Results from TASBE

Jacob Beal
BBN Technologies
10 Moulton Street
Cambridge, MA, USA 02138
jakebeal@bbn.com

Ron Weiss
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge, MA, USA 02139
rweiss@mit.edu

Douglas Densmore
Boston University
8 Saint Mary's St.
Boston, MA, USA 02215
dougd@bu.edu

1. INTRODUCTION

The TASBE (A Tool-Chain to Accelerate Synthetic Biological Engineering) project [2] developed a tool-chain (Figure 1) to design and build synthetic biology systems. These tools convert a circuit description written in a high-level language to an implementation in cells, assembled with laboratory robots. Each tool addresses a different sub-problem. This paper describes each tool and its key results.

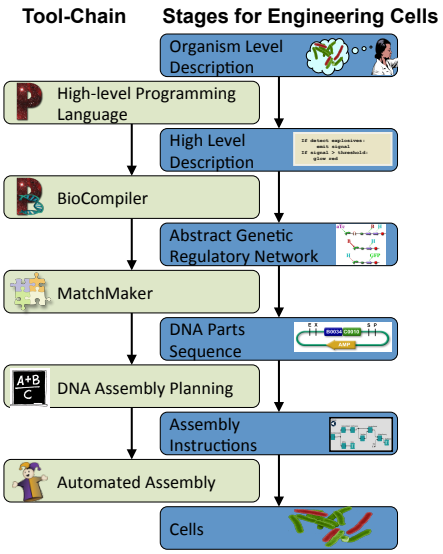

Figure 1: Engineering process and corresponding TASBE tools.

TASBE Characterization is a detailed methodology that gathers highly accurate data for synthetic biology parts. This data enable the transformations done by the tools in the tool-chain. The TASBE project gathered data for many biological parts. The **BioCompiler** begins with a design written in a high-level language. The design is compiled and optimized, producing an abstract genetic regulator network (AGRN). The resulting optimized designs are equivalent to those produced by human experts. The AGRN can be simulated to verify that the circuit produces the desired effect. **MatchMaker** converts this AGRN into an instantiated genetic regulatory network (GRN) by selecting parts from a database of parts that meet the TASBE Characterization standards. MatchMaker ensures that the parts used in the GRN are signal compatible, thus enabling composite design. Finally, **DNA Assembly Planning** and **Automated Assembly** (including robotic assembly) converts the GRN into a part sequence and assembly instructions for a robot or human. The DNA sequence can then be assembled and inserted into cells for execution.

Work partially sponsored by DARPA I2O under contract HR0011-10-C-0168; the views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Government.
IWBDA '12 San Fransisco, California, USA

2. TASBE CHARACTERIZATION

Our work in TASBE has shown us that, with regards to DNA part characterization, any type of compositional design will need at least: 1) Large numbers of single-cell measurements (as opposed to population average values), 2) Measurements of the level of part output signal(s) across the full dynamic range of levels of part input signal(s), 3) Data to determine the per-copy effect of the construct, and 4) The statistical distribution of single-cell output levels for each input level, in order to estimate the variability of behavior.

Prior characterization efforts, however, have generally not yielded enough high-quality information to enable compositional design. In the TASBE project we have developed a new characterization technique (both analytics and wet-lab) capable of producing such data (Figure 2). We have published a technical report [3] that describes the techniques we have developed, along with examples of their application, so that the techniques can be accurately used by others.

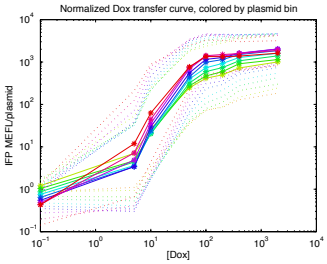

Figure 2: Transfer curve obtained for Dox induction using TASBE characterization techniques.

