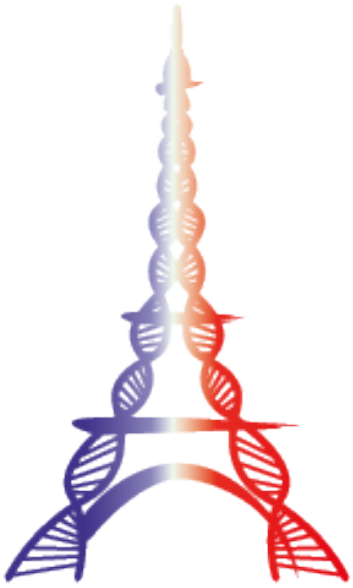


IWBDA 2022

October 2022 Paris, France

14th International Workshop on Bio-Design Automation
Paris, France
October 24th-26th 2022

Foreword

Welcome to IWBDA 2022!

The IWBDA 2022 Organizing Committee welcomes you to the Fourteenth International Workshop on Bio-Design Automation (IWBDA). The Fourteenth International Workshop on Bio-Design Automation (IWBDA) brings together researchers from the synthetic biology, systems biology, and design automation communities to discuss concepts, methodologies and software tools for the computational analysis and synthesis of biological systems.

The field of synthetic biology, still in its early stages, has largely been driven by experimental expertise, and much of its success can be attributed to the skill of the researchers in specific domains of biology. There has been a concerted effort to assemble repositories of standardized components; however, creating and integrating synthetic components remains an ad hoc process. Inspired by these challenges, the field has seen a proliferation of efforts to create computer-aided design tools addressing synthetic biology's specific design needs, many drawing on prior expertise from the electronic design automation (EDA) community.

The IWBDA offers a forum for cross-disciplinary discussion, with the aim of seeding and fostering collaboration between the biological and the design automation research communities.
This year, the program consists of 6 workshops, 16 contributed talks, and 20 lightning talks for posters: The talks are organized into 4 sessions:

- Biofoundries and Automation
- Modeling
- Measurement
- Software and Pipelines

In addition, we have a distinguished invited speaker, Dr. Gregory Batt from Inria for a keynote seminar. We would like to thank all the participants for their contributions to IWBDA. We would also like to highlight the efforts of the Program Committee and the Student Volunteers.

IWBDA is proudly organized by the non-profit Bio-Design Automation Consortium (BDAC). BDAC is an officially recognized 501(c)(3) tax-exempt organization.

Sponsors

Organizing Committee

Organizing Committee

General Chair - Alexis Casas, Imperial College London

Program Committee Chair - Alejandro Vignoni, Universitat Politècnica de València

Publication Chair - Ayush Pandey, California Institute of Technology

Local Chair - Olivier Borkowski, INRAE, Université Paris-Saclay

Student Volunteer Chair - Zoé Pincemaille, UTC Compiègne

Co-Web Chair - Aaron Adler, BBN Technologies

Co-Web Chair - Prashant Vaidyanathan, Microsoft Research

Finance Chair - Traci Haddock-Angelli, Asimov

Bio-Design Automation Consortium

President - Aaron Adler, BBN Technologies

Vice-President - Natasa Miskov-Zivanov, University of Pittsburgh

Treasurer - Traci Haddock, Asimov

Board Member - Douglas Densmore, Boston University

Program Committee

Aaron Adler	BBN Technologies
Jacob Beal	BBN Technologies
Lukas Buecherl	CU Boulder
Alexis Casas	Imperial College London
William Poole	California Institute of Technology
Prashant Vaidyanathan	Microsoft Research

