BioSystems 148 (2016) 40–46

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Managing bioengineering complexity with AI techniques

Jacob Beal*, Aaron Adler, Fusun Yaman

Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, United States

ARTICLE INFO

Article history:
Received 9 April 2015
Received in revised form 3 August 2015
Accepted 14 August 2015
Available online 3 September 2015

Keywords:
Synthetic biology
Artificial intelligence
Automation

ABSTRACT

Our capabilities for systematic design and engineering of biological systems are rapidly increasing. Effectively engineering such systems, however, requires the synthesis of a rapidly expanding and changing complex body of knowledge, protocols, and methodologies. Many of the problems in managing this complexity, however, appear susceptible to being addressed by artificial intelligence (AI) techniques, i.e., methods enabling computers to represent, acquire, and employ knowledge. Such methods can be employed to automate physical and informational "routine" work and thus better allow humans to focus their attention on the deeper scientific and engineering issues. This paper examines the potential impact of AI on the engineering of biological organisms through the lens of a typical organism engineering workflow. We identify a number of key opportunities for significant impact, as well as challenges that must be overcome.

1. Introduction

In recent years, humanity has rapidly increased its capabilities for engineering the behavior of cells by modification of their genetic machinery. These emerging practices of synthetic biology hold the potential for revolutionary advances in many important application areas, including medical therapies, vaccination, manufacturing of proteins and other organic compounds, environmental remediation, etc. As these capabilities and their potential applications grow, however, the complexity and cross-disciplinary knowledge required to employ them is growing rapidly as well. Managing the complexity of biological engineering is thus a problem of increasing importance, as no single laboratory can contain all of the different types of cross-disciplinary expertise necessary for effective integration of all of these techniques into a reliable, rapid, and effective engineering workflow. Similarly, the rapid pace of advancement also makes it important to have good methods for integration of new knowledge and methods into organism engineering workflows, and to reduce friction in interactions both within a laboratory and in the transfer of knowledge, specifications, and materials from laboratory to laboratory.

Fortunately, hints for how to proceed may be drawn from the way in which other fields have approached similar problems in managing engineering complexity. In areas as disparate as the engineering of software, electronics, and mechanical systems, the common response has been to use artificial intelligence (AI) techniques to capture human expert knowledge and embed it into assistive tools.[1] In particular, our discussion in this paper refers to AI in the sense of knowledge-based computing systems, after the fashion of Winston (1992), including the representation of knowledge (e.g., semantic networks, frame representations), the acquisition of knowledge (e.g., machine learning, hypothesis generation), its employment in planning and decision making (e.g., expert systems, constraint-based reasoning, planning under uncertainty), and also in automated action (e.g., robotics). Such tools can then carry out routine work and error checking, allowing a human engineer to operate at a higher level of abstraction and to focus on the more core issues in systems design.

Arguably, systems in other fields like software and mechanical engineering may be better understood and less complex than biological organisms, but already there are many opportunities for AI techniques to make a major difference in the efficacy of organism engineering. Likewise, the high degree of success found in bioinformatics, which involves much application of AI techniques to systems biology, augurs well for their applicability to engineering as well, where in some ways the problem may actually be easier because here we have control over the contents of the system to be designed and analyzed. Finally, the history of AI impact on other fields suggests that AI is likely to have transformative impact not

* Corresponding author.
E-mail address: jakebeal@gmail.com (J. Beal).

[1] It is worth noting that in many cases the work is so successful and wide-reaching that we forget that its foundation is AI: consider, for example, the fact that programming languages are the distilled result of AI research into "automatic programming."

http://dx.doi.org/10.1016/j.biosystems.2015.08.006
0303-2647/© 2015 Elsevier Ireland Ltd. All rights reserved.

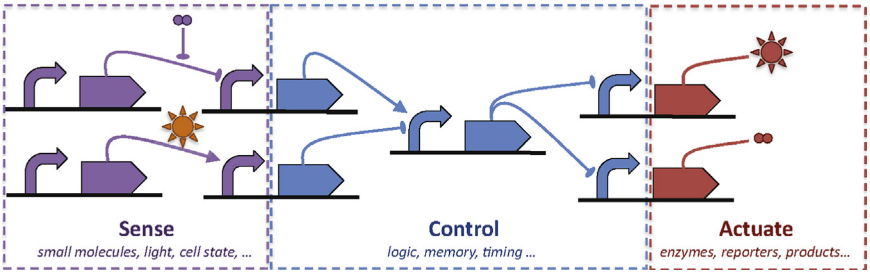
Sense
small molecules, light, cell state, ...
Control
logic, memory, timing ...
Actuate
enzymes, reporters, products...

