precise geometry in 2D or 3D, the coordinates for each node can be inputted directly in Blender.

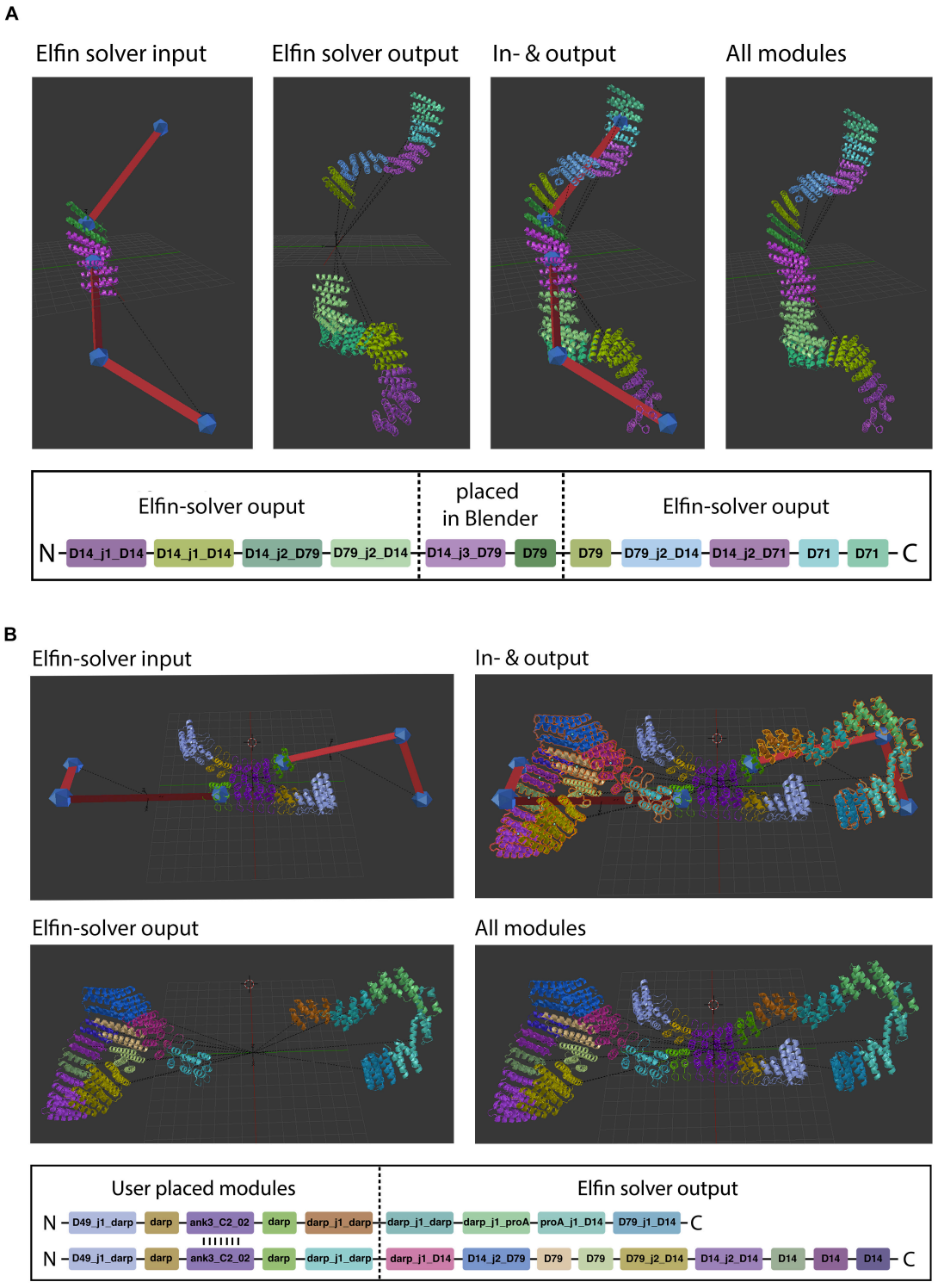


FIGURE 6 | Hybrid design. Elfin UI allows users to build shapes that include selected modules in specific positions. The path guide parts are solved by Elfin solver and merged in Elfin UI. **(A,B)** Show single-chain and two-chains hybrid designs, respectively.