3. BIOCOMPILER

We defined a high-level programming language [1] for biological designs. This language is based on a spatial computing language to support modeling the multi-cellular interactions that will be necessary for synthetic biology applications. The designs specified in the high-level language are compiled to AGRNs by composing motifs and optimizations (Figure 3). BioCompiler is the first tool that allows arbitrary boolean logic and feedback systems to be specified and then designs an appropriate genetic regulatory network

automatically. The optimization is competitive with human experts and homologous with hand designed circuits. Additionally, the function of the circuit can be verified using an ODE simulation. Team biologists now routinely use the BioCompiler to design AGRNs because the output is less error prone and faster than hand designs.

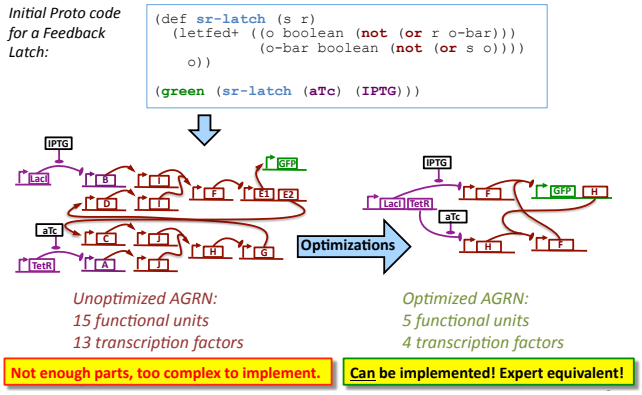

Figure 3: A high-level program is compiled to an AGRN and then optimized.

4. MATCHMAKER

In order to realize an AGRN, we need to instantiate the abstract components of the network with actual biological parts. We formally defined the problem of transforming the abstract network produced by the BioCompiler into a sequence of DNA parts given the availability of the parts and the biological constraints on them. We identified three steps in this transformation (Figure 4): 1) *Feature Matching* is the problem of assigning a single feature to each node in the AGRN such that the repression/activation relationships are satisfied. Basically this converts an AGRN into a GRN. 2) *Signal Matching* is the problem of finding the best GRN with respect to the chemical signal compatibility. 3) *Part Matching* is the problem of finding the shortest part sequence that implements the GRN. We studied the theoretical complexity of these subproblems. We have implemented our algorithms in the software tool MatchMaker, which is also integrated with Clotho [4] for accessing databases.

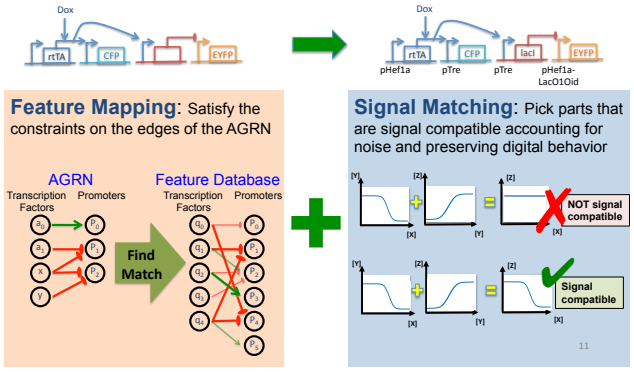

Figure 4: A visualization of the first two MatchMaker steps: feature matching and signal matching. The AGRN is transformed into a GRN.

5. DNA ASSEMBLY PLANNING AND AUTOMATED ASSEMBLY

The last stages of the tool-chain[5] plan how to assemble the part sequence and then convert the sequence into assembly instructions that can be executed on a laboratory robot. Versions of these tools, customized for specific laboratory hardware and cellular platforms, are running at the MIT and BU labs. These tools take into account resource allocation and integrate design and data management tools with a language for protocol specification and robotic execution.

