

UC Berkeley

UC Berkeley Previously Published Works

Title

Wet Lab Accelerator: A Web-Based Application Democratizing Laboratory Automation for Synthetic Biology

Permalink

<https://escholarship.org/uc/item/4ks1v8sx>

Journal

ACS Synthetic Biology, 6(1)

ISSN

2161-5063

Authors

Bates, Maxwell
Berliner, Aaron J
Lachoff, Joe
et al.

Publication Date

2017-01-20

DOI

10.1021/acssynbio.6b00108

Peer reviewed

Wet Lab Accelerator: A Web-Based Application Democratizing Laboratory Automation for Synthetic Biology

Maxwell Bates,^{†,§} Aaron J. Berliner,^{†,§} Joe Lachoff,[†] Paul R. Jaschke,^{†,‡} and Eli S. Groban^{*,†}

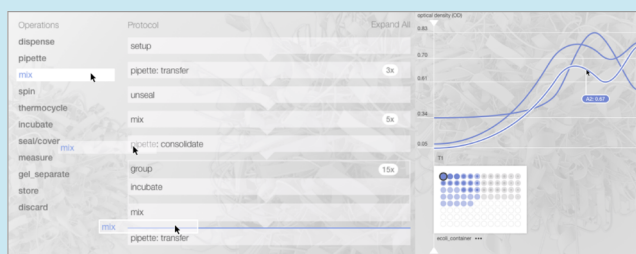
[†]BioNano Research Group, Autodesk Research, San Francisco, California 94111, United States

[‡]Department of Chemistry & Biomolecular Sciences, Macquarie University, Sydney, New South Wales 2109, Australia

S Supporting Information

ABSTRACT: Wet Lab Accelerator (WLA) is a cloud-based tool that allows a scientist to conduct biology via robotic control without the need for any programming knowledge. A drag and drop interface provides a convenient and user-friendly method of generating biological protocols. Graphically developed protocols are turned into programmatic instruction lists required to conduct experiments at the cloud laboratory Transcriptic. Prior to the development of WLA, biologists were required to write in a programming language called “Autoprotocol” in order to work with Transcriptic. WLA relies on a new abstraction layer we call “Omniprotocol” to convert the graphical experimental description into lower level Autoprotocol language, which then directs robots at Transcriptic. While WLA has only been tested at Transcriptic, the conversion of graphically laid out experimental steps into Autoprotocol is generic, allowing extension of WLA into other cloud laboratories in the future. WLA hopes to democratize biology by bringing automation to general biologists.

KEYWORDS: synthetic biology, automation, standardization, cloud-lab, scientific reproducibility, autoprotocol, democratization



Synthetic biology has been described as the introduction of engineering and manufacturing principles into traditional molecular and cell biology.¹ However, the field is poorly documented, its procedures and products lack standards, and high capital and training costs mean that laboratory automation is seldom available to academics or start-up companies.^{2–4} This situation has led to the emergence of cloud-based biological laboratories such as Emerald Cloud Laboratories,⁵ Synthego,⁶ and Transcriptic.⁷ Cloud laboratories offer access to automation platforms and other laboratory tools as a service, thereby enabling the democratization and standardization of biological manufacturing. Typically, the cloud laboratory makes a capital expense investment ranging from \$5,000,000 to \$15,000,000 into laboratory hardware offered as a service.^{8,9} This service model provides standardized hardware operations supplemented by a standard experimental design paradigm. By enabling the robotic automation of microbiological methods, cloud laboratories hope to democratize access to high cost items while improving the standardization of experiments and thus the repeatability of results. A large barrier to entry in the cloud model is the absence of a visual programming interface, which renders these foundries inaccessible to many scientists, particularly those lacking expertise with computer programming languages. We developed Wet Lab Accelerator (WLA, <https://wla.bionano.autodesk.com>) as a framework for democratizing the design, visualization, control, and distribution of standardized, automatable biological laboratory protocols.¹⁰ We believe WLA reduces the high barrier-to-entry for biologists and establishes a foundation to speed adoption of cloud

laboratories while facilitating automated protocol sharing and collaboration. The following Note details WLA’s motivation, architecture, and utility.