Pablo Carbonell	Universitat Politècnica de València
Fernando Nóbel Santos Navarro	Universitat Politècnica de València
Daniel Schindler	Max Planck Institute for Terrestrial Microbiology
Gonzalo Vidal	Newcastle University
Bryan Bartley	BBN Technologies
Jose Luis Navarro	Universitat Politècnica de València
Ayush Pandey	California Institute of Technology
Sonja Billerbeck	The University of Groningen
Vinoo Selvarajah	iGEM Foundation
Gael Chambonnier	Massachusetts Institute of Technology
Yadira Boada	Universitat Politècnica de València
Thomas Gorochowski	University of Bristol
Eszter Csibra	Imperial College London
Jonathan Tellechea	Universitat Politècnica de València
Jonathan Tellechea Luzardo	Universitat Politècnica de València
Olivier Borkowski	Inria, France
Ángel Goñi-Moreno	Technical University of Madrid
Giansimone Perrino	Imperial College London
Gilberto Reynoso-Meza	Pontifícia Universidade Católica do Paraná (PUCPR)
Daniel Georgiev	University of West Bohemia
Zoila Jurado	California Institute of Technology
Sara Napolitano	Institut Pasteur
Eran Agmon	University of Connecticut

Program

All times CEST (UTC + 1) Paris local time.

Monday, 24th October 2022 (at Learning Planet Institute)

09:30 – 10:30 **Registration**

10:30 – 10:35 **Welcome & Opening Remarks**

10:35 – 11:45 **Workshop 2: SBOL Version 3: Data Exchange throughout the Bioengineering Lifecycle**

Tom Mitchell, Jacob Beal and Bryan Bartley.

11:45 – 12:00 **Short Break**

12:00 – 13:00 **Workshop 2 (continued)**

13:00 – 14:30 **Lunch on your own**

14:30 – 15:30 **Workshop 3: Automating Laboratory Protocols with the Laboratory Open Protocol (LabOP) language**

Bryan Bartley, Jacob Beal and Dan Bryce.

15:30 – 15:45 **Short Break**

15:45 – 16:45 **Workshop 3 (continued)**

16:45 – 17:00 **Short Break**

17:00 – 17:30 **Lightning Talks 1 (90 seconds presentation of each one of the posters)**

17:30 – 18:30 **Posters:**

(1) *Steps Towards Functional Synthetic Biology.* Ibrahim Aldulijan, Jacob Beal, Sonja Billerbeck, Jeff Bouffard, Gaël Chambonnier, Nikolaos Delkis, Isaac Guerreiro, Martin Holub Martin Holub, Daisuke Kiga, Jacky Loo, Paul Ross, Vinoo Selvarajah, Noah Sprent, Gonzalo Vidal and Alejandro Vignoni

(2) *Adapting Malware Detection to DNA Screening.* Dan Wyschogrod, Jeff Manthey, Tom Mitchell, Steven Murphy, Adam Clore and Jacob Beal

(3) *Artificial Metabolic Networks: enabling neural computations with metabolic networks.* Léon Faure and Jean-Loup Faulon

(4) *Developing a scoring system to optimise the design of CRISPR Cas12 diagnostics.* Akashaditya Das and Ana Pascual-Garrigos

(5) *DBTL bioengineering cycle: developing a population oscillator.* Andrés Arboleda-García, Iván Alarcon-Ruiz, Yadira Boada, Jesús Picó and Eloisa Jantus-Lewintre

(6) *Computer-aided enhancement of genetic design data.* Matthew Crowther and Angel Goñi-Moreno

(7) *Exploring Advantages and Limitations of Discrete Modeling of Biological Network Motifs.* Difei Tang, Gaoxiang Zhou and Natasa Miskov-Zivanov

(8) *SynPath – An Automated Biosynthetic Pathway Design and Analysis Tool.* Carol Gao, Helena van Tol and Xi Wang

(9) *The Context Matrix: A Framework for Context-Aware Synthetic Biology.* Camillo Moschner, Charlie Wedd and Somenath Bakshi

(10) *An Interactive Microfluidic Design and Control Workflow.* Yangruirui Zhou and Douglas Densmore

(11) *Dynamic Behavior Alters Influences and Sensitivities in Biological Networks.* Gaoxiang Zhou and Natasa Miskov-Zivanov

(12) *Experimental Data Converter.* Sai Samineni, Gonzalo Vidal, Jeanet Mante, Guillermo Yañez-Feliú, Carlos Vidal-Céspedes, Chris Myers and Timothy J. Rudge

