Opportunities and Challenges in Applying Artificial Intelligence to Bioengineering

Fusun Yaman, Aaron Adler, and Jacob Beal

Abstract Our capability to engineer biological systems is increasing rapidly in both speed and scale, leading to explosive growth in the complexity of bioengineering projects that can be contemplated. Artificial intelligence techniques have helped to tame such complexity in many other fields, and are beginning to be employed in the same way to the engineering of biological organisms. Using these techniques, computers represent, acquire, and employ domain knowledge to automate "more routine" processes and allow humans to focus on deeper scientific and engineering issues. At the same time, applying more sophisticated techniques such as these imposes new demands on biological systems experimentation, particularly with respect to representation and curation of data. This chapter surveys the state of the art in applying artificial intelligence to bioengineering, as well as discussing opportunities and challenges for the future.

1 Introduction

Synthetic biology is the systematic design and engineering of biological systems, principally through modification of their genetic code. Engineering living organisms in this manner holds clear potential for enabling revolutionary advances across many different fields of application, including preventive, diagnostic, and therapeu-

Fusun Yaman

Aaron Adler

Raytheon BBN Technologies, USA, e-mail: aaronadler@alum.mit.edu

Jacob Beal

Raytheon BBN Technologies, USA, e-mail: fusun.yaman@raytheon.com

Raytheon BBN Technologies, USA, e-mail: jakebeal@ieee.org

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.

tic medicine; energy production and storage; chemical and materials manufacturing; agriculture; and environmental management and remediation.

In recent years, our ability to engineer biological systems has been increasing rapidly, both in the types of modifications of living organisms that can be routinely contemplated and in the application of automation, systematization, and standardization to decrease costs and increase the pace at which modifications can be performed. For example, where once there were only a few effective genetic regulatory devices, now there are growing libraries of high-performance devices based on a diverse set of biological mechanisms in a number of different organisms, including TALE proteins [48], CRISPR [40, 31], recombinase [12, 76], and TetR homologs [68]. At the same time, new combinatorial and high-throughput protocols are beginning to enable efficient fabrication and screening of vast numbers of genetic constructs (e.g., [61, 66]) and engineering at the scale of entire genomes (e.g., [57, 60]).

This ongoing growth of capabilities, pace, and scale likewise increases the amount of information and knowledge that must be marshaled, across multiple disciplines, in order for the full potential of these capabilities to be realized in effective organism engineering workflows. Moreover, the ongoing rapid pace of advancement in the field means that it is likewise critical for such workflows to be flexible and capable of rapidly integrating new knowledge, protocols, and capabilities. Similarly, since no single person or laboratory can contain all of the different types of crossdisciplinary expertise that are becoming necessary, workflows need to be able to minimize friction in the transfer of knowledge, specifications, and materials both within an organization and between cooperating organizations. In all of these ways, we see the increasing importance of identifying techniques for managing the complexity of biological engineering.

Similar challenges of managing engineering complexity have been faced in other fields as well, such as electronics (e.g., microprocessors with billions of transistors), software engineering (e.g., operating systems with millions of lines of code), and mechanical engineering (e.g., commercial aircraft with millions of components). In these and other disparate areas, the response has been to use artificial intelligence (AI) techniques to capture the knowledge of human experts, embedding this knowledge into assistive tools that can carry out routine work and error checking (e.g., microchip transistor layout, optimization, and validation), and into standard interfaces that allow such tools to be readily connected into customizable engineering workflows (e.g., VLSI for high-level specification of an integrated circuit, logical formulae for validation and testing, and geometry files for the final layout to be fabricated). Together, these allow human engineers to operate at a higher level of abstraction, focusing more on core issues of intention and design, and also allowing problems to be more readily factored across teams and organizations.

Already, the profusion of data from high-throughput protocols and assays has been leading practitioners to turn toward AI approaches, particularly methods from the machine learning and "big data" communities. AI, however, offers a much richer landscape of potentially applicable techniques in addition to these, including knowledge representation (e.g., semantic networks, frame representations), knowledge acquisition (e.g., hypothesis generation), planning and decision making (e.g., expert systems, constraint-based reasoning, planning under uncertainty), and automated action (e.g., robotics). In general, these approaches involve encoding a much greater degree of domain-specific knowledge. Such knowledge-rich approaches can complement and enhance data-intensive machine learning methods, as well as addressing other classes of problems that machine learning and big data methods do not.

Another important lesson from prior domains that must also be applied to the automation of synthetic biology, however, is that no amount of sophistication in methods can obviate the foundational requirements of metrology and curation. In order for workflows to make effective use of data, models, and specifications produced by different organizations, they must be expressed in compatible data structures and with comparable units. The better the comparability and curation of such inputs, the more that our tools can be put to use engineering novel capabilities, rather than in attempting to reconstruct missing or degraded information.

Biological mechanisms may arguably be less well understood and more complex than the building blocks of mechanical engineering or electronics, but there are still many opportunities for AI techniques to be applied. Furthermore, as in other fields, the most transformative impacts of AI are likely to be cumulative, through the incrmental effects of many AI-based tools, each addressing different bottlenecks or points of friction.

Accordingly, in this chapter (an extended version of [8]) we present an analysis of opportunities and challenges relating to the application of AI techniques to the engineering of biological organisms. We begin with a review of a typical organism engineering workflow in Section 2, and discuss opportunities for AI-based tools to help address challenges in such a workflow in Section 3. We then discuss representations to support curation and integration of tools into workflows in Section 4. Finally, Section 5 identifies key challenges for the future and Section 6 concludes.

2 Organism Engineering

There are many different complex problems that must be addressed in various forms and applications of organism engineering. Rather than attempting to cover the whole breadth, we will narrow the focus of discussion for this chapter to one important and widely addressed class of synthetic biology systems—genetic regulatory networks that implement a sense-control-actuate paradigm—and to a prototypical designbuild-test workflow for engineering such systems.

In particular, the sense-control-actuate paradigm means any synthetic biology system that can be mapped onto the three stages illustrated in Figure 1:

• **Sense** refers to the transduction of properties of the environment or cell state, such as small molecule concentration, light, or nutrient stress, into informational signals (typically represented by transcriptional or translational activity).

Fusun Yaman, Aaron Adler, and Jacob Beal



Fig. 1 Many synthetic biology systems can be viewed as composed of three components: sensing of environmental or cell state, a control system that processes the signals from these sensors to determine appropriate cellular behavior, and actuators that convert the control signals into actions such as enzymatic pathway regulation or reporter expression.

- **Control** refers to the processing of signals from the sensors to determine appropriate behaviors in response from the cell, and may also include cell-to-cell communication as part of this processing.
- Actuate refers to the transduction of the computed control signals into actions on the cell and its environment, such as regulation of enzymatic pathways or expression of a reporter protein.

A great many synthetic biology applications can be readily mapped onto this paradigm, particularly many therapeutic and diagnostic applications, as well as environmental sensing, and even the regulatory aspects of chemical or material fabrication.

As we shall see, there are a wide range of ways in which AI techniques might be applied to aid in engineering such genetic regulatory networks. Moreover, many of these applications would likely apply similarly to other classes of synthetic biology systems as well, further indicating the breadth of potential in the combination of AI and synthetic biology.

2.1 Typical Workflow

For our discussion in this chapter, we shall consider one of the frequently used synthetic biology workflows for organism engineering, which may be viewed as comprising three steps: design, build, and test (Figure 2).