Manual Module Placement

Elfin UI can be used as a sandbox environment to interactively explore the construction of complex protein architectures. Users can select modules and place them directly into the scene, growing chains progressively by addition of new compatible modules (**Figure 4**). When a new module is placed the color can be changed. If a new module causes clashes with the existing chains, an error box is raised, preventing the addition. This check can be disabled by toggling the auto_collision_check box in the elfin panel.

The main module operators are:

- Place modules: place a new module in the scene
- Extrude module: place a new module next to the current one extending the protein chain; the new module is selected among the compatible ones
- Link by mirror: associate two or more identical modules; when one of these modules is extruded, all the linked ones are extruded accordingly, if the extrusion is possible. Added modules are considered linked to each other
- Unlink mirror: remove the mirror linkage, so that extrusion can be performed independently.

Modules are derived from existing experimental structures (Kramer et al., 2010; Brunette et al., 2015, 2020; Wu et al., 2017; Youn et al., 2017) and connected through peptide bonds. The interfaces between modules and their relative orientation are also derived from crystal structures and SAXS-confirmed models, ensuring a correct module placement. This information is stored in the elfin and Blender databases (see section "Methods").

Mirror linking is used to build symmetric structures or structures containing only some symmetric parts (**Figure 5**). Mirror-linked modules need to be of the same type. Modules derived from experimentally validated oligomers (Fallas et al., 2017) contain multiple chains that can potentially grow in a symmetric fashion, when the same module is added to each chain. Symmetric hubs are automatically mirror-linked. Modules extruded from mirror-linked modules are automatically mirror-linked.

Hybrid Design

Manual module placement can be used in conjunction with path guides to partially define a design, if the user already knows what protein module needs to be positioned (e.g., predefined binding sites) and in which orientation (**Figure 6**). The user places modules directly into the scene and translates and rotates them.

When a protein module is placed directly on a path guide joint, Elfin UI infers that the bridges connecting to that joint are intended to be "extrusions" from the protein module. The "move joint to module" operator allows to place an existing joint on a module, after selecting both.

Hybrid design can be used when the position and orientation of specific modules of the desired protein are known. By building a guide path from them, elfin will search for a compatible solution to connect the modules. The initial input and the design output should be then combined in a single network, using the "join network" operator to obtain the combined structure. This approach can be used, for example, to build multivalent ligands to engage multiple cell receptors at the same time, by placing