(13) *Expanding the metaheuristic framework: evolving cells with the bat algorithm.* Víctor Reyes, Nicolás Hidalgo and Martín Gutiérrez

(14) *PLATERO: A Plate Reader Calibration Protocol to work with different instrument gains and asses measurement uncertainty.* Yadira Boada, Alba González-Cebrián, Joan Borràs-Ferrís, Jesús Picó, Alberto Ferrer and Alejandro Vignoni

(15) *Spatially Solving the Graph Coloring Problem Using Intercell Communication.* Daniela Moreno, Diego Araya and Martín Gutiérrez

(16) *A comparison between D-optimal and model-based design of experiments for efficient biomanufacturing.* Iván Blázquez Arenas, Pablo Carbonell and Irene Otero-Muras

(17) *Probabilistic programming for synthetic gene networks.* Lewis Grozinger and Angel Goñi-Moreno

(18) *In-silico design for fold-change detection (FCD) synthetic circuits.* Rongying Huang and Ramez Daniel

(19) *Rule-based generation of synthetic genetic circuits.* Daisuke Kiga, Kazuteru Miyazaki, Shoya Yasuda, Ritsuki Hamada, Sota Okuda, Ryoji Sekine, Naoki Kodama and Masayuki Yamamura

(20) *Standardizing the Representation of Parts and Devices for Build Planning.* Jacob Beal, Vinoo Selvarajah, Gael Chambonnier, Traci Haddock-Angelli, Alejandro Vignoni, Gonzalo Vidal and Nicholas Roehner

Tuesday, 25th October 2022 (at Learning Planet Institute)

08:00 – 09:00 **Registration**

09:00 – 10:00 **Workshop 4: Principles of genetic circuit design**

Hatem Abdelrahman.

10:00 – 10:30 **Demo: DySE: Dynamic System Explanation framework**

Gaoxiang Zhou, Difei Tang and Natasa Miskov-Zivanov.

10:30 – 10:45 **Short Break**

10:45 – 11:45 **Workshop 5: Python Tools for Modeling of Biocircuits From High-Level Specification to Parameter Inference**

Ayush Pandey and Zoltan Tuza

11:45 – 12:00 **Short Break**

12:00 – 13:00 **Workshop 5: (continued)**

13:00 – 14h:0 **Lunch on your own**

14:30 – 15:30 **Workshop 6: OneModel: an easy-to-use tool for modular building of biocircuits models (Part 1)**

Fernando N. Santos-Navarro.

15:30 – 15:45 **Short Break**

15:45 – 16:45 **Workshop 6: OneModel: an easy-to-use tool for modular building of biocircuits models (Part 2)**

Fernando N. Santos-Navarro.

16:45 – 17:00 **Short Break**

17:00 – 18:00 **Keynote: Adding automation and reactiveness to your experiments: motivation, tools and applications.**

Gregory Batt

Wednesday, 26th October 2022 (at iGEM Paris Expo – Porte de Versailles)

09:30 – 09:45 **Welcome & Opening Remarks**

09:45 – 10:00 **Sponsored Talk: Asimov – Intelligent design of living systems to enable next-generation biotechnologies**

Traci Haddock

10:00 – 10:30 **Lightning talks (90 seconds presentation of each one of the posters)**

10:30 – 10:45 **Short Break**

10:45 – 11:45 **Talks Session 1: Biofoundries and Automation, Chair: Olivier Borkowski**

10:45 – 11:00 *Efficient Droplet Microfluidic Characterization for Design Automation*

Diana Arguijo and Douglas Densmore

11:00 – 11:15 *Low-cost Open Source Benchtop Bioreactor*

Vitor Marchesan, Lívia Galinari, Luiza Possa, Tiago Mendes, João Vitor Molino and Livia Ferreira-Camargo

11:15–11:30 *Harnessing Biofoundries for the forward engineering of strains, with a focus on increased cis, cismuconic acid titers in yeast*