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.

through the construction of a single dramatic system, but rather through the progressive relaxation of many different bottlenecks and points of friction by the deployment of a wide variety of AI-based tools.

Accordingly, this paper examines opportunities for applications of AI techniques to managing the complexity of engineering biological organisms, with a particular eye to short-term “low hanging fruit” applications where a high impact is likely to be achievable with readily available AI methods. We develop this analysis against a typical organism engineering workflow, presented in Section 2, then identify particular opportunities for AI contributions in Section 3. In a complementary vein, Section 4 identifies key challenges likely to be encountered in the pursuit of such work. Finally, Section 5 closes with recommendations for future work.

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 paper 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 design–build–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 Fig. 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).
- **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 paper, 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 (Fig. 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.

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

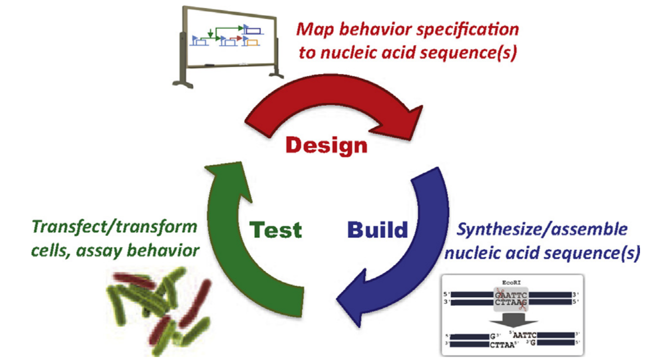

Fig. 2. A typical synthetic biology workflow for organism engineering may be viewed as a cycle of three stages: *design* maps a behavior specification to a nucleic acid sequence intended to realize this behavior, *build* draws on synthesis and/or assembly protocols to fabricate said nucleic acid sequence,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

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., Wang et al., 2007; Esvelt et al., 2011; Kiani et al., 2014), 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 BioBricks (Canton et al., 2008; Knight, 2003), Gateway-Gibson (Invitrogen Corporation, 2010; Gibson et al., 2009), or MoClo (Weber et al., 2011)) 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 (Fig. 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 organism to meet a specification typically requires many iterations of this design-build-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.

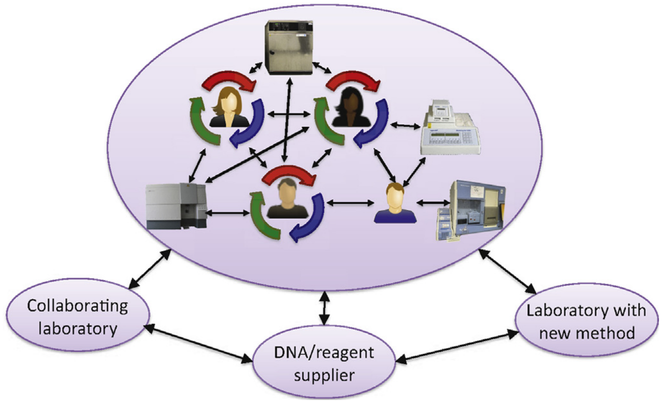

Fig. 3. Organism engineering does not take place in isolation: in addition to potentially many design–build–test cycles of a single project, there are often complex interactions both within a laboratory and between laboratories, in which many different workflows interact.

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.

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,

Table 1
Summary of near-term opportunities where there is a high potential for artificial intelligence techniques to contribute to engineering of sense–control–actuate systems in biological organisms, as discussed in Section 3.

Engineering challenge	Key AI techniques	Examples
Machine-assisted gene circuit design	Expert systems, constraint-based reasoning, heuristic search, optimization, machine learning, multi-agent systems	Beal et al. (2011), Yaman et al. (2012), Czar et al. (2009), Pedersen and Phillips (2009), Berkeley Software (2009), Marchisio and Stelling (2011), Beal and Yaman (2012), Beal and Bachrach (2008)
Flexible protocol automation	Robotics, planning under uncertainty	Densmore et al. (2010), Hillson et al. (2012), Beal et al. (2012), Linshiz et al. (2012), Synthace (2015), Vasilev et al. (2011)
Assay interpretation and modeling	Machine learning, qualitative reasoning	King et al. (2009)
Lab management and optimization	Heuristic search, optimization, planning under uncertainty	Chen et al. (2015, 2014), Vasilev et al. (2011), Densmore et al. (2010)
Represent/exchange designs	Semantic networks, ontologies	Galdzicki et al. (2014, 2011), Eilbeck et al. (2005)
Represent/exchange protocols	Semantic networks, schemas	Linshiz et al. (2012), Synthace (2015)

AI techniques are already key to a number of specialized subtasks within the design process: for example, machine learning and heuristic search techniques are frequently used in genome mining (e.g., Stanton et al., 2014) and protein design (e.g., Rohl et al., 2004).

Turning to the integrative design challenges that are the focus of this paper, there are many opportunities there as well, of which we shall discuss three 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, given in Beal et al. (2011), uses a motif-based technique to design and optimize genetic regulatory network topologies from behavior specifications expressed in a high-level programming language. The resulting “abstract” genetic regulatory networks can then be mapped to a fully instantiated genetic regulatory network by the MatchMaker (Yaman et al., 2012) constraint-based reasoning system, forming a complete design ready to be built. A number of other design tools have been developed that approach the design synthesis problem from a variety of perspectives, all enabled by various forms of knowledge-representation and reasoning (e.g., Czar et al., 2009; Pedersen and Phillips, 2009; Berkeley Software, 2009; Marchisio and Stelling, 2011).

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 Xie et al. (2011) 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 much faster, more reliably, and likely with better results as well, as demonstrated with the information-based method presented in Beal and Yaman (2012). 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 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 and Leibler (2000) 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 Beal and Bachrach (2008), 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 hand-designed band detector (Basu et al., 2005). 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, and it is just such systems at which work of the multi-agent systems community is squarely targeted.

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., Densmore et al., 2010; Hillson et al., 2012; Beal et al., 2012).

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 automation much cheaper to employ by increasing the flexibility with which protocols can be applied and allowing them to be specified in a more lightweight manner 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 number 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. Early systems targeted at such automation include (Linshiz et al., 2012; Synthace, 2015; Vasilev et al., 2011).

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. Furthermore, learning can be taken one step further into active design of experiments: one extreme prototype eliminates humans entirely (King et al., 2009), 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 are 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 sleep-deprived 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 (Chen et al., 2015), Diva (Chen et al., 2014), and Puppeteer (Vasilev et al., 2011) 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 Densmore et al. (2010).

3.5. *Knowledge representation and data interchange*

Beyond an individual laboratory, there is a need to be able to effectively exchange specifications, designs, protocols, and knowledge between different practitioners. Knowledge representation techniques from AI can allow such methods to be mediated by computers, greatly increasing the effective pace and power available in such interchanges. Such methods are already widespread in business-to-business integration in the commercial world, and the synthetic biology community has already been working on developing standards for the representation and interchange of nucleic acid sequence designs, in the form of the Synthetic Biology Open Language (SBOL) (Galdzicki et al., 2011, 2014), which is based on the AI techniques of semantic network representations and ontology construction. This work also encompasses an effort to visually represent these designs, and builds upon existing ontological representation of genetic constructs such as genes and promoters (Eilbeck et al., 2005).

Representation and exchange of experimental protocols has also attracted interest (e.g., Linshiz et al., 2012; Synthace, 2015), with the aim of allowing experiments to be reliably replicated in different laboratories, and other areas of replication and exchange friction are also likely to become the focus of similar knowledge-representation efforts in the future.

4. 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.

4.1. *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, instruments logs, etc.), but in either case may be expected to require investment and cooperation from both synthetic biology experts and AI experts.

4.2. *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., West and Gallagher, 2006; Lakhani and Von Hippel, 2003). 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 (and some such efforts that already exist are mentioned below in Section 5).

4.3. *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., Lou et al., 2012; Mutalik et al., 2013; Davidsohn et al., 2015; Beal et al., 2015), 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.

4.4. *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.

4.5. *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.