■ THE WET LAB ACCELERATOR APPLICATION

Our tool provides a visual method for designing, executing, and sharing fully automated and standardized biological protocols and the resulting experimental data (Figure 1). While we chose to integrate the cloud laboratory Transcriptic into this first version of WLA, since Transcriptic was the most operationally mature, the overall framework of WLA can be applied to any cloud lab.

The design goal of the WLA user interface is to make it easy for users to quickly develop, test, and execute successful robotic wet lab protocols without requiring any background in scripting. Each step in a protocol is accompanied by a visualization of the containers and operations involved (Figure 2). Researchers interact directly with dynamic representations of the source and target wells to define operations, and each container is displayed with a different color to help avoid confusion and reduce errors. Since the flow of operations in the current Autoprotocol standard is always linear, the user interface for WLA shows the protocol as a linear series of operations. These operations can be grouped together. Groups of operations can be looped any number of times. In-line ordering affords easy running and sharing of full experimental

Received: April 8, 2016

Published: August 16, 2016

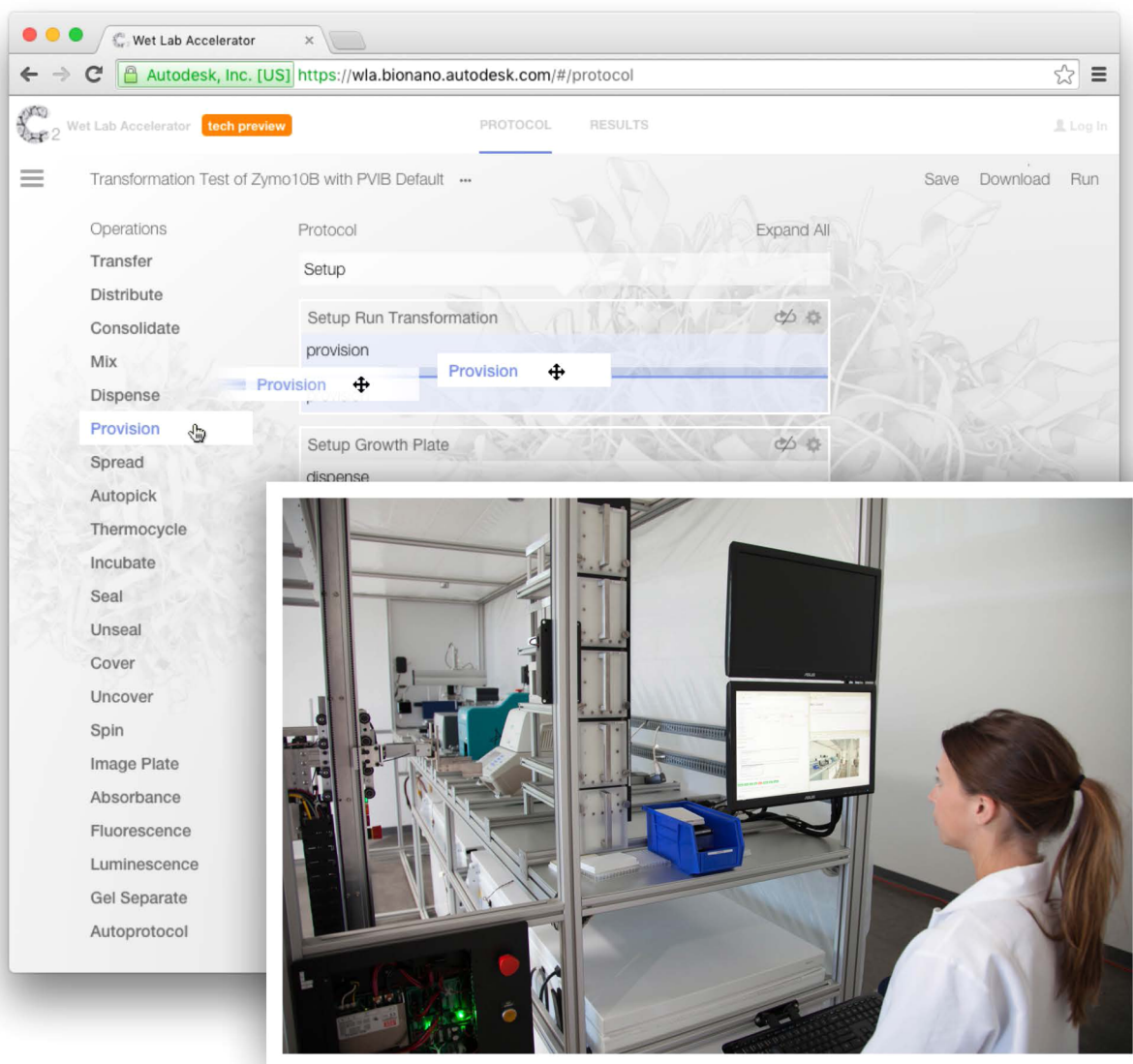


Figure 1. Using Wet Lab Accelerator: The Screenshot of the Wet Lab Accelerator web application in the background illustrates the Autoprotocol operation set and the modern web application environment, and demonstrates WLA's drag-and-drop usability. The photo in the foreground depicts a Transcriptic technician monitoring the status of a protocol generated in WLA and executed at Transcriptic.

pipelines, encompassing design, reagent procurement, execution, and data postprocessing. Integration with lab inventory and ordering ensures efficient use of reagents and supplies while developing a protocol. Prior to running a protocol, any missing information is clearly highlighted, and a cost estimate is provided. Once a run has completed, the software provides a graphical interface for analyzing results with dynamic data plots and results photographs.