Kealan Exley, Zofia Dorota Jarczynska, Linas Tamošaitis and Vijayalakshmi Kandasamy

11:30–11:45 *The iBioFoundry: Automated, Low-Cost, High-Throughput Experimentation*

Camillo Moschner, Charlie Wedd, Georgeos Hardo and Somenath Bakshi

11:45 – 12:00 **Short Break**

12:00 – 13:00 **Talks Session 2: Modeling, Chair: Alejandro Vignoni**

12:00 – 12:15 *Model-driven analysis and debugging of synthetic logic circuits with new CRISPRi components*

Davide De Marchi, Roman Shaposhnikov, Paolo Magni and Lorenzo Pasotti

12:15 – 12:30 *From Specification to Implementation: Assume-Guarantee Contracts for Synthetic Biology*

Ayush Pandey, Inigo Incer, Alberto Sangiovanni Vincentelli and Richard M Murray

12:30 – 12:45 *A Bounded Model Checking Framework for the Analysis of Chemical Reaction Network Models*

Mohammad Ahmadi, Lukas Buecherl, Zhen Zhang, Chris Myers, Chris Winstead and Hao Zheng

12:45 – 13:00 *Characterization of integrase and excisionase activity in cell-free protein expression system using a modeling and analysis pipeline*

Ayush Pandey, Makena L Rodriguez, William Poole and Richard M Murray

14:30 – 15:30 **Talks Session 3: Measurement, Chair: Zoé Pincemaille**

14:30 – 14:45 *FPCountR: improved analytical methods enable absolute protein quantification*

Eszter Csibra and Guy-Bart Stan

14:45 – 15:00 *magmiX – An automated magnetic bio-separator for sustainable biomedical research*

Christoph Sadee, Julian Alexander Zagalak, George Konstantinou and Jernej Ule

15:00 – 15:15 *Towards an automated assay for the quantification of secreted proteins*

Sara Napolitano, Sebastián Sosa Carrillo, François Bertaux, Hélène Philippe and Gregory Batt

15:15 – 15:30 *Rapid gene circuits prototyping with JUMP assembly*

Rizki Mardian, Marcos Valenzuela-Ortega, Jin Wong and Christopher French

15:30 – 15:45 **Short Break**

15:45 – 16:45 **Panel: Young Biofoundries**

Daniel Schindler (MaxGENESYS), François Bertaux (Lesaffre), Stéphane Lemaire (Sorbonne Université)

16:45 – 17:00 **Short Break**

17:00 – 18:00 **Talks Session 4: Software and Pipelines, Chair: Ayush Pandey**

17:00 – 17:15 *GUARDIAN: Ensemble Detection of Engineering Signatures*

Aaron Adler, Joel Bader, Brian Basnight, Jitong Cai, Elizabeth Cho, Joseph Collins, Yuchen Ge, John Grothendieck, Kevin Keating, Tyler Marshall, Anton Persikov, Helen Scott, Roy Siegelmann, Mona Singh, Allison Taggart, Benjamin Toll, Daniel Wyschogrod, Fusun Yaman, Eric Young and Nicholas Roehner

17:15 – 17:30 *SIMPLIFE: An automated pipeline for inserting functional domains into globular proteins*

Georgie Hau Sorensen, Fabio Parmeggiani and Thomas Gorochowski

17:30 – 17:45 *Galaxy-SynBioCAD: Automated Pipeline for Industrial Biotechnology*

Joan Hérisson, Thomas Duigou, Kenza Bazi-Kabbaj, Mahnaz Sabeti Azad, Manish Kushwaha and Jean-Loup Faulon

17:45 – 18:00 *Implementing Cross-Platform Protocol Execution with the Laboratory Open Protocol language*

Bryan Bartley, Jacob Beal, Daniel Bryce, Alexis Casas, Jeremy Cahill, Timothy Fallon, Robert Goldman, Luiza Hesketh, Tim Dobbs and Alejandro Vignoni

18:00 – 18:15 **Closing Remarks**

Keynote Presentation

Adding automation and reactiveness to your experiments: motivation, tools and applications