This workflow is often invoked for engineering any type of synthetic biology system, not just sense-control-actuate systems, but the content of the steps can be different for other classes of system (e.g., design of a novel heat-tolerant biosynthesis process might largely neglect selection and arrangement of components and instead focus primarily on molecular dynamics simulations of one key enzyme). We discuss each of these steps in turn, emphasizing opportunities for improvements in the typical current workflow. Opportunities and Challenges in Applying Artificial Intelligence to Bioengineering



Fig. 2 A typical synthetic biology workflow for organism engineering may be viewed as a cycle of three stages: *design* maps a behavior specification to a nucleic acid sequence intended to realize this behavior, *build* draws on synthesis and/or assembly protocols to fabricate said nucleic acid sequence, and *test* assays the behavior of cells modified to include the sequence, feeding back this information into the design process to complete the cycle.

2.1.1 Design

Design encompasses a number of different interacting aspects of engineering encountered along the path from an abstract specification of desired organism behavior to one or more nucleic acid sequences intended to implement the specified behavior. At the most abstract level, the engineer must determine the arrangement of sensors, actuators, regulatory relationships, and/or enzymatic pathways that will be used to implement the desired behavior. Such an arrangement must then be mapped onto the set of DNA or RNA components that are actually available, or new components must be engineered to fit the needed specifications for those particular components or interactions. It is further necessary to ensure that there will not be conflicts between the components selected, either directly (e.g., by gene products with undesired interactions with other elements in the system), or indirectly (e.g., by collectively over-straining cellular resources). Finally, the resultant networks must be linearized onto one or more nucleic acid sequences (linear referring to the single dimension of the nucleic acid backbone, on which the elements of the network must be placed). These sequences must also be chosen so that they can be synthesized or assembled with the resources available to the engineer and also be delivered to operate inside the cell line that is being engineered. At present, the selection and arrangement of components is typically carried out largely by hand, with little usable characterization data to guide component selection and poor models to quantitatively predict the behavior of the resultant composite system; component engineering features some

more principled approaches (e.g., [74, 27, 41]), but is still generally a rather slow, costly, and hard to predict process.

2.1.2 Build

The build stage creates organisms modified with the designed nucleic sequence(s). First, the sequence or sequences are synthesized or assembled (e.g., via Bio-Bricks [15, 44], Gateway-Gibson [34, 32], or MoClo [75]) to produce actual physical samples, and the host organisms are cultured to be ready to receive these sequences. The sequences are then delivered to the organism by one of a variety of protocols, to either operate autonomously or to be integrated into the cell's DNA, depending on the protocols involved. Both of these stages have a number of issues in yield and quality assurance, particularly as many protocols seem to require a "magic touch" by which some practitioners get reliable results and others frequently build systems with problematic flaws. Next-generation sequencing may help to address issues of quality control, but planning, resourcing, and executing build protocols effectively is still an open and challenging problem.

2.1.3 Test

Finally, the behavior of the newly constructed organism or organisms is assayed to determine how well it corresponds with the original specification, and to help debug misbehavior such that the next iteration of the design can be closer to the desired behavior. Typically this involves culturing the organism under specific conditions (though it can also involve delivering the organism for *in vivo* testing), and processing it through assay instruments at one or more time points to obtain phenotypic information. Here, one of the biggest challenges is in relating assay data to the original specification: many assays produce data in great volume, but the mapping back to the original specification is often qualitative or relative, rather than absolute. Likewise, it is often not clear how to relate the observed behavior to predictive models that can provide principled guidance in how to adjust the design phase in order to produce improved results.

2.2 Layers of Organism Engineering Interactions

Beyond the individual workflow discussed above, it is important to note that organism engineering rarely takes place in isolation. The complexity of managing organism engineering is typically complicated by interactions on several levels (Figure 3).

First, note that the previous subsection focused only on single steps in the engineering of an organism intended to meet a particular specification. These are considered together as a cycle, however, because in current practice, engineering an



Fig. 3 Organism engineering does not take place in isolation: in addition to potentially many design-build-test cycles of a single project, there are often complex interactions both within a laboratory and between laboratories, in which many different workflows interact.

organism to meet a specification typically requires many iterations of this designbuild-test cycle. Across these iterations, an organism engineer needs to be able to track and integrate results, to accelerate the process by carrying out some cycles in overlapping stages rather than waiting for each to finish completely (particularly when lengthy protocols are involved), and more generally to optimize the execution of the workflow with respect to practical resource constraints.

Furthermore, within any given laboratory, there are often many organism engineering projects ongoing in parallel. These may introduce friction in a project, as other projects compete for resources or to schedule time on shared equipment. They can also be beneficial, however, through the sharing of knowledge, techniques, and protocols. Furthermore, complex projects may have many people working simultaneously on different aspects of an organism (which must remain compatible), or may have some of the work involved carried out by specialists.

Finally, different laboratories and organizations often need (or could benefit from) various forms of interaction. Examples include exchange of information about assay results, exchange of materials (allowing some organizations to specialize as high-efficiency suppliers), exchange of protocols and methods, and outsourcing of quality assurance testing.

Together, all of these layers of interaction form a much more complex "ecosystem" of organism engineering, meaning that there are many more opportunities beyond those suggested by considering an individual design-build-test workflow, where AI-based improvements may greatly accelerate the overall development of new organism capabilities by means of reducing friction from cross-workflow interactions and by improving the exchange of knowledge between individuals and organizations.

3 Potential AI Contributions

Currently most organism engineering workflows have little automation and rely heavily on domain expertise, a limited amount of which is shared through publications. There are a number of places, however, where tools to support or carry out information integration and informed decision making might improve the efficiency and speed of organism engineering, as well as enabling better results to be produced. Such integration and decision-making points (summarized in Table 1) are good opportunities where the application of AI techniques might prove valuable for the practice of synthetic biology. Accordingly, in this section, we consider the application of AI techniques to each of the aspects reviewed in the previous section in turn.

Engineering Challenge	Key AI techniques	Examples
Machine-assisted gene circuit design	expert systems, constraint-based reasoning, heuris-	[3, 80, 20, 58, 11, 52, 4, 5]
	tic search, optimization, machine learning, multi-	
	agent systems	
Flexible protocol automation	robotics, planning under uncertainty	[24, 35, 6, 49, 69, 72]
Assay interpretation and modeling	machine learning, qualitative reasoning	[42]
Lab management and optimization	heuristic search, optimization, planning under un-	[16, 17, 72, 24]
	certainty	
Represent/exchange designs	semantic networks, ontologies	[30, 29, 25]
Represent/exchange protocols	semantic networks, schemas	[49, 69]

 Table 1
 Summary of bioengineering challenges for which there is a high potential for AI techniques to contribute to the solution.

3.1 Design: Machine-Assisted Engineering of Control Circuits

Design is a clearly knowledge-dependent portion of the workflow, and so it is unremarkable that AI techniques should be applicable to this phase in a number of different ways. Indeed, AI techniques are already key to a number of specialized sub-tasks within the design process: for example, in genome mining (e.g., [68]) a wide array of machine learning methods for clustering, analysis, and inference are often used to determine the significance of sequence elements and relations between sequences, and in protein design (e.g., [63]) heuristic search techniques are frequently used to explore the staggeringly large number of candidate sequences that might satisfy a design goal, guided by heuristic rules that capture the intuitions of human experts with regards to protein structure and behavior, such as the tendency of proteins to be more stable when they have a hydrophobic core.