6. RESULTS

A synthetic biology tool-chain can bring the ideas of programmability, abstraction, and languages to synthetic biology. The goal of this project was to validate the viability of the tool-chain approach. We have implemented a working proof-of-concept implementation of the TASBE infrastructure: decomposing the problem has made the development process more tractable, results are rapidly usable by other components (progress on characterization can be exploited by MatchMaker), and we have been able to bring programming language, artificial intelligence, CAD, and biology expertise to bear on the problem despite no individual member of the team being an expert in all fields. Three key results provide evidence that TASBE is a unique, novel and viable architecture: 1) High-level programs have compiled to designs equivalent to hand-designed systems of DNA parts that are operating correctly *in vivo*, 2) Characterization of transcriptional logic parts has shown acceptable amplification to support digital abstractions and tractable part matching, and 3) The upper portions of TASBE are completely modular with respect to the choice of assembly target between BioBrick-protocol parts for *E. coli* and new-protocol parts for mammalian cells (Figure 5).

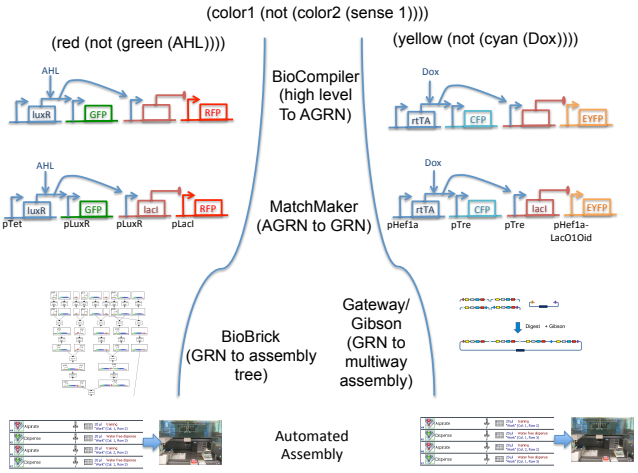

Figure 5: The same high-level program can be compiled to a platform specific (left: *E. coli*; right: mammalian) program using TASBE.

We plan to make these tools available either as Clotho Apps or in the case of BioCompiler and TASBE Characterization through a web service interface. Finally the TASBE project provided a foundation for DARPA efforts such as the Living Foundries program.

7. ADDITIONAL AUTHORS

Aaron Adler (BBN Technologies, email: aadler@bbn.com), Evan Appleton (Boston University, email: eapple@bu.edu), Jonathan Babb (MIT, email: jbabb@mit.edu), Swapnil Bhatia (Boston University, email: swapnilb@bu.edu), Noah Davidsohn (MIT, email: ndavidso@mit.edu), Traci Haddock (Boston University, email: thaddock@bu.edu), Joseph Loyall (BBN Technologies, email: jloyall@bbn.com), Richard Schantz (BBN Technologies, email: rschantz@bbn.com) Viktor Vasilev (Boston University, email: vvasilev@bu.edu), and Fusun Yaman (BBN Technologies, email: fusun@bbn.com).

8. REFERENCES

[1] J. Beal, T. Lu, and R. Weiss. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS ONE*, 6(8):e22490, August 2011.

[2] J. Beal, R. Weiss, D. Densmore, A. Adler, J. Babb, S. Bhatia, N. Davidsohn, T. Haddock, F. Yaman, R. Schantz, and J. Loyall. TASBE: A tool-chain to accelerate synthetic biological engineering. In *Proceedings of the 3rd International Workshop on Bio-Design Automation*, pages 19–21, June 2011.

[3] J. Beal, R. Weiss, F. Yaman, N. Davidsohn, and A. Adler. A method for fast, high-precision characterization of synthetic biology devices. Technical Report MIT-CSAIL-TR-2012-008, MIT, April 2012. http://hdl.handle.net/1721.1/69973.

[4] D. Densmore, A. V. Devender, M. Johnson, and N. Sritanyaratana. A platform-based design environment for synthetic biological systems. In *TAPIA '09*, pages 24–29. ACM, April 2009.

[5] V. Vasilev, C. Liu, T. Haddock, S. Bhatia, A. Adler, F. Yaman, J. Beal, J. Babb, R. Weiss, and D. Densmore. A software stack for specification and robotic execution of protocols for synthetic biological engineering. In *Proceedings of the 3rd International Workshop on Bio-Design Automation*, pages 24–25, June 2011.