Designing laboratory workflows with WLA is as easy as dragging-and-dropping standard laboratory operations into an ordered list called a protocol. A protocol is composed of references and operations. References, shown in the Setup Window of Figure 2 are the set of containers (like a flat 96-well plate, or source materials like Luria Broth or enzymes) that will be used in the protocol. Operations, shown in the Operation Window on the left of Figure 2, are the list of instructions to be performed. Because WLA currently only produces Autoprotocol, a language developed by Transcriptic, its list of available operations is replicated in WLA: transfer, distribute, consol-

idate, mix, dispense, provision, spread, autopick, thermocycle, incubate, (un)seal, (un)cover, spin, image plate, absorbance, fluorescence, luminescence, and gel separate. As shown in Figure 2, a simple protocol can be created from both WLA and python, but the protocol created in WLA is easy to read, understand, and modify without coding experience.

The use of WLA was tested through the design and execution of a simple protocol to transform *Escherichia coli* (*E. coli*) with the standard pUC19 plasmid. The transformation, detailed in Figure S1, provides a proof of concept for WLA. Once a protocol has been designed, submitted, and executed at Transcriptic, WLA can be used to visualize results (Figures S2 and S4). Results data can take the form of optical density from absorbance, light intensity data from fluorescence or luminescence, or a photograph from the "image plate" command. The left side of the results page of an experiment features the executed protocol. Selecting an operation where data has been collected prompts the generation of dynamic visualizations in the browser.