Turning to the integrative design challenges that are the focus of this chapter, there are many opportunities there as well, of which we shall discuss several examples where the application of AI techniques is particularly clear and prominent. One example is the application of knowledge-based approaches to the design of genetic regulatory networks directly from specifications of behavior. One approach,



Fig. 4 Biocompiler stiches motifs that compose the high level program, producing an abstract genetic regulatory network (AGRN). Figure adapted from [5].

used by the Proto BioCompiler [5], is a motif-based technique to design and optimize genetic regulatory network topologies from behavior specifications expressed in a high-level programming language (Figure 4). One drawback of copy-paste style programming is a design that is not optimized. This potentially has a high impact on the viability of the biological circuits as it presents more points of failure (for example due to mutation) and puts an unnecessary stress on the host organism, wasting its resources. Interestingly, as shown in BioCompiler, many software compiler optimization techniques, such as copy propagation and dead code elimination, are applicable to simplifying an abstract genetic regulatory network (AGRN). Figure 5 demonstrates how the AGRN in Figure 4 can be automatically simplified (it is worth noting that research in compilers and code optimization has its roots in AI "automatic programming," compilers having later on evolved into a specialized field of their own).

Abstract genetic regulatory networks can then be mapped to a fully instantiated genetic regulatory network by the MatchMaker [80] constraint-based reasoning system, forming a complete design ready to be built. Figure 6 outlines the two main constraint satisfaction problems Matchmaker solves.

Motif-based stitching is also utilized in Cello [56], a descendant tool combining BioCompiler and MatchMaker functionality with an existing high-level language. Cello uses Verilog as its description language (Verilog was developed for electronic design automation). The users of Cello specify a circuit in Verilog along with the sensors, actuators, and a "user constraints file" that defines the organism, operating conditions, and gate technology. The Cello software then designs the circuits using a gate database like MatchMaker, and can simulate performance. The designs Cello produced were tested and found to produce correct functionality in 45 of 60 circuits and for 92% of internal output states. A number of other design tools have also been developed, approaching the design synthesis problem from a variety of perspec-



Fig. 5 Genetic regulatory networks can be optimized using code optimization techniques adapted from software compilers. Figure adapted from [5].

tives and all enabled by various forms of knowledge-representation and reasoning (e.g., [20, 58, 11, 52]).

Another place where AI techniques are likely to be useful is in the identification of biomarkers for sensing targets. Machine learning techniques have already been widely applied in systems biology for a variety of applications, such as identifying relevant biomarkers. Just so, these same techniques may be applied to identifying the best sensors to incorporate into a synthetic biology circuit. For example, the cancer detection circuit in [79] uses a set of six miRNA markers identified heuristically by hand from a large number of potential candidates. Machine learning is ideally suited for automating such target identification, and can identify such sets of markers more quickly, more reliably, and likely with better results as well, as demonstrated with the information-based method presented in [4]. These same techniques are likely to be useful in a wide range of other similar applications as well, in all of which machine learning methods applied to design are likely to be valuable in improving the speed and quality of selection of target sets.

A third opportunity lies in the application of multi-agent systems methods to the engineering of cell populations. Here, each cell can be viewed as an "agent" (a living one) and a collection of cells, such as a colony, biofilm, or tissue, can then be viewed as a multi-agent system. Much work from the multi-agent community could

Opportunities and Challenges in Applying Artificial Intelligence to Bioengineering



Fig. 6 Matchmaker maps abstract parts to concrete parts, making sure the logical and signal properties of the parts match.

potentially be applied to cells within this framework, though the methods will likely need to be adapted for the slower diffusion and reaction times typical of cell-to-cell communication, as well as the very limited number of distinguishable signals currently readily accessible. For example, the coordinated repressilator developed by Elowitz [26] is likely to be susceptible to analysis and tuning with such methods. The spatial nature of cells also lends itself to spatial computing, as explored in [3], which describes how a high-level spatial computing language may be a good method for automated design of multicellular systems, such as the Weiss laboratory's handdesigned band detector [2]. Figure 7 demonstrates the alignment between the designs, the spatial-computing vs. hand-designed. Furthermore, as synthetic biology systems become more complex, it will be possible to consider coordinating multiple different sensors and actuators in differentiated roles across systems comprising large numbers of cells, coordinating to accomplish a task. It is at just such systems that work from the multi-agent systems community is squarely targeted, particularly the study of collective and emergent behavior, as observed in [67].

Finally, a number of design tools focus on the engineering of individual parts using physical models to reason directly in terms of nucleic acid or amino acid sequences. For example, ribosome interactions with mRNA are controlled by the partial unfolding of RNA structures, the amount of single-stranded surface area, the absence of cooperative binding, and the potential for ribosomal sliding. A biophysical model using first principles of thermodynamics and a four-parameter free energy model can accurately predict ribosomal translation rates. These calculations have been built into a tool that uses the model to calculate the RBS translation rate for a specified RNA sequence [13, 64]. Likewise, in order to understand the biological properties of protein molecules, it is critical to understand their structure. Computational methods have been developed to obtain such information. The ROSETTA



Fig. 7 Cells can be seen as spatial computers that execute the same program based on the local input they receive from their neighbors. Figure adapted from [3].

method for *ab initio* protein structure prediction uses a Monte Carlo procedure with an energy function, clustering, heuristics, models, and databases [65]. Both of these share a property typical of physical models: large numbers of parameters implying a massive and complex search space, along with many design features and significant uncertainty as to which are good predictors of ultimate behavior. AI methods for search and machine learning have been engineered for addressing such challenges in other spaces, and may prove applicable in this area as well.

3.2 Build: Flexible Protocol Automation

The use of robotics in synthetic biology is appealing because many protocols are repetitive, error-prone, and time-sensitive. Automation requires more precise descriptions of laboratory protocols than are typically reported at present, but in exchange promises to efficiently provide reliable and reproducible results. A number of efforts have already demonstrated the potential value of automation in the assembly of DNA sequences from standardized biological parts (e.g., [24, 35, 6]). For example the Assembly Planner [24] algorithm utilizes three different heuristics to find an assembly sequence that minimizes assembly steps while maximizing the sharing of intermediate products (Figure 8).

Much current laboratory automation, however, is designed to run a fixed procedure many times (e.g., analyze many samples using a standard procedure). AI robotic, planning, and reasoning methods offer the potential to make such automa-



Fig. 8 Opitmizing binary process of assembling BioBricks by reusing intermediate products.

tion less expensive to employ by increasing the flexibility with which protocols can be applied and allowing them to be specified in a more lightweight manner and at a much coarser granularity. Advanced robotic methods can increase the range of automatable protocols and likelihood of protocol success, while planning and reasoning methods allow specification of protocols in terms of goals and requirements (allowing the automation to fill in the details), rather than an exhaustive specification of steps to be carried out.

Planning under uncertainty is likely to be highly useful in this area as well. Protocols for assembly of nucleic acid sequences, culturing of cells, and transformation/transfection all contain several complex actions than depend on each other causally, and have temporal and resource constraints as well as actions that can be carried on concurrently. Many of these actions also have non-deterministic outcomes, either inherent to the protocol or due to the many vicissitudes of laboratory execution. Furthermore, there are often several alternative ways to achieve each step, some of which are already standard protocols, while some are cultural to particular laboratories. Although difficult, planning and optimization of such processes are well understood problems with many available methods to address them in the AI community.