Gregory Batt

Speaker Biography

Dr. Gregory Batt is a senior research scientist at Inria and the head of the InBio group at Institut Pasteur. He studied molecular and cellular biology and computer science at the Ecole Normale Supérieure de Lyon. He received his PhD in computer science from the University of Grenoble in 2006. Prior to joining Inria in 2007, he was a postdoctoral researcher at Boston University. Since 2017, he leads the InBio team, an Inria/Institut Pasteur research group. The InBio team is interested in understanding, controlling and optimizing cellular processes from the single cell to the cell population levels. InBio members combine wet and dry biology in the same lab. They employ systems and synthetic biology approaches with control and active learning methods, together with stochastic and statistical modeling frameworks. They also develop affordable bioreactor-based platforms with automated measurements and reactive experiment control. In recent applications, they have notably designed an artificial differentiation system in yeast and used it to create consortia with tuneable composition, and have characterized protein secretion under various stress conditions to optimize production in yeast.

Keynote Abstract

Small-scale, low-cost bioreactors are emerging as powerful tools for microbial systems and synthetic biology research. They allow tight control of cell culture parameters over long durations. These unique features enable researchers to perform sophisticated experiments and to achieve high reproducibility. However, existing setups are limited in their measurement capabilities. It is often essential to follow over time key characteristics of the cultured cell population, such as gene expression levels, cellular stress levels, and cell size and morphology. Researchers usually need to manually extract, process and measure culture samples to run them through sensitive and specialized instruments. Manual interventions strongly constrains the available temporal resolution and reactiveness capabilities. In this talk, I will present ReacSight, a generic and flexible strategy to enhance bioreactor arrays with automated measurements capabilities and reactive experiment control. It can also be used to enhance any computer-controlled plate-based measurement device with pipetting capabilities and automation. ReacSight leverages the affordable Opentrons pipetting robots. It is ideally suited to integrate open-source, open-hardware components but can also accommodate closed-source, GUI-only components. Applications include the control of an artificial differentiation system in yeast to create consortia with tuneable composition,

and the characterization of protein secretion under various stress conditions to optimize production in yeast. In the same spirit, we have also developed MicroMator, a software tool to streamline the use of Micromanager and enable the realization of smart, reactive microscopy experiments. MicroMator also fosters throughput, reproducibility and reactivity.

Contributed Talks

The following 16 abstracts feature as contributed talks in this year's IWBDA program:

Talks Session 1: Biofoundries and Automation

(1) *Efficient Droplet Microfluidic Characterization for Design Automation.* Diana Arguijo and Douglas Densmore
(2) *Low-cost Open Source Benchtop Bioreactor.* Vitor Marchesan, Lívia Galinari, Luiza Possa, Tiago Mendes, João Vitor Molino and Livia Ferreira-Camargo
(3) *Harnessing Biofoundries for the forward engineering of strains, with a focus on increased cis, cismuconic acid titers in yeast.* Kealan Exley, Zofia Dorota Jarczynska, Linas Tamošaitis and Vijayalakshmi Kandasamy
(4) *The iBioFoundry: Automated, Low-Cost, High-Throughput Experimentation.* Camillo Moschner, Charlie Wedd, Georgeos Hardo and Somenath Bakshi

Talks Session 2: Modeling

(1) *Model-driven analysis and debugging of synthetic logic circuits with new CRISPRi components.* Davide De Marchi, Roman Shaposhnikov, Paolo Magni and Lorenzo Pasotti
(2) *From Specification to Implementation: Assume-Guarantee Contracts for Synthetic Biology.* Ayush Pandey, Inigo Incer, Alberto Sangiovanni Vincentelli and Richard M Murray
(3) *A Bounded Model Checking Framework for the Analysis of Chemical Reaction Network Models.* Mohammad Ahmadi, Lukas Buecherl, Zhen Zhang, Chris Myers, Chris Winstead and Hao Zheng
(4) *Characterization of integrase and excisionase activity in cell-free protein expression system using a modeling and analysis pipeline.* Ayush Pandey, Makