A Novel Automated Assembly Approach for Use in Synthetic Circuit Construction

Traci Haddock¹, Swapnil Bhatia¹, Viktor Vasilev¹, Chenkai Liu¹, Aaron Adler², Jake Beal¹, Jonathan Babb³, Ron Weiss³ and Douglas Densmore¹

1 Department of Electrical and Computer Engineering, Boston University, Boston, MA; 2 BBN Technologies, Cambridge, MA; 3 Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA

INTRODUCTION

Synthetic biology today is lacking assembly automation tools to help simplify the creation of synthetic DNA circuits in the laboratory. While current automation can employ liquid handling robots, our automation vision encompasses a forward engineering approach which begins with high level descriptions of the desired functionality. We are designing and implementing an integrated tool-chain that can be applied throughout the synthetic biological design workflow, starting with the translation of an abstract behavioral description all the way to the execution of protocols on a liquid handling robot (Beal et al, A Tool-Chain to Accelerate Synthetic Biological Design, SB5.0). Here, we focus on the final stages of this flow and how it fits into the context of an overall design automation, to create a synthetic gene circuit in Escherichia coli.

THE DNA ASSEMBLY TOOL STACK

Application Layer
- Links assembly graph to protocol library and creates Puppeteer code

Language and Resource Management Layers
- Computes resource requirements
- Produces CHRIS commands that will carry out protocol
- Tracks resources and performs optimized resource allocations

Hardware Abstraction Layer
- Converts CHRIS commands into robot instructions and sends them to the robot platform

Hardware Layer
- Assembles DNA parts

THE DNA ASSEMBLY TOOL STACK

APPLICATIONS LAYER
Assembly Planner → Puppet Show

The assembly planner colors the assembly graph with the names of protocols required to assemble the DNA as designed.

Puppet Show then links these protocols with the actual step-wise procedures stored in a protocol library and converts them to a Puppeteer script.

The step-wise procedures are written as they would be run on a robot.

LANGUAGE AND RESOURCE MANAGEMENT LAYERS
Puppeteer → Resource Manager → CHRIS

Puppeteer is a new language developed in this work to specify these DNA assembly protocols and its parameters specify only biological protocol details and abstract away resources and scheduling.

The language layer combines biological protocol details with resource (reagents, samples, plates, etc.) and scheduling details from the Resource Manager to generate CHRIS (Common Human Robot Instruction Set), which are then sent to HAL (Hardware Abstraction Layer).

FUTURE DIRECTIONS
- Implement protocols on the Tecan robot and expand HAL beyond the Tecan API to increase usage onto many robotic platforms
- Coordinate smart multi-task scheduling on the robotic platform to increase assembly efficiency and high output
- Generate human readable instructions to allow those lacking robotic platforms to take advantage of these new tools

Hardware Abstraction Layer
CHRIS Commands → Robot API

HAL converts CHRIS commands into robot specific instructions. Currently this is only implemented for the Tecan Freedom EVO 150 liquid handling robot. HAL uses the Tecan API calls to execute the protocols on the robot platform.

CURRENT STATUS

The current implementation of this work flow goes from the assembly graph down to running protocols on a simulator for the Tecan Freedom EVO150 robot.

Puppeteer code is generated from the assembly graph in the Applications Layer and is successfully interpreted in the Language Layer and generates CHRIS. The CHRIS command set is still evolving but can be used to run two common laboratory protocols (restriction digest and ligation).

ACKNOWLEDGMENTS

We would like to thank Fusun Yaman from BBN Technologies and Marjan Hadipour, James DiCarlo and Arthur Sun from Boston University for their contributions to this work.

For more information, please visit www.cidarlab.org