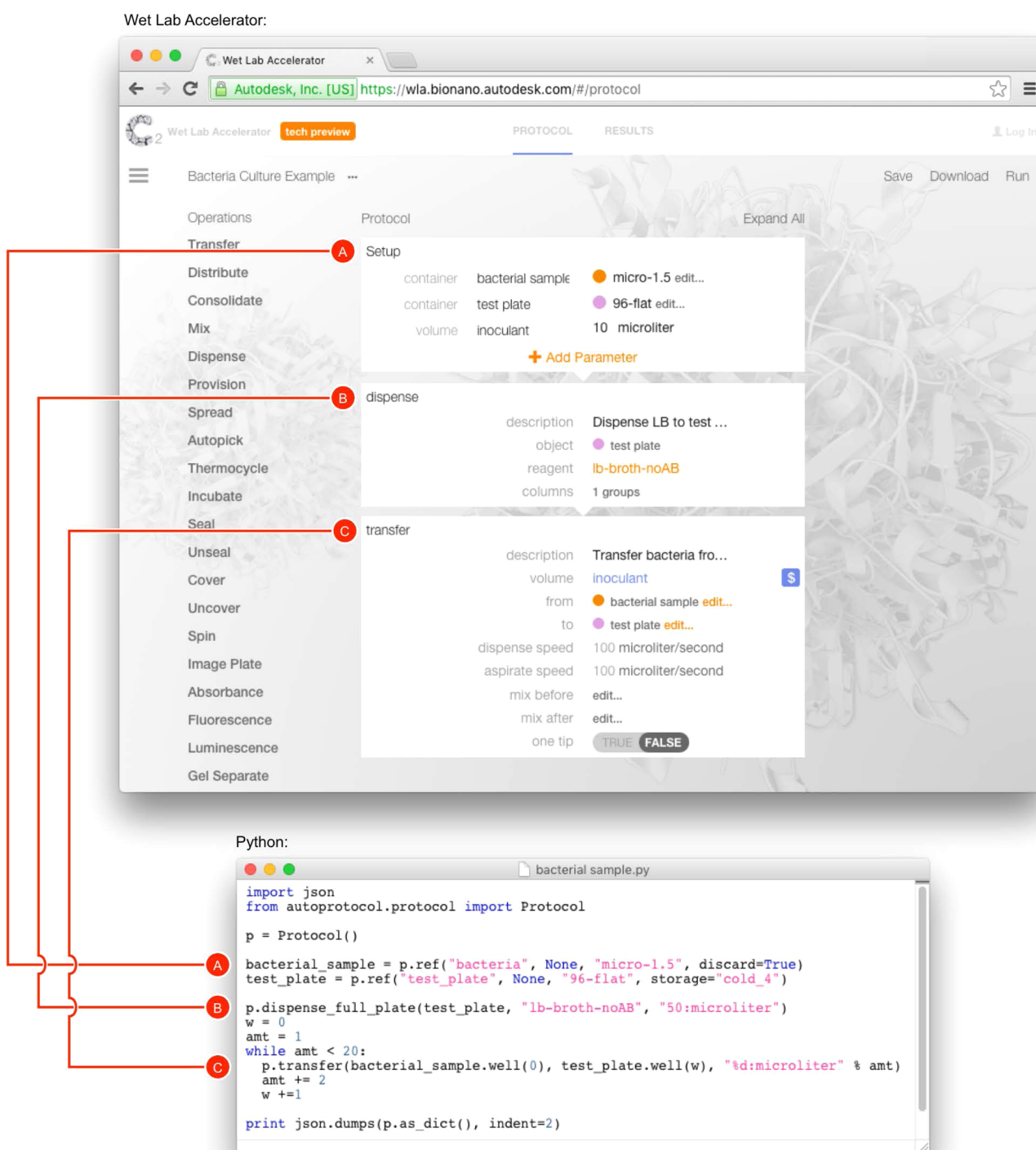


Figure 2. Wet Lab Accelerator vs Scripting: Comparison of generating a protocol using WLA (top) to generating the same protocol in Autoprotocol using python libraries (bottom). WLA requires no coding to accomplish what was done with 13 lines of python.

WLA provides a method for biological operations to be created and shared among users. After a protocol has been created, it can be downloaded as either an Omniprotocol or Autoprotocol JSON file. A downloaded Autoprotocol file can be used in further operations with the Autoprotocol framework provided by Transcriptic. This Autoprotocol can be shared with other scientists, who can modify it as a simple JSON in a text-editor or submit the protocol as-is for execution. A downloaded Omniprotocol file can be stored or shared with other WLA

users. By dragging and dropping the Omniprotocol file onto the setup menu of WLA, a user can recreate the editor environment, edit the protocol, and resubmit the run. Results from a run can be downloaded as a package of JSON or CSV files. The result package can be shared with other users.

■ AUTOPROTOCOL AND OMNIPROTOCOL

The basis for the design of protocols in WLA is Transcriptic's now open standard Autoprotocol. Autoprotocol is a language