A different planning problem arises for automatic assembly robots. These robots have limited real estate in terms of placing plates of cells and allocation of reaction wells. While a human executing an assembly protocol is often comfortable (rightly or wrongly) with ad-hoc allocation, a robot has to plan where each sub-product is going to be put, and augment the protocol steps with actions to place and pick up the right samples. Planning for robotic assembly produces scripts that then be fed into the robot for unsupervised execution.

Systems targeted at such automation include [72, 43, 49, 69, 70]. For example, Puppeteer [72] consists of two components, a planner and a compiler, for automating the process of DNA assembly. It is a web-based tool that generates assembly instructions and can track, manage, and control laboratory tasks, reagents, and equipment. Puppeteer operates in two stages, as shown in Figure 9, first planning for assembly steps, then executing resource allocation, such as assigning plates and wells to intermediate products. This system generates both human- and machine-readable instructions that optimize the use of capacity-limited resources, using a intermediate language called the Common Human Robot Instruction Set (CHRIS). The compiler



Fig. 9 Puppeteer plans for assembly steps while simultaneously solving the resource allocation and referencing problem.

in Puppeteer can then compile CHRIS into low-level instructions for either a human or a laboratory robot.

Another notable example is Aquarium [43], which uses strongly typed operations that can be composed into workflows that can be specified using the Aquarium Workflow Language (AWL). Aquarium workflows are executed by lab technicians, essentially allowing researchers to specify experiments to run, have lab technicians precisely execute the protocols, and then return results back to the researcher. While the automation is performed by humans not robots, the planning challenge is much the same and the procedures are repeatable. Aquarium also performs backtracking that can produce a plan that achieves a specified goal.

Other languages are more directly automation focused, such as Autoprotocol [70], a formal language for specifying experimental protocols that is intended to address both automation challenges and reproducibility issues in biological research. It requires specifying all the parameters necessary for an operation to remove ambiguity. Autoprotocol is less of a planning system or language, however, and more a knowledge representation for encoding protocols. At a higher level, Antha [69] is a programming language to provide biologists a flexible interface to lab automation and analytics. The goal is to abstract away the details of, for example, particular liquid handling robots, automating experiments and freeing up the skilled experts to provide insights and use their time more productively. Antha leverages machine learning techniques to understand complex systems, including the use of active learning techniques. As can be seen, a variety of different AI techniques can be applied to address different aspects of the automation challenge.

3.3 Test: Assay Interpretation and Modeling

The final stage of the core workflow, testing, poses clear parallels to systems biology and bioinformatics, where AI techniques, particularly machine learning, have already proven themselves quite useful. Here the challenges are more focused on interpretation of data, particularly in the areas of:

- disentangling the multifarious different aspects of stochasticity in delivery, cellular systems, and observation from one another,
- coping with the potentially massive volumes of data that can be produced by high-throughput assays, and
- integration of many different results from qualitatively different experiments and assays.

A wide variety of AI techniques, including machine learning, model construction, qualitative reasoning, and automated hypothesis generation, are likely to be applicable here, and to aid in the feedback from assay results to model adjustments to the next iteration of design.

For example, the Empirical Quantitative Incremental Prediction (EQuIP) [21] method accurately predicts genetic regulatory network behavior from detailed characterizations of individual genetic components. EQuIP utilizes learned models to predict the performance of the composite circuit. The stages of EQuIP from data gathering to predictions are as follows: 1) Calibrated experimental observation of the behavior of regulatory and constitutive elements in cells (top-left Figure 10). 2) Data, factoring in circuit copy number, is used to build rate functions for time-dependent regulated production and for loss of protein concentration, which can be mathematically integrated for computational simulation (bottom-left Figure 10). 3) The behavior of a biological circuit is predicted by linking production functions for each regulatory relation and loss functions for each relevant protein, according to circuit topology, then simulating concentrations over time according to the network of rate functions (right Figure 10).



Fig. 10 Predictive engineering of a cascade using EQuIP. Figure adapted from [21]

Learning can also be taken yet further into active design of experiments: one extreme prototype eliminates humans entirely [42], but pragmatically the impact is more likely to come from assistive interfaces where the human and machine work together to apply the interpretation of test results.

3.4 Laboratory Management and Optimization

Tight resource constraints in shared laboratory space and equipment is another source of difficult conflicts, and where effective solutions can be explored as an application of AI techniques for heuristic search, optimization, and planning under uncertainty. For example, before even starting an experiment, a graduate student may have to schedule time on high-demand assay instruments such as flow cytometers, so that the instrument will be available when the samples need to be evaluated. This involves guessing times for build and test protocols, and often results in inefficient conservative scheduling of longer instrument times than necessary. Reagents and other materials also often have considerable costs, and must be managed carefully and ordered at appropriate times, particularly given the propensity of some to degrade or their requirements for special storage environments with limited availability. Combined with automation of the build stage of the workflow, this may also allow scheduling of protocols such that shared laboratory equipment would be optimally used, as well as eliminating late-night operation of equipment by sleepdeprived humans. A number of existing laboratory information management systems (LIMS) products already attempt to support this, but typically provide only shallow automation and require a high degree of micromanagement by their users; next-generation systems such as Organick [16], Diva [17], and Puppeteer [72] make more explicit use of AI planning and reasoning techniques to enable a higher degree of automation, but are still just scratching the surface of what is possible.

Finally, further optimization may be possible if experiments are jointly planned, such that they can benefit from sharing complete or intermediate build stage products. For example, if two projects are building DNA sequences that contain a shared sub-construct, coordinating the build process can ensure that it is produced only once, then used in the production of both final products, as has been demonstrated in [24].

4 Knowledge Representation, Integration, and Workflows

While AI techniques are likely to lead to significant improvements from individual tools focused on specific sub-problems in organism engineering, the history of AI-assisted automation in other engineering areas suggests that the largest impact is likely to come from workflow-based approaches that integrate many small improvements across multiple different tools, both for individual practitioners and including the exchange of information between different practitioners and organizations.

For example, in both electromechanical systems engineering and integrated circuit design, standard formats for the exchange of design specifications help to support a complex ecosystem of computer aided design (CAD) tools, vendor-supplied libraries providing packaged subsystems, tools for analysis, simulation, and optimization specialized for particular physics or application domains, and the factoring and outsourcing of manufacturing and production across many organizations and facilities. Similarly, software engineering has also been radically transformed over the past half century by the development of a highly integrated and diverse network of automation tools, including high-level languages and compilers, library linking, optimization methods, cross-system compilation and maintenance, dependency management, automation-assisted code integration, testing automation, and assistive development environments. AI techniques for knowledge representation have been a critical enabling technology in these other domains, and are likely to be for addressing these challenges in synthetic biology as well.

Knowledge representation focuses generally on the organization of information about the world into a form suitable for machine reasoning. Commonly used formalisms for knowledge representation include logical predicates, semantic networks, frame representations, rules, and ontologies. Used within a tool, these can form the structures over which the tool operates using either domain-general knowledge (e.g., rules of logical inference) or domain-specific knowledge (e.g., DNA sequence design heuristics). Knowledge representations also provide a framework for eliciting information from human engineers, organizing presentation of information back to humans, and for exchange of information between tools. Critically, knowledge representations often explicitly offer support for representing uncertainty, unresolved decisions, and contradictions, allowing tools to represent not only complete and correct information but also partially resolved problems in need of attention and assistance from humans or other tools.