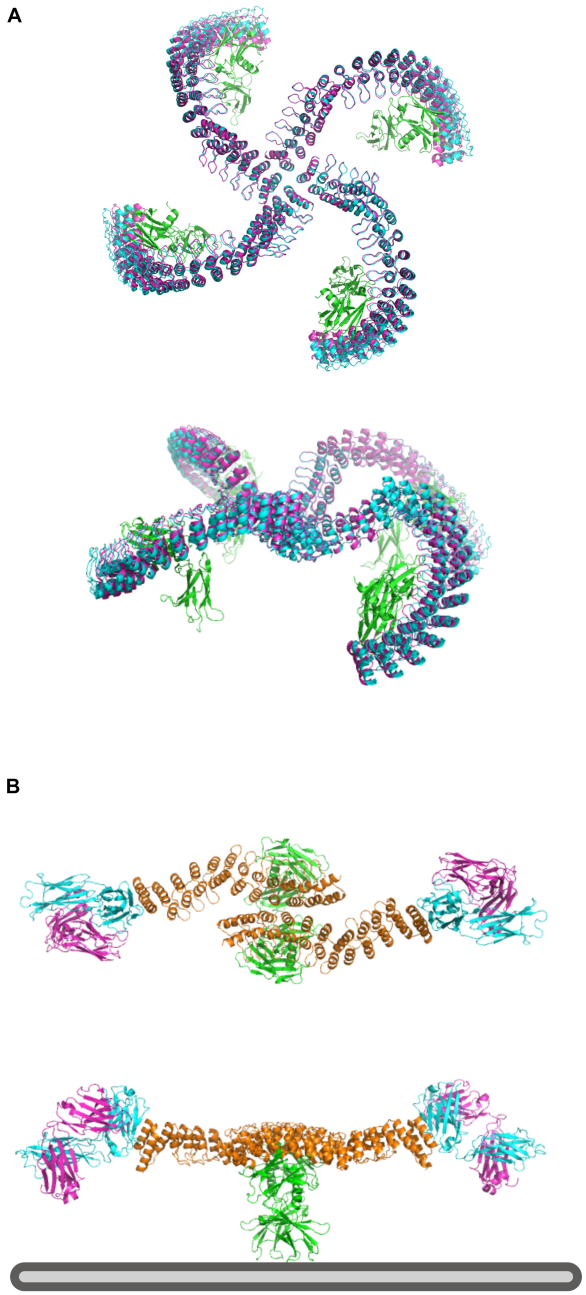


FIGURE 7 | Design of multivalent ligands. **(A)** Tetramer binder for epoR, top and side view. The receptor is in green, the elfin UI design in cyan and the repacked and energy minimized model in magenta, showing only small deviation from the coarse-grained design. **(B)** Bispecific binder, top and side view. In orange is the dimeric design, in green epoR and in cyan and magenta the Fab fragment. Each design chain binds one copy of the receptor and one Fab fragment, orienting the antibody binding site toward the plasma membrane (bottom, gray) where it could engage with a target receptor of interest.

binding interfaces in the desired positions and orientation and searching for structures that can accommodate them.

Designing With Elfin UI: Multivalent Erythropoietin Receptor Ligands

Elfin UI can be used to rapidly design rigid protein scaffolds to control the display of ligands for cell surface receptors. Dimeric ankyrin-based ligands for the erythropoietin receptor (EpoR) have been shown to induce receptor dimerization and modulate the signaling output as a function of the distance and orientation of binding sites (Mohan et al., 2019). We have used this system as a test case to assess the ability of Elfin UI to rapidly design models for alternative geometries and increased valency through manual module placement.

The first design has been generated by choosing a central tetrameric hub, extending it progressively, and ending with an ankyrin module that hosts the EpoR ligand, while avoiding clashes with the receptors (**Figure 7A**). The second model has been designed to provide multiple specificities. The scaffold contains two EpoR binding sites and two protein A domains able to bind a conserved region of a Fab antibody fragment, which can provide additional specificity for a desired cell surface target (**Figure 7B**). The designed structures are preserved after cycles of minimization and side chain repacking.

Each design with Elfin UI required about 1 h of work, including energy minimization and side chain repacking with Rosetta. In the second case, the Elfin UI design was used as a starting point for further engineering, shortening the proA module and moving the binding site to allow the placement of FAB in a position more compatible with multivalent binding. The output files are provided in the **Supplementary Materials.**

DISCUSSION

Elfin UI is a dedicated tool for coarse-grained design of custom protein architectures through building blocks combinations. Modular units are connected to form a single or multiple chains structure, depending on the modules used. The process is much faster than other backbone building methods, but it requires a highly curated database containing already all the possible pairs of modules in the correct orientation. Because of the nature of the database, interfaces between modules are already defined and further sequence design is not needed, contributing to improve the design speed, both in terms of automated solutions and feedback to users that build structures interactively. However, repacking and energy minimization are recommended to eliminate small discrepancies at the connection points between modules. External software tools (e.g. Rosetta) are required for modifications at atomic level, including repacking, energy minimization and point mutations.

Elfin UI represents a new type of interactive design software for protein design. While other tools traditionally operate directly on atomic models, Elfin UI allows the user to act at a higher level, enabling a rapid design for a desired shape which is not arbitrary, but it is connected to the information in the module database. Quality, size and fit to the design task of the database are key factors for successful designs. The precomputed database is one of the factors influencing design speed, together with the visualization of our modules, which are represented by rigid meshes, appearing in blender as full-fledged secondary structures. Moreover, all calculations (e.g., collision detection, partial overlap, distance) are performed with each module considered as a sphere with defined radius, therefore drastically reducing the computational costs.