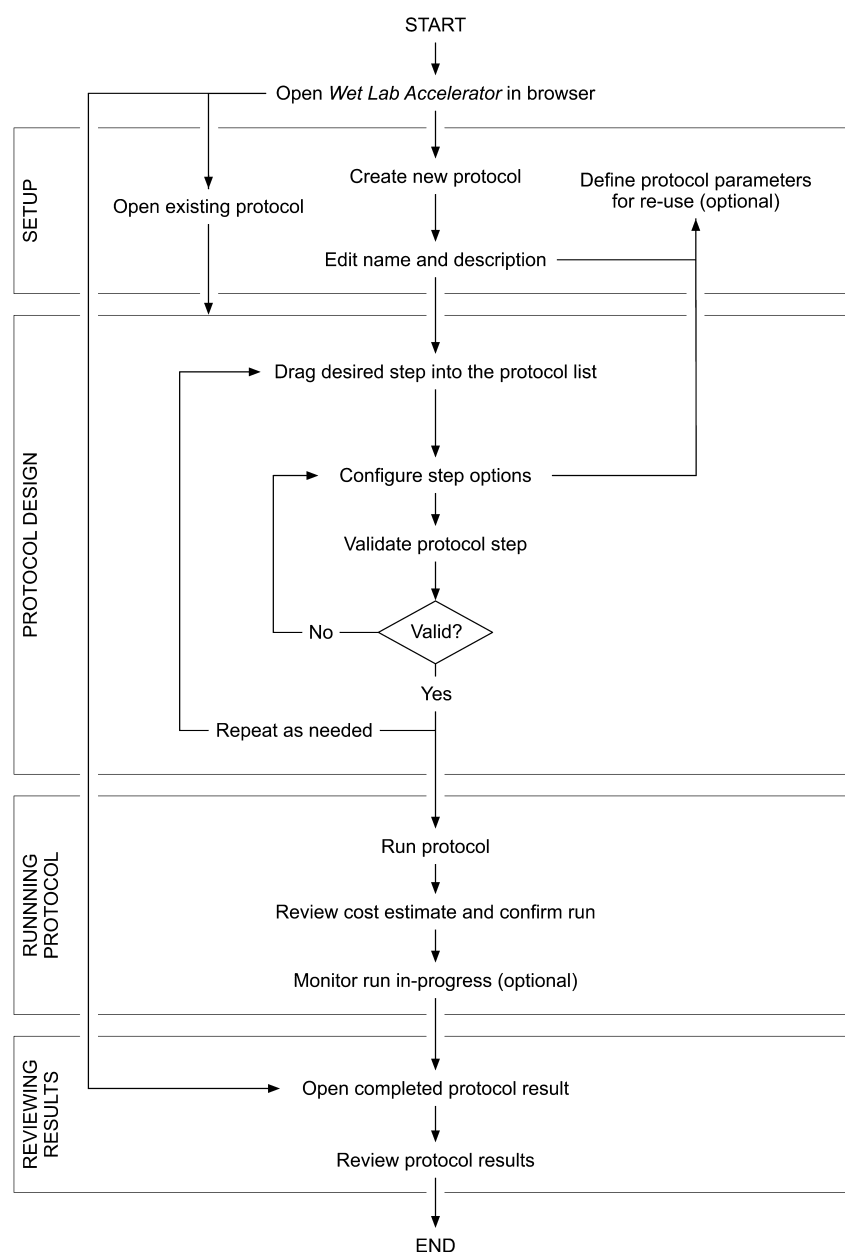


Figure 3. User Journey: This graphic depicts the typical user journey through Wet Lab Accelerator, starting with opening the application in a browser; spanning protocol design and execution; and ending with results browsing.

for specifying experimental protocols for biological research in a way that is precise, unambiguous, and understandable by both biologists and robotic compilers executing on biological automation.¹¹ The design goals of Autoprotocol are flexibility (expansive protocol possibilities from a small set of operations), composition (protocols can be built on top of each other), easy execution (protocols can be mapped directly to hardware), platform independence (protocols can be mapped beyond Transcriptic), and understandability (the operations that compose a protocol can be easily understood). The standard way to generate Autoprotocol is to either manually compose a JSON file or use a python library to create and export such a file. In order to provide a modern, web-based visual programming alternative to python scripting, we developed the Autoprotocol-inspired schema Omniprotocol. Autoprotocol is a high-level definition describing laboratory commands which can be compiled to machine instructions. In contrast,

Omniprotocol is a higher-level definition which provides support for scripting language features such as looping and the use of variables. Omniprotocol was originally created to simplify web application development, and has evolved as a means for abstracting Autoprotocol for graphical-user interaction when designing and visualizing a protocol, and abstracts the application from dependence on Autoprotocol. WLA's use of Omniprotocol provides the front-end for user-interaction as shown in Figure 2. The interaction between WLA, Omniprotocol, and Autoprotocol is described in Figure S5.