With regards to this latter case, knowledge representation is often confused with two related classes of artifact, file formats and application programming interfaces (APIs). These are best viewed, however, as artifacts that help to realize a knowledge representation. A file format is just a particular way of serializing a knowledge representation, and an API just a particular collection of operators for manipulating a knowledge representation. Viewed from a knowledge representation perspective, there can potentially be many such artifacts all supporting the same representation. Furthermore, the knowledge representation community has developed generalized file formats such as JSON and RDF, along with accompanying software libraries presenting generalized APIs, which can be used to support a wide range of information and interactions with little or no customization. This, in turn, makes it much easier to incrementally develop and improve representations, tools, and workflows. Such methods are already extremely widespread in business-to-business integration in the commercial world, and the generalized technologies developed there are likely to apply just as well to synthetic biology as they do to the wide diversity of other tasks where they are already applied.

The synthetic biology community has already been working on developing standards for representation and interchange in a number of areas. One, described above in Section 3.2, is the specification of protocols, which supports not just automation but also interchange between organizations. More recently, some work in this area has in fact begun to focus explicitly on cross-organization integration and planning of experiments [14]. Other important areas of focus include ontologies, design specifications, composable models, and integration across workflows. The biology community has long made use of ontologies, which allow systematic cataloging and reasoning over the relationships of biological entities. Some of these ontologies are more focused on taxonomic organization of classes of entities and relationships, such as the sequence ontology [25], which provides a taxonomy of nucleic acid constructs such as promoters and coding sequences, or BioPAX [23], which provides a language for description of biochemical pathways. Others are more focused on producing a systematic catalog of objects of interest such as small molecules [22], proteins [71], or pathways [38]. The power of such ontologies are further enhanced when they are organized and linked together, as in the EDAM meta-ontology [37] and the Ontobee linked data servers [78].



Fig. 11 The Synthetic Biology Open Language (SBOL) [30, 62] represents both structure and function of biological designs, as shown in this example of a system comprising two modules (dashed lines): in the left module, aTc de-represses the pTet promoter by repressing the TetR protein, which regulates the GFP-producing right module (image presented using SBOL Visual diagram language [18]).

Building upon these, the Synthetic Biology Open Language (SBOL) [30, 29, 62, 19] provides a means of describing biological designs in terms of both their structure (e.g., nucleic acids or protein sequence) and their function (e.g., genetic production and regulatory interactions). This standard has been developed by an open community comprising researchers from both "wet" and "dry" disciplines, based on the AI techniques of semantic network representations and ontology construction. Using these techniques allows the standard to unambiguously define and identify elements of a biological design, as well as to support systematic integration of components into complex systems and combinatorial libraries. Using semantic networks also allows users and tools to create custom extensions of the standard that are still compatible with tools that are unaware of those extensions, thus providing a smooth path for improvement of representations over time. Presentation of knowledge is also an important ingredient of knowledge representation, and the SBOL community has accordingly also produced a complementary SBOL Visual standard for visual representation (Figure 11) of designs [59, 18], which both extends and is compatible with the prior Systems Biology Graphical Notation [46]. Complementary to these (and able to be linked to them) are modeling representations such as the Systems Biology Markup Language (SBML) [28], which is designed to represent biological

reactions, processes, and pathways in a composable framework, and SED-ML [73], which is designed for reproducibly recording biological simulations.



Fig. 12 SBOL 2.2 [19] supports integration of design, build, test, and model information through all different stages of organism engineering workflows (image from [1]).

More recently, work on representations has also begun to address the challenge of integrating across all of these different stages of engineering workflows. The recently completed SBOL 2.2 standard [19] addresses this with the aid of the PROV-O provenance ontology [47], which supports encoding of the history and derivation of information, as well as the agents and activities acting to create that information. Through this, SBOL 2.2 supports integration of information through all different stages of organism engineering: design information is linked to the physical samples that instantiate those designs, which are in turn linked to the experimental data collected from those samples, and onward to models derived from that information and to new designs, all the way around the design-build-test loop (Figure 12). Along the way, information about tools and protocols (e.g., as discussed in Section 3.2) can be attached through activity representations, potentially linking all of the metadata associated with an engineering effort.

Curation systems can then leverage such representations to provide a single integration point for both users and software tools, thereby supporting flexible engineering workflows and reducing the friction between researcher, software tools, and community. One such curation tool is the Joint BioEnergy Institute Inventory of Composable Elements (JBEI-ICE) [33], an open source registry platform for managing information about biological parts. It can store information about plasmids and DNA parts in many formats, and provides both a web-based interface and an API. The API allows JBEI-ICE instances to be connected, and enables connections to other tools. JBEI-ICE enables advanced searches as well as connections to other tools such as BLAST, and allows export/import of sequence data in different formats, e.g., GenBank or SBOL. Its complement, also developed by JBEI, is the Experimental Data Depot (EDD) [54], which aims to provide similar facilities for experimental data. Another curation tool, this one providing an all-in-one interface to both designs and experimental data, is SynBioHub [53], an open-source design repository built on the SBOL Stack functionality [51]. SynBioHub also provides both a web-based interface and an API for searching and sharing designs, translating between data formats, and federating across different repositories including the iGEM repository, JBEI-ICE, and instances of SynBioHub.

5 Challenges

So far, our discussion has focused primarily on the potential opportunities and benefits in applying AI techniques to benefit synthetic biology organism engineering. In all of the areas discussed, however, work is at a relatively early stage of development, and realizing the anticipated benefits requires much additional work. In particular, a number of key challenges exist that are likely to be encountered in pursuit of these applications, and which must be addressed in order to fully realize the potential from the synthesis of these two fields.

5.1 Curation and Comparable Data

AI methods, for all their power, are strongly limited by the quality of the information with which they are supplied. In order to be available for AI methods to be applied to, the artifacts of engineering, such as designs, protocols, and experimental data, must be tracked and organized in a manner susceptible to machine interpretation, rather than being kept in lab notebooks or ad hoc files and formats. Likewise, many forms of experimental data are currently most often taken in arbitrary or relative units, which cannot be directly compared between laboratories or even between experiments within a single laboratory. Even powerful modern machine learning and inference methods are no panacea for these basic challenges of curation and metrology: while they may be able to help "clean" poorly calibrated or organized data to some degree, doing so injects additional degrees of freedom that decrease the amount that can be learned or inferred with regards to the biology. AI techniques can help to simplify the processes of curation, by making more of this task implicit and a natural part of engineering workflows, but practitioners must still choose to invest in and deploy such infrastructure. Likewise, recent results have shown that reproducible calibration of measurements such as fluorescence is much more possible and valuable than has often been assumed [39, 36, 10, 9], but most published studies still use arbitrary or relative units.

5.2 Difficulty in Capturing Expert Knowledge

Many of the potential AI contributions discussed above depend critically on capturing the knowledge of experts in the form of rules, constraints, or representations. This is generally quite difficult to do, however, because much of the knowledge held by experts is not actually explicitly written down anywhere, or is documented in a way that counts on a human reader to make "common sense" assumptions and fill in gaps in the explicitly represented knowledge. Other forms of expertise, particularly in complex physical processes, are transmitted more through apprenticeship than explicit communication. It is reasonable to expect that this will hold for synthetic biology as well, and that one of the key challenges in applying AI techniques to the field will be obtaining and encoding the knowledge held by experts. This can be done either directly (e.g., by having knowledge engineering experts engage in discussions and conduct interviews with synthetic biology practitioners) or implicitly (e.g., by data mining of activity traces of synthetic biology practitioners working in the laboratory that have been captured by cameras, personal electronics, or instrument logs), but in either case may be expected to require investment and cooperation from both synthetic biology experts and AI experts.

5.3 Structural Barriers to Knowledge Exchange

Even given the technical capability to capture expert knowledge in engineering tools, designers may not be able to access or share this knowledge due to cultural, organizational, or legal barriers. Many aspects of organism engineering and an organization's engineering workflow may be considered proprietary, depend on closed systems that are not designed for integration with automation processes, or may be subject to intellectual property claims, all of which can pose significant non-technical obstacles to the application of AI techniques in aid of synthetic biology goals. In computer science, these types of barriers have been mitigated by strong movements in both the scientific and business communities that promote open exchange of knowledge, systems, and methods, and these movements are often credited as an important enabling factor for the rapid advancement of the information economy (e.g., [77, 45]). Similar community organization and establishment of standards and practices that promote open information flow and the exchange of systems and methods will likely be valuable for synthetic biology. Existing efforts include conferences, organizations, and standards groups, e.g., the International Workshop on Bio-Design Automation (IWBDA), the Synthetic Biology: Engineering, Evolution & Design (SEED) conference, the Bio-Design Automation Consortium (BDAC), the Synthetic Biology Open Language (SBOL) standards development group, the BioBricks Foundation (BBF), and the International Genetically Engineered Machine competition (iGEM).

5.4 Gaps in Scientific Knowledge

Because biological organisms are so complex, and so many critical pieces of information are unknown, another barrier likely to be encountered is gaps in the scientific knowledge underlying practices in organism engineering. AI techniques can only produce effective improvement or automation of processes carried out by humans if the processes are fairly well understood in the first place. While this is potentially a serious limitation in some areas, recent results in improving the modeling and predictability of composition in synthetic biology systems (e.g., [50, 55, 21, 7]), give evidence that at least some areas of organism engineering appear to be at a sufficient level of maturity to support application of AI techniques.

5.5 Rapidly Advancing Knowledge and Methods

The continuing rapid advancement in both knowledge and methods also poses the threat that specific AI-enabled methods will be rapidly rendered obsolete. For example, there is ongoing rapid evolution of both DNA synthesis and protocols for assembling DNA fragments into large systems, so any planning technique designed for a specific protocol is likely to have only a short period of relevance. Impactful AI applications will thus most likely need to focus not on specific methods, but instead on providing somewhat more general frameworks for the rapid capture and automation of methods.

5.6 Cost of Adoption vs. Rapid Advance

Finally, adopting new technologies always has a cost in time and energy. No matter how inefficient an existing workflow, switching to a new workflow will always involve a transition period in which the new workflow is integrated with systems, retraining is ongoing, etc. In a rapidly advancing field, this can pose a significant barrier to adoption of new technologies, since substantial process improvements can also be realized simply by waiting for the next improvement in the underlying technological substrate: for example, in the computer world, many promising architectural improvements have been sidelined by the ongoing frequently doublings of processor capabilities. Since most laboratories already have complex and highly customized processes in place, adoption barriers are likely to be a significant challenge for synthetic biology as well. The three main paths to overcoming this challenge are: 1) adoption in new "clean build" environments without an established workflow, as is already happening in a number of synthetic biology startup companies, 2) emergence of significant pain-points that cannot be overcome simply by waiting, and 3) realization of large enough benefits to overcome adoption cost, even in a rapidly advancing environment.

Opportunities and Challenges in Applying Artificial Intelligence to Bioengineering

6 Conclusion

This chapter identifies a number of key opportunities where the application of AI methods may enable significant improvements in the engineering of biological organisms. The general theme of these contributions is management of complexity, by automation of more "routine" processes, streamlined integration of new knowledge and methods, and reduction of friction in interactions both within a laboratory and between organizations.

From an AI perspective, there are many interesting problems for application, particularly given the massive scope and complexity of biological organisms and the problems encountered in their engineering. Complementarily, from a biology perspective, there are many potentially large benefits from integration of AI techniques. Realizing these benefits is likely to require tight collaboration between practitioners of both disciplines. It is also critical to improve awareness and training in both disciplines, as currently most synthetic biologists have had little exposure or training regarding AI methods and so do not appreciate their potential, while most AI researchers have at best a shallow understanding of the problems faced by synthetic biologists and little understanding of where and how AI methods might be effectively applied in this domain. We thus strongly recommend that practitioners interested in realizing these benefits seek out complementary colleagues. As the synthetic biology and AI communities continue to grow in their recognition of what they have to offer one another, we have every confidence that the potential benefits of synthesis between these two disciplines can be realized.

References

- Bartley B, Beal J, Gruenberg R, McLaughlin J, Myers C, Roehner N, Wipat A (Retrieved 2018) SEP 019 – using SBOL to model the design-build-test-learn cycle. https://github.com/SynBioDex/SEPs/blob/master/sep_019.md
- [2] Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular systems for programmed pattern formation. Nature 434:1130– 1134
- [3] Beal J, Bachrach J (2008) Cells are plausible targets for high-level spatial languages. In: Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, IEEE Computer Society, Washington, DC, USA, SASOW '08, pp 284–291, DOI 10.1109/SASOW.2008.14, URL http://dx.doi.org/10.1109/SASOW.2008.14
- [4] Beal J, Yaman F (2012) Toward automated design of cell state detectors. In: 4th International Workshop on Bio-Design Automation
- [5] Beal J, Lu T, Weiss R (2011) Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. PLoS ONE 6(8):e22,490, DOI 10.1371/journal.pone.0022490

- [6] Beal J, Weiss R, Densmore D, Adler A, Appleton E, Babb J, Bhatia S, Davidsohn N, Haddock T, Loyall J, et al (2012) An end-to-end workflow for engineering of biological networks from high-level specifications. ACS synthetic biology 1(8):317–331
- [7] Beal J, Wagner TE, Kitada T, Azizgolshani O, Parker JM, Densmore D, Weiss R (2014) Model-driven engineering of gene expression from RNA replicons. ACS synthetic biology
- [8] Beal J, Adler A, Yaman F (2016) Managing bioengineering complexity with AI techniques. Biosystems 148:40–46
- [9] Beal J, DeLateur N, Teague B, Weiss R, Sexton J, Castillo-Hair S, Tabor JJ (2017) Toward quantitative comparison of fluorescent protein expression levels via fluorescent beads. In: International Workshop on Bio-Design Automation
- [10] Beal J, Haddock-Angelli T, Baldwin G, Gershater M, Dwijayanti A, Storch M, de Mora K, Lizarazo M, Rettberg R, et al (2018) Quantification of bacterial fluorescence using independent calibrants. PloS one 13(6):e0199,432
- [11] Berkeley Software 2009 iGem Team (2009) Eugene. http://2009.igem.org/ Team:Berkeley_Software/Eugene
- [12] Bonnet J, Yin P, Ortiz ME, Subsoontorn P, Endy D (2013) Amplifying genetic logic gates. Science 340(6132):599–603
- [13] Borujeni AE, Channarasappa AS, Salis HM (2013) Translation rate is controlled by coupled trade-offs between site accessibility, selective RNA unfolding and sliding at upstream standby sites. Nucleic acids research p gkt1139
- [14] Bryce D, Goldman R, Kuter U, Beal J, DeHaven M, Geib CS, Plotnick A, Nguyen T, Roehner N (2018) Domain agnostic experimental planning for synthetic biology. In: Hierarchical Planning Workshop
- [15] Canton B, Labno A, Endy D (2008) Refinement and standardization of synthetic biological parts and devices. Nature Biotechnology 26:787–93
- [16] Chen B, Cahoon D, Canton B, Che A (2015) Software for engineering biology in a multi-purpose foundry. In: International Workshop on Bio-Design Automation
- [17] Chen J, Goyal G, Plahar HA, Keasling JD, Stawski N, Hillson NJ (2014) DIVA: More science, less DNA construction. In: International Workshop on Bio-Design Automation
- [18] Cox RS, Madsen C, McLaughlin J, Nguyen T, Roehner N, Bartley B, Bhatia S, Bissell M, Clancy K, Gorochowski T, et al (2018) Synthetic biology open language visual (SBOL visual) version 2.0. Journal of integrative bioinformatics 15(1)
- [19] Cox RS, Madsen C, McLaughlin JA, Nguyen T, Roehner N, Bartley B, Beal J, Bissell M, Choi K, Clancy K, et al (2018) Synthetic biology open language (SBOL) version 2.2.0. Journal of integrative bioinformatics 15(1)
- [20] Czar M, Cai Y, Peccoud J (2009) Writing DNA with GenoCAD. Nucleic Acids Research 37(W40-7)

- [21] Davidsohn N, Beal J, Kiani S, Adler A, Yaman F, Li Y, Xie Z, Weiss R (2014) Accurate predictions of genetic circuit behavior from part characterization and modular composition. ACS Synthetic Biology
- [22] Degtyarenko K, de Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcntara R, Darsow M, Guedj M, Ashburner M (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Research 36:D344–D350, URL http://nar.oxfordjournals.org/content/36/suppl_1/D344.short
- [23] Demir E, Cary MP, Paley S, Fukuda K, Lemer C, Vastrik I, Wu G, D'eustachio P, Schaefer C, Luciano J, et al (2010) The BioPAX community standard for pathway data sharing. Nature biotechnology 28(9):935
- [24] Densmore D, Hsiau THC, Kittleson JT, DeLoache W, Batten C, Anderson JC (2010) Algorithms for automated DNA assembly. Nucleic Acids Research 38(8):2607–2616, URL http://www.ncbi.nlm.nih.gov/pubmed/20335162
- [25] Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M (2005) The sequence ontology: a tool for the unification of genome annotations. Genome biology 6(5):R44
- [26] Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403:335–8
- [27] Esvelt KM, Carlson JC, Liu DR (2011) A system for the continuous directed evolution of biomolecules. Nature 472(7344):499–503
- [28] Finney A, Hucka M, Bornstein BJ, Keating SM, Shapiro BM, Matthews J, Kovitz BK, Schilstra MJ, Funahashi A, Doyle J, Kitano H (2006) Software infrastructure for effective communication and reuse of computational models. In: Szallasi Z, Stelling J, Periwal V (eds) System Modeling in Cell Biology: From Concepts to Nuts and Bolts, MIT Press
- [29] Galdzicki M, Wilson ML, Rodriguez CA, Pocock MR, Oberortner E, Adam L, Adler A, Anderson JC, Beal J, Chandran D, Densmore D, Drory OA, Endy D, Gennari JH, Grunberg R, Ham TS, Hillson NJ, Johnson JD, Kuchinsky A, Lux MW, Madsen C, Misirli G, Myers CJ, Peccoud J, Plahar H, Platt D, Roehner N, Sirin E, Smith TF, Stan GB, Villalobos A, Wipat A, , Sauro HM (2012) Synthetic Biology Open Language (SBOL) Version 1.1.0. RFC 87, URL doi: 1721.1/66172
- [30] Galdzicki M, Clancy KP, Oberortner E, Pocock M, Quinn JY, Rodriguez CA, Roehner N, Wilson ML, Adam L, Anderson JC, et al (2014) The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. Nature biotechnology 32(6):545–550
- [31] Gander MW, Vrana JD, Voje WE, Carothers JM, Klavins E (2017) Digital logic circuits in yeast with CRISPR-dCas9 NOR gates. Nature Communications 8:15,459, DOI 10.1038/ncomms15459
- [32] Gibson DG, Young L, Chuang RY, Venter JC, 3rd CAH, Smith HO (2009) Enzymatic assembly of DNA molecules up to several hundred kilobases. Nature Methods 6(5):343–345

- [33] Ham TS, Dmytriv Z, Plahar H, Chen J, Hillson NJ, Keasling JD (2012) Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools. Nucleic acids research 40(18):e141–e141
- [34] Hartley JL, Temple GF, Brasch MA (2000) DNA cloning using in vitro sitespecific recombination
- [35] Hillson NJ, Rosengarten R, Keasling JD (2012) j5 DNA assembly design automation software. ACS Synthetic Biology 1(1)
- [36] Hoffman RA, Wang L, Bigos M, Nolan JP (2012) NIST/ISAC standardization study: Variability in assignment of intensity values to fluorescence standard beads and in cross calibration of standard beads to hard dyed beads. Cytometry Part A 81(9):785–796
- [37] Ison J, Kalaš M, Jonassen I, Bolser D, Uludag M, McWilliam H, Malone J, Lopez R, Pettifer S, Rice P (2013) Edam: an ontology of bioinformatics operations, types of data and identifiers, topics and formats. Bioinformatics 29(10):1325–1332
- [38] Kanehisa M, Goto S (2000) Kegg: kyoto encyclopedia of genes and genomes. Nucleic acids research 28(1):27–30
- [39] Kelly JR, Rubin AJ, Davis JH, Ajo-Franklin CM, Cumbers J, Czar MJ, de Mora K, Glieberman AL, Monie DD, Endy D (2009) Measuring the activity of BioBrick promoters using an in vivo reference standard. Journal of Biological Engineering 3(4)
- [40] Kiani S, Beal J, Ebrahimkhani MR, Huh J, Hall RN, Xie Z, Li Y, Weiss R (2014) CRISPR transcriptional repression devices and layered circuits in mammalian cells. Nature Methods 11(7):723–726, DOI 10.1038/nmeth.2969
- [41] Kiani S, Beal J, Ebrahimkhani MR, Huh J, Hall RN, Xie Z, Li Y, Weiss R (2014) CRISPR transcriptional repression devices and layered circuits in mammalian cells. Nature methods
- [42] King RD, Rowland J, Oliver SG, Pir P, Aubrey W, Liakata M, Markham M, Soldatova LN, Whelan KE, Clare A, et al (2009) The robot scientist Adam. Computer 42(8):46–54
- [43] Klavins E (Retrieved 2018) Aquarium. http://klavinslab.org/aquarium
- [44] Knight T (2003) Idempotent vector design for standard assembly of BioBricks. Tech. rep., MIT Synthetic Biology Working Group Technical Reports
- [45] Lakhani KR, Von Hippel E (2003) How open source software works: free userto-user assistance. Research policy 32(6):923–943
- [46] Le Novere N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, Demir E, Wegner K, Aladjem MI, Wimalaratne SM, et al (2009) The systems biology graphical notation. Nature biotechnology 27(8):735
- [47] Lebo T, Sahoo S, McGuinness D, Belhajjame K, Cheney J, Corsar D, Garijo D, Soiland-Reyes S, Zednik S, Zhao J (2013) PROV-O: the PROV ontology. W3C recommendation, 30 april 2013. World Wide Web Consortium
- [48] Li Y, Jiang Y, Chen H, Liao W, Li Z, Weiss R, Xie Z (2015) Modular construction of mammalian gene circuits using tale transcriptional repressors. Nature chemical biology 11(3):207–213

