ABSTRACT

With the growing complexity of synthetic biological circuits, robust and systematic methods are needed for design and test. Leveraging lessons learned from the semiconductor and design automation industries, synthetic biologists are starting to adopt computer-aided design and verification software with some success. However, due to the great challenges associated with designing synthetic biological circuits, this nascent approach has to address many problems not present in electronic circuits. In this session, three leading synthetic biologists will share how they have developed software tools to help design and verify their synthetic circuits, the unique challenges they face, and their insights into the next generation of tools for synthetic biology.

Categories and Subject Descriptors
J.3 [LIFE AND MEDICAL SCIENCES]

General Terms
synthetic biology, systems biology, computational biology, biological circuits

Keywords
bio-design automation, genetic compiler, chemical reaction networks, molecular computation, biological parts

1. INTRODUCTION

Electronic circuit design is the process of selecting components from a group of well understood electronic components (e.g. transistors, op amps, diodes, etc) and combining these components together to implement the desired functionality. This design process currently benefits from having well-understood, characterized components. In addition, there are rules for the composition of circuit primitives, mature manufacturing processes, and computer software to help with this process.

By contrast, biological circuit design is in its infancy. While it is tempting to broadly apply techniques and analogies from electronic circuit design, it is important to respect key differences between the two disciplines. Chief among these are the need for truly orthogonal signaling mechanisms, metabolic requirements of host organisms, and the lack of control and/or observability in the systems being designed. Biological circuit design will ideally integrate algorithms and techniques which help to address these issues while leveraging biological systems’ ability to adapt, evolve, and be self maintaining/repairing. In addition, biological circuit design methodologies must also leverage biophysical modeling tools to create predictive design software to tune designs for specific performance in the face of varied, often loosely characterized primitives.

This special session begins to discuss these challenges and lay the foundation for designing biological circuits in a more disciplined manner. The first talk is about biological part characteristics and how to use these parts to create controllers for gene expression engineering. The second talk is about creating genetic compilers that incorporate biophysical models and promise to raise the level of abstraction at which biological systems are designed. Finally, the third talk is about techniques to create chemical reaction networks in order to provide a new computational paradigm in this space.
2. SPEAKER SUMMARIES

Scalable Parts Families, Context, and Computational Design for Gene Expression Engineering
Adam Arkin (Speaker)
Univ. of California, Berkeley

Our current ability to engineer biological circuits is hindered by design cycles that are costly in terms of time and money. Constructs invariably fail to operate as desired, or evolving away from the desired function once deployed. Synthetic biologists seek to understand biological design principles and use them to create technologies that increase the efficiency of the genetic engineering design cycle. Central to the approach is the creation of biological parts—encapsulated functions that can be composited together to create new pathways with predictable behaviors. We have defined five desirable characteristics of biological parts—indirection, reliability, tunability, orthogonality and composable. We propose that the creation of appropriate sets of families of parts with these properties is a prerequisite for efficient, predictable engineering of new function in cells and will enable a large increase in the sophistication of genetic engineering applications.

We demonstrate these concepts with examples of gene expression controllers that exercise these properties and point to how the engineering goals of synthetic biology can be met. Using the 5’ UTR as an controlling switchboard for gene regulation we show how a specially chosen set of families of RNA-based parts can create a nearly complete set of controllers for gene expression engineering. We further show how they composite together and are affected by the cellular context in which they find themselves. The latter measures point to host systems that might be engineered to increase the independence of our engineered systems from host physiology.

While there are many applications for these in basic metabolic engineering and production host optimization, we further argue that the true power of such a framework is only realized when engineering the complex behaviors of cells, such as required for operation beyond the bioreactor for applications, for example, in agriculture, cell/virus-based therapies, and bioremediation. We discuss a few of our results in this area.

Gene and Cellular Circuit Design
Chris Voigt (Speaker)
Univ. of California, San Francisco

A genetic compiler for synthetic biology would enable biological systems to be specified in a high-level language (akin to Java or Verilog), which is then automatically converted into a DNA sequence. I will describe several advances in our lab towards this goal. First, genetic circuits need to be designed specifically for assembly by computational algorithms. They need to be simple, modular, and easily rewired. To this end, we have developed several logic gates that can be layered. I will describe work to use DNA synthesis to rapidly create many orthogonal variants of these gates. Second, methods need to be developed to computationally predict sequences that will properly connect these gates. We have developed biophysical models that predict the strength of genetic parts to aid this process. Finally, relevant methods need to be developed to convert a desired higher-level language into integrated gates. Together, these approaches move towards the dream of being able to program living cells.

Compiling and Verifying DNA-Based Chemical Reaction Network Implementations
Seung Woo Shin
California Institute of Technology

One goal of molecular programming and synthetic biology is to build chemical circuits that can control chemical processes at the molecular level. Remarkably, it has been shown that synthesized DNA molecules can be used to construct complex chemical circuits that operate without any enzyme or cellular component. However, designing DNA molecules at the individual nucleotide base level is often difficult and laborious, and thus formal chemical reaction networks (CRNs) have been proposed as a higher-level programming language. So far, several general-purpose constructions have been described for designing synthetic DNA molecules that simulate the behavior of arbitrary CRNs, and many more are being actively investigated.

Here, we solve two problems related to this topic. First, we present a general-purpose CRN-to-DNA compiler that can apply user-defined compilation schemes for translating formal CRNs to domain-level specifications for DNA molecules. In doing so, we develop a language in which such schemes can be concisely and precisely described. This compiler can greatly reduce the amount of tedious manual labor faced by researchers working in the field. Second, we present a general method for the formal verification of the correctness of such compilation. We first show that this problem reduces to testing a notion of behavioral equivalence between two CRNs, and then we construct a mathematical formalism in which that notion can be precisely defined. Finally, we provide algorithms for testing that notion. This verification process can be thought of as an equivalent of model checking in molecular computation, and we hope that the generality of our verification techniques will eventually allow us to apply them not only to DNA-based CRN implementations but to a wider class of molecular programs.