5. Summary and recommendations

This paper has identified 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. We thus strongly recommend that practitioners interested in realizing these benefits seek out complementary colleagues. Furthermore, there now exist a number of cross-disciplinary events and organizations that include an explicit intent to support the development of such collaborations, including:

- Conferences and organizations focusing on biological design automation, such as the International Workshop on Bio-Design Automation (IWBDA), and the Bio-Design Automation Consortium (BDAC).
- Various groups focused on developing synthetic biology standards, e.g., for protocol automation, methods exchange, design representation, and data exchange, such as the Synthetic Biology Open Language (SBOL) standards development group, the BioBricks Foundation (BBF), and the NIST Synthetic Biology Standards Consortium (SBSC).
- Student-focused synthetic biology events, such as the International Genetically Engineered Machine competition (iGEM).

Thus, 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

Basu, S., Gerchman, Y., Collins, C.H., Arnold, F.H., Weiss, R., 2005. A synthetic multicellular systems for programmed pattern formation. Nature 434, 1130–1134.

Beal, J., Bachrach, J., 2008. Cells are plausible targets for high-level spatial languages. In: IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, pp. 284–291, http://doi.ieeecomputersociety.org/10.1109/SASOW.2008.14.

Beal, J., Yaman, F., 2012. Toward automated design of cell state detectors. In: 4th International Workshop on Bio-Design Automation.

Beal, J., Lu, T., Weiss, R., 2011. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. PLoS ONE 6 (8), e22490+, doi:10.1371/journal.pone.0022490. URL http://dx.doi.org/10.1371/journal.pone.0022490.

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 Synth. Biol. 1 (8), 317–331.

Beal, J., Wagner, T.E., Kitada, T., Azizgolshani, O., Parker, J.M., Densmore, D., Weiss, R., 2015. Model-driven engineering of gene expression from RNA replicons. ACS Synth. Biol. 4 (1), 48–56, http://dx.doi.org/10.1021/sb500173f.

Berkeley Software 2009 iGem Team, Eugene, http://2009.igem.org/Team:Berkeley_Software/Eugene (October 2009).

Canton, B., Labno, A., Endy, D., 2008. Refinement and standardization of synthetic biological parts and devices. Nat. Biotechnol. 26, 787–793.

Chen, J., Goyal, G., Plahar, H.A., Keasling, J.D., Stawski, N., Hillson, N.J., 2014. Diva: more science less DNA construction. In: International Workshop on Bio-Design Automation.

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.

Czar, M., Cai, Y., Peccoud, T., 2009. Writing DNA with GenoCAD. Nucleic Acids Res. 37 (W40-7).

Davidsohn, N., Beal, J., Kiani, S., Adler, A., Yaman, F., Li, Y., Xie, Z., Weiss, R., 2015. Accurate predictions of genetic circuit behavior from part characterization and modular composition. ACS Synth. Biol. 4 (6), 673–681, http://dx.doi.org/10.1021/sb500263b.

Densmore, D., Hsiau, T.H.C., Kittleson, J.T., DeLoache, W., Batten, C., Anderson, J.C., 2010. Algorithms for automated DNA assembly. Nucleic Acids Res. 38 (8), 2607–2616, http://dx.doi.org/10.1093/nar/gkq165, URL http://dx.doi.org/10.1093/nar/gkq165.

Eilbeck, K., Lewis, S.E., Mungall, C.J., Yandell, M., Stein, L., Durbin, R., Ashburner, M., 2005. The sequence ontology: a tool for the unification of genome annotations. Genome Biol. 6 (5), R44.

Elowitz, M.B., Leibler, S., 2000. A synthetic oscillatory network of transcriptional regulators. Nature 403, 335–338.

Esvelt, K.M., Carlson, J.C., Liu, D.R., 2011. A system for the continuous directed evolution of biomolecules. Nature 472 (7344), 499–503.

Galdzicki, M., Wilson, M.L., Rodriguez, C.A., Adam, L., Adler, A., Anderson, J.C., Beal, J., Chandran, D., Densmore, D., Drory, O.A., Endy, D., Gennari, J.H., Grünberg, R., Ham, T.S., Kuchinsky, A., Lux, M.W., Madsen, C., Misirli, G., Myers, C.J., Peccoud, J., Plahar, H., Pocock, M.R., Roehner, N., Smith, T.F., Stan, G.-B., Villalobos, A., Wipat, A., Sauro, H.M. Synthetic Biology Open Language (SBOL) Version 1.0.0, RFC 85 (Oct. 2011). doi: 1721.1/66172.

Galdzicki, M., Clancy, K.P., Oberortner, E., Pocock, M., Quinn, J.Y., Rodriguez, C.A., Roehner, N., Wilson, M.L., Adam, L., Anderson, J.C., et al., 2014. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. Nat. Biotechnol. 32 (6), 545–550.

Gibson, D.G., Young, L., Chuang, R.-Y., Venter, J.C., Hutchison, C.A., Smith, H.O., 2009. Enzymatic assembly of DNA molecules up to several hundred kilobases. Nat. Methods 6 (5), 343–345.

Hillson, N.J., Rosengarten, R.D., Keasling, J.D., 2012. j5 DNA assembly design automation software. ACS Synth. Biol. 1 (1), 14–21.

Invitrogen Corporation, May 2010. Gateway® Technology, Version E Edition.

Kiani, S., Beal, J., Ebrahimkhani, M.R., Huh, J., Hall, R.N., Xie, Z., Li, Y., Weiss, R., 2014. Crispr transcriptional repression devices and layered circuits in mammalian cells. Nat. Methods. 11 (7), 723–726.

King, R.D., Rowland, J., Oliver, S.G., Pir, P., Aubrey, W., Liakata, M., Markham, M., Soldatova, L.N., Whelan, K.E., Clare, A., et al., 2009. The robot scientist adam. Computer 8, 46–54.

Knight, T., 2003. Idempotent vector design for standard assembly of BioBricks. In: Tech. rep., MIT Synthetic Biology Working Group Technical Reports.

Lakhani, K.R., Von Hippel, E., 2003. How open source software works:free user-to-user assistance. Res. Policy 32 (6), 923–943.

Linshiz, G., Stawski, N., Poust, S., Bi, C., Keasling, J.D., Hillson, N.J., 2012. Par-par laboratory automation platform. ACS Synth. Biol. 2 (5), 216–222.

Lou, C., Stanton, B., Chen, Y.-J., Munsky, B., Voigt, C.A., 2012. Ribozyme-based insulator parts buffer synthetic circuits from genetic context. Nat. Biotechnol. 30 (11), 1137–1142.

Marchisio, M.A., Stelling, J., 2011. Automatic design of digital synthetic gene circuits. PLoS Comput. Biol. 7 (2), e1001083, http://dx.doi.org/10.1371/journal.pcbi.1001083.

Mutalik, V.K., Guimaraes, J.C., Cambray, G., Lam, C., Christoffersen, M.J., Mai, Q.-A., Tran, A.B., Paull, M., Keasling, J.D., Arkin, A.P., et al., 2013. Precise and reliable gene expression via standard transcription and translation initiation elements. Nat. Methods 10 (4), 354–360.

Pedersen, M., Phillips, A., 2009. Towards programming languages for genetic engineering of living cells. J. R. Soc. Interface, S437–S450, http://dx.doi.org/10.1098/rsif.2008.0516.focus.

Rohl, C.A., Strauss, C.E., Misura, K.M., Baker, D., 2004. Protein structure prediction using rosetta. Methods Enzymol. 383, 66–93.

Stanton, B., Nielsen, A., Tamsir, A., Clancy, K., Peterson, T., Voigt, C., 2014. Genomic mining of prokaryotic repressors for orthogonal logic gates. Nat. Chem. Biol. 10 (2), 99–105, http://dx.doi.org/10.1038/nchembio.1411.

Synthace. Antha programming language. https://www.antha-lang.org (Retrieved 2015).

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.

Wang, C., Bradley, P., Baker, D., 2007. Protein-protein docking with backbone flexibility. J. Mol. Biol. 373 (2), 503–519.

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), e16765.

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.

Winston, P., 1992. Artificial Intelligence, 3rd ed. Addison-Wesley, Reading, MA.

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, arXiv:http://www.sciencemag.org/content/333/6047/1307.full.pdf, doi:10.1126/science.1205527. URL http://www.sciencemag.org/content/333/6047/1307.abstract.

Yaman, F., Bhatia, S., Adler, A., Densmore, D., Beal, J., 2012. Automated selection of synthetic biology parts for genetic regulatory networks. ACS Synth. Biol. 1 (8), 332–344, arXiv:http://pubs.acs.org/doi/pdf/10.1021/sb300032y, doi:10.1021/sb300032y. URL http://pubs.acs.org/doi/abs/10.1021/sb300032y.