This setup allows for rapid prototyping of potential structures of interest, exploring sequences with different lengths and shape. The option to generate custom databases allows for greater flexibility in cases where only specific types of modules could be used, e.g., peptide or protein binding domains.

Elfin UI's intuitive approach makes protein design of novel protein structures, and in particular large custom scaffolds, accessible to non-experts and to the general public, and represents a new educational and outreach tool.

Precise and reliable design of biological systems is one of the goals of synthetic biology. With Elfin, custom structures with functional domains in specific positions and orientations can be easily and rapidly designed, bringing proteins into the realm of DNA nanotechnology.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/Parmeggiani-Lab/elfin.

AUTHOR CONTRIBUTIONS

C-TY wrote the software. LO tested and optimized the software. FP devised and supervised the project and wrote the manuscript. All the authors read and commented on the manuscript.

FUNDING

The project and FP were supported by the EPSRC Early Career fellowship EP/S017542/1 and BBSRC/EPSRC BrisSynBio grant BB/L01386X/1.

ACKNOWLEDGMENTS

We would like to thank the Advanced Computing Research Centre (ACRC) and the Bristol Biodesign Institute (BBI) at the University of Bristol for support and access to the BlueCrystal and Bluegem supercomputers.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2020.568318/full#supplementary-material

REFERENCES

Adolf-Bryfogle, J., and Dunbrack, R. L. Jr. (2013). The pyrosetta toolkit: a graphical user interface for the rosetta software suite. *PLoS One* 8:e66856. doi: 10.1371/journal.pone.0066856

Andrei, R. M., Callieri, M., Zini, M. F., Loni, T., Maraziti, G., Pan, M. C., et al. (2012). Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics* 13:S16. doi: 10.1186/1471-2105-13-S4-S16

Benson, E., Mohammed, A., Gardell, J., Masich, S., Czeizler, E., Orponen, P., et al. (2015). DNA rendering of polyhedral meshes at the nanoscale. *Nature* 523, 441–444. doi: 10.1038/nature14586

Brunette, T. J., Bick, M. J., Hansen, J. M., Chow, C. M., Kollman, J. M., and Baker, D. (2020). Modular repeat protein sculpting using rigid helical junctions. *PNAS* 117, 8870–8875. doi: 10.1073/pnas.1908768117

Brunette, T. J., Parmeggiani, F., Huang, P.-S., Bhabha, G., Ekiert, D. C., Tsutakawa, S. E., et al. (2015). Exploring the repeat protein universe through computational protein design. *Nature* 528, 580–584. doi: 10.1038/nature16162

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. *Interact. Comput.* 15, 289–308. doi: 10.1016/S0953-5438(03)00010-9

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., et al. (2010). Predicting protein structures with a multiplayer online game. *Nature* 466, 756–760. doi: 10.1038/nature09304

DeLano, W. L. (2002). Pymol: An open-source molecular graphics tool. *CCP4 Newsletter on Protein Crystallography*. 40, 82–92.

Douglas, S. M., Marblestone, A. H., Teerapittayanon, S., Vazquez, A., Church, G. M., and Shih, W. M. (2009). Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Res.* 37, 5001–5006. doi: 10.1093/nar/gkp436

Durrant, J. D. (2019). BlendMol: advanced macromolecular visualization in Blender. *Bioinformatics* 35, 2323–2325. doi: 10.1093/bioinformatics/bty968

Fallas, J. A., Ueda, G., Sheffler, W., Nguyen, V., McNamara, D. E., Sankaran, B., et al. (2017). Computational design of self-assembling cyclic protein homo-oligomers. *Nat. Chem.* 9, 353–360. doi: 10.1038/nchem.2673