- [49] Linshiz G, Stawski N, Poust S, Bi C, Keasling JD, Hillson NJ (2012) PaR-PaR laboratory automation platform. ACS synthetic biology 2(5):216–222
- [50] Lou C, Stanton B, Chen YJ, Munsky B, Voigt CA (2012) Ribozyme-based insulator parts buffer synthetic circuits from genetic context. Nature biotechnology 30(11):1137–1142
- [51] Madsen C, McLaughlin JA, Mısırlı G, Pocock M, Flanagan K, Hallinan J, Wipat A (2016) The SBOL Stack: A Platform for Storing, Publishing, and Sharing Synthetic Biology Designs. ACS Synthetic Biology 5(6):487–497, DOI 10.1021/acssynbio.5b00210, URL http://pubs.acs.org/doi/10.1021/acssynbio.5b00210
- [52] Marchisio MA, Stelling J (2011) Automatic design of digital synthetic gene circuits. PLoS Comput Biol 7(2):e1001,083, DOI 10.1371/journal.pcbi.1001083
- [53] McLaughlin JA, Myers CJ, Zundel Z, Misirli G, Zhang M, Ofiteru ID, Goñi Moreno A, Wipat A (2018) SynBioHub: A standards-enabled design repository for synthetic biology. ACS synthetic biology
- [54] Morrell WC, Birkel GW, Forrer M, Lopez T, Backman TW, Dussault M, Petzold CJ, Baidoo EE, Costello Z, Ando D, et al (2017) The experiment data depot: a web-based software tool for biological experimental data storage, sharing, and visualization. ACS synthetic biology 6(12):2248–2259
- [55] Mutalik VK, Guimaraes JC, Cambray G, Lam C, Christoffersen MJ, Mai QA, Tran AB, Paull M, Keasling JD, Arkin AP, et al (2013) Precise and reliable gene expression via standard transcription and translation initiation elements. Nature methods 10(4):354–360
- [56] Nielsen AA, Der BS, Shin J, Vaidyanathan P, Paralanov V, Strychalski EA, Ross D, Densmore D, Voigt CA (2016) Genetic circuit design automation. Science 352(6281):aac7341
- [57] Ostrov N, Landon M, Guell M, Kuznetsov G, Teramoto J, Cervantes N, Zhou M, Singh K, Napolitano MG, Moosburner M, et al (2016) Design, synthesis, and testing toward a 57-codon genome. Science 353(6301):819–822
- [58] Pedersen M, Phillips A (2009) Towards programming languages for genetic engineering of living cells. Journal of the Royal Society Interface the Royal Society 6(4):S437–S450, URL http://www.ncbi.nlm.nih.gov/pubmed/19369220
- [59] Quinn JY, Cox III RS, Adler A, Beal J, Bhatia S, Cai Y, Chen J, Clancy K, Galdzicki M, Hillson NJ, et al (2015) SBOL visual: a graphical language for genetic designs. PLoS biology 13(12):e1002,310
- [60] Richardson SM, Mitchell LA, Stracquadanio G, Yang K, Dymond JS, DiCarlo JE, Lee D, Huang CLV, Chandrasegaran S, Cai Y, et al (2017) Design of a synthetic yeast genome. Science 355(6329):1040–1044
- [61] Rocklin G, Chidyausiku T, Goreshnik I, Ford A, Houliston S, Lemak A, Carter L, Ravichandran R, Mulligan V, Chevalier A, Arrowsmith C, Baker D (2017) Global analysis of protein folding using massively parallel design, synthesis, and testing. Science 357:168–175, DOI 10.1126/science.aan0693

- [62] Roehner N, Beal J, Clancy K, Bartley B, Misirli G, Grnberg R, Oberortner E, Pocock M, Bissell M, Madsen C, et al (2016) Sharing structure and function in biological design with SBOL 2.0. ACS synthetic biology 5(6):498–506
- [63] Rohl CA, Strauss CE, Misura KM, Baker D (2004) Protein structure prediction using Rosetta. Methods in enzymology 383:66–93
- [64] Salis HM, Mirsky EA, Voigt CA (2009) Automated design of synthetic ribosome binding sites to control protein expression. Nature Biotechnology 27(10):946–950
- [65] Simons KT, Strauss C, Baker D (2001) Prospects for ab initio protein structural genomics. Journal of molecular biology 306(5):1191–1199
- [66] Smanski MJ, Bhatia S, Zhao D, Park Y, Woodruff LB, Giannoukos G, Ciulla D, Busby M, Calderon J, Nicol R, et al (2014) Functional optimization of gene clusters by combinatorial design and assembly. Nature biotechnology 32(12):1241
- [67] Solé R, Amor DR, Duran-Nebreda S, Conde-Pueyo N, Carbonell-Ballestero M, Montañez R (2016) Synthetic collective intelligence. Biosystems 148:47– 61
- [68] Stanton B, Nielsen A, Tamsir A, Clancy K, Peterson T, Voigt C (2014) Genomic mining of prokaryotic repressors for orthogonal logic gates. Nature Chemical Biology 10(2):99–105, DOI 10.1038/nchembio.1411
- [69] Synthace (Retrieved 2018) Antha programming language. https://www.anthalang.org
- [70] Transcriptic (Retrieved 2018) Autoprotocol. http://autoprotocol.org/
- [71] UniProt Consortium (2014) UniProt: a hub for protein information. Nucleic acids research 43(D1):D204–D212
- [72] Vasilev V, Liu C, Haddock T, Bhatia S, Adler A, Yaman F, Beal J, Babb J, Weiss R, Densmore D (2011) A software stack for specification and robotic execution of protocols for synthetic biological engineering. In: 3rd International Workshop on Bio-Design Automation
- [73] Waltemath D, Adams R, Bergmann FT, Hucka M, Kolpakov F, Miller AK, Moraru II, Nickerson D, Sahle S, Snoep JL, et al (2011) Reproducible computational biology experiments with SED-ML-the simulation experiment description markup language. BMC systems biology 5(1):198
- [74] Wang C, Bradley P, Baker D (2007) Protein–protein docking with backbone flexibility. Journal of molecular biology 373(2):503–519
- [75] Weber E, Engler C, Gruetzner R, Werner S, Marillonnet S (2011) A modular cloning system for standardized assembly of multigene constructs. PloS one 6(2):e16,765
- [76] Weinberg BH, Pham NTH, Caraballo LD, Lozanoski T, Engel A, Bhatia S, Wong WW (2017) Large-scale design of robust genetic circuits with multiple inputs and outputs for mammalian cells. Nature Biotechnology 35(5):453– 462, DOI 10.1038/nbt.3805
- [77] West J, Gallagher S (2006) Challenges of open innovation: the paradox of firm investment in open-source software. R&d Management 36(3):319–331

- [78] Xiang Z, Mungall C, Ruttenberg A, He Y (2011) Ontobee: A linked data server and browser for ontology terms. In: ICBO
- [79] Xie Z, Wroblewska L, Prochazka L, Weiss R, Benenson Y (2011) Multi-input RNAi-based logic circuit for identification of specific cancer cells. Science 333(6047):1307–1311, DOI 10.1126/science.1205527
- [80] Yaman F, Bhatia S, Adler A, Densmore D, Beal J (2012) Automated selection of synthetic biology parts for genetic regulatory networks. ACS Synthetic Biology 1(8):332–344, DOI 10.1021/sb300032y