■ FUTURE WORK

Wet Lab Accelerator is open-source, and freely available for use and contribution. Transcriptic is the first system to use the Autoprotocol specification, so WLA executes on this cloud laboratory as a default. WLA has been designed to easily integrate with future cloud laboratories using Autoprotocol as

they become available. Because Transcriptic is constantly evolving both their hardware and Autoprotocol language, potential inconsistencies may develop over time. Though we have demonstrated that WLA is capable of sharing protocols through an upload/download of Omniprotocol files, in the future we hope to provide a more streamlined in-app sharing process among users for both protocols and experimental data. The current WLA user interface reflects many compromises that are typical in an application's first release. The user journey shown in Figure 3 will evolve as we learn more from talking with and observing researchers using the software. Planned feature improvements include better handling of inventory, more dynamic data visualization, and a stronger visual relationship between a protocol and the corresponding experimental results. Additional development is also required for connecting WLA to other tools in order to form more complex workflows.

■ ASSOCIATED CONTENT

● Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: [10.1021/acssynbio.6b00108](https://doi.org/10.1021/acssynbio.6b00108).

Additional figures and captions detailing the setup of the example protocol, results visualization, and the extended application architecture (ZIP)

JSON and CSV records of the experimental Transcriptic runs described in SI-Figures-and-Captions.zip. See SI-Data README.md for details (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: eli.groban@autodesk.com.

Author Contributions

M.B. and A.B. contributed equally to this work. M.B. and A.B. designed WLA in collaboration. M.B. designed the WLA software. A.B. tested the system in the initial use cases. J.L. designed the user interaction and user experience. P.J. assisted A.B. with further testing. M.B., A.B., J.L., P.J., and E.G. wrote the paper.

Author Contributions

[§]M.B. and A.B. contributed equally to this work.

Notes

The authors declare the following competing financial interest(s): The corresponding author and several current or former Autodesk employees who are co-authors own Autodesk stock.

■ ACKNOWLEDGMENTS

This work was funded entirely by Autodesk. The authors thank Tali Herzka, Jeremy Apthorp, Conny Scheitz, Max Hodak, and the Transcriptic Team, as well as Peter Jones, Dorothy Silverman, Dion Whitehead Amago, Chris Venter, Florencio Mazzoldi, Andrew Hessel, Larry Peck, Jeff Kowalski, and the BioNano Research Group at Autodesk.

■ NOMENCLATURE

WLA, Wet Lab Accelerator; JSON, JavaScript Object Notation; CSV, Comma Separated Values; UI, User Interface

■ REFERENCES

- (1) Brophy, J. A. N., and Voigt, C. A. (2014) Principles of genetic circuit design. *Nat. Methods* 11, 508–520.
- (2) Cardinale, S., and Arkin, A. P. (2012) Contextualizing context for synthetic biology -identifying causes of failure of synthetic biological systems. *Biotechnol. J.* 7, 856–866.
- (3) 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., and Endy, D. (2013) Precise and reliable gene expression via standard transcription and translation initiation elements. *Nat. Methods* 10, 354–360.
- (4) Begley, C. G., and Ellis, L. M. (2012) Drug development: Raise standards for preclinical cancer research. *Nature* 483, 531–533.
- (5) *Emerald Cloud Lab: A web-based life sciences lab*, developed by scientists for scientists; 2016; <http://emeraldcloudlab.com/>.
- (6) *CRISPRvolution: Synthetic Guide RNA for CRISPR Genome Editing*; 2016; <https://www.synthego.com/>.
- (7) *transcriptic: YOUR LAB IN THE CLOUD*; 2016; <https://www.transcriptic.com/>.
- (8) Personal communication with Dr. Patrick Cai, University of Edinburgh. Private communication, 2016.
- (9) Personal conversations with various companies. Private communication, 2016.
- (10) *Autodesk/bionano-wetLabAccelerator: A tool for researchers working in synthetic biology and virology to design robotic wet lab protocols using a visual UI without coding*; 2016; The code is open source and the repo can be found at <https://github.com/Autodesk/bionano-wetLabAccelerator>.
- (11) *AUTOPROTOCOL.org: An open standard for life science experimental design and automation*; 2016; <http://autoprotocol.org/>.