Gainza, P., Nisonoff, H. M., and Donald, B. R. (2016). Algorithms for protein design. *Curr. Opin. Struct. Biol.* 39, 16–26. doi: 10.1016/j.sbi.2016.03.006

Gonen, S., DiMaio, F., Gonen, T., and Baker, D. (2015). Design of ordered two-dimensional arrays mediated by noncovalent protein-protein interfaces. *Science* 348, 1365–1368. doi: 10.1126/science.aaa9897

Grochmal, A., Ferrero, E., Milanesi, L., and Tomas, S. (2013). Modulation of in-membrane receptor clustering upon binding of multivalent ligands. *J. Am. Chem. Soc.* 135, 10172–10177. doi: 10.1021/ja404428u

Hsia, Y., Bale, J. B., Gonen, S., Shi, D., Sheffler, W., Fong, K. K., et al. (2016). Design of a hyperstable 60-subunit protein icosahedron. *Nature* 535, 136–139. doi: 10.1038/nature18010

Huang, P.-S., Ban, Y.-E. A., Richter, F., Andre, I., Vernon, R., Schief, W. R., et al. (2011). RosettaRemodel: a generalized framework for flexible backbone protein design. *PLoS One* 6:e24109. doi: 10.1371/journal.pone.0024109

Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD: visual molecular dynamics. *J. Mol. Graph.* 14, 33–38. doi: 10.1016/0263-7855(96)00018-5

Jacobs, T. M., Williams, B., Williams, T., Xu, X., Eletsky, A., Federizon, J. F., et al. (2016). Design of structurally distinct proteins using strategies inspired by evolution. *Science* 352, 687–690. doi: 10.1126/science.aad8036

Johnson, G. T., Autin, L., Goodsell, D. S., Sanner, M. F., and Olson, A. J. (2011). ePMV embeds molecular modeling into professional animation software environments. *Structure* 19, 293–303. doi: 10.1016/j.str.2010.12.023

Johnson, G. T., and Hertig, S. (2014). A guide to the visual analysis and communication of biomolecular structural data. *Nat. Rev. Mol. Cell Biol.* 15, 690–698. doi: 10.1038/nrm3874

Jost, C., Schilling, J., Tamaskovic, R., Schwill, M., Honegger, A., and Plückthun, A. (2013). Structural basis for eliciting a cytotoxic effect in her2-overexpressing cancer cells via binding to the extracellular domain of HER2. *Structure* 21, 1979–1991. doi: 10.1016/j.str.2013.08.020

King, N. P., Bale, J. B., Sheffler, W., McNamara, D. E., Gonen, S., Gonen, T., et al. (2014). Accurate design of co-assembling multi-component protein nanomaterials. *Nature* 510, 103–108. doi: 10.1038/nature13404

Kleffner, R., Flatten, J., Leaver-Fay, A., Baker, D., Siegel, J. B., Khatib, F., et al. (2017). Foldit standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics* 33, 2765–2767. doi: 10.1093/bioinformatics/btx283

Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T. B., Montelione, G. T., et al. (2012). Principles for designing ideal protein structures. *Nature* 491, 222–227. doi: 10.1038/nature11600

Kramer, M. A., Wetzel, S. K., Plückthun, A., Mittl, P. R. E., and Grütter, M. G. (2010). structural determinants for improved stability of designed ankyrin repeat proteins with a redesigned C-capping module. *J. Mol. Biol.* 404, 381–391. doi: 10.1016/j.jmb.2010.09.023

Lai, Y.-T., Cascio, D., and Yeates, T. O. (2012). Structure of a 16-nm cage designed by using protein oligomers. *Science* 336, 1129–1129. doi: 10.1126/science.1219351

Lanci, C. J., MacDermaid, C. M., Kang, S., Acharya, R., North, B., Yang, X., et al. (2012). Computational design of a protein crystal. *PNAS* 109, 7304–7309. doi: 10.1073/pnas.1112595109

Leman, J. K., Weitzner, B. D., Lewis, S. M., Adolf-Bryfogle, J., Alam, N., Alford, R. F., et al. (2020). Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nat. Methods* 17, 665–680. doi: 10.1038/s41592-020-0848-2

Ljubetič, A., Lapenta, F., Gradišar, H., Drobnak, I., Aupič, J., Strmšek, Ž, et al. (2017). Design of coiled-coil protein-origami cages that self-assemble in vitro and in vivo. *Nat Biotechnol.* 35, 1094–1101. doi: 10.1038/nbt.3994

Mohan, K., Ueda, G., Kim, A. R., Jude, K. M., Fallas, J. A., Guo, Y., et al. (2019). Topological control of cytokine receptor signaling induces differential effects in hematopoiesis. *Science* 364, eaav7532. doi: 10.1126/science.aav7532

Parmeggiani, F., and Huang, P.-S. (2017). Designing repeat proteins: a modular approach to protein design. *Curr. Opin. Struct. Biol.* 45, 116–123. doi: 10.1016/j.sbi.2017.02.001

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., et al. (2004). UCSF chimera—A visualization system for exploratory research and analysis. *J. Comput. Chem.* 25, 1605–1612. doi: 10.1002/jcc.20084

Rajendiran, N., and Durrant, J. D. (2018). Pyrite: a blender plugin for visualizing molecular dynamics simulations using industry-standard rendering techniques. *J. Comput. Chem.* 39, 748–755. doi: 10.1002/jcc.25155

Schenkelberg, C. D., and Bystroff, C. (2015). InteractiveROSETTA: a graphical user interface for the PyRosetta protein modeling suite. *Bioinformatics* 31, 4023–4025. doi: 10.1093/bioinformatics/btv492

Sesterhenn, F., Yang, C., Bonet, J., Cramer, J. T., Wen, X., Wang, Y., et al. (2020). De novo protein design enables the precise induction of RSV-neutralizing antibodies. *Science* 368:eaay5051. doi: 10.1126/science.aay5051

Shaw, A., Lundin, V., Petrova, E., Fördōs, F., Benson, E., Al-Amin, A., et al. (2014). Spatial control of membrane receptor function using ligand nanocalipers. *Nat. Meth.* 11, 841–846. doi: 10.1038/nmeth.3025

Veneziano, R., Ratanalert, S., Zhang, K., Zhang, F., Yan, H., Chiu, W., et al. (2016). Designer nanoscale DNA assemblies programmed from the top down. *Science* 352, 1534–1534. doi: 10.1126/science.aaf4388

Williams, S., Lund, K., Lin, C., Wonka, P., Lindsay, S., and Yan, H. (2009). "Tiamat: a three-dimensional editing tool for complex DNA structures," in *DNA Computing Lecture Notes in Computer Science*, eds A. Goel, F. C. Simmel, and P. Sosík (Berlin: Springer), 90–101. doi: 10.1007/978-3-642-03076-5_8

Wood, C. W., Heal, J. W., Thomson, A. R., Bartlett, G. J., Ibarra, A. Á, Brady, R. L., et al. (2017). ISAMBARD: an open-source computational environment for biomolecular analysis, modelling and design. *Bioinformatics* 33, 3043–3050. doi: 10.1093/bioinformatics/btx352

Wu, Y., Batyuk, A., Honegger, A., Brandl, F., Mittl, P. R. E., and Plückthun, A. (2017). Rigidly connected multispecific artificial binders

with adjustable geometries. *Sci. Rep.* 7:11217. doi: 10.1038/s41598-017-11472-x

Yeh, C.-T., Brunette, T., Baker, D., McIntosh-Smith, S., and Parmeggiani, F. (2018). Elfin: an algorithm for the computational design of custom three-dimensional structures from modular repeat protein building blocks. *J. Struct. Biol.* 201, 100–107. doi: 10.1016/j.jsb.2017.09.001

Youn, S.-J., Kwon, N.-Y., Lee, J. H., Kim, J. H., Choi, J., Lee, H., et al. (2017). Construction of novel repeat proteins with rigid and predictable structures using a shared helix method. *Sci. Rep.* 7:2595. doi: 10.1038/s41598-017-02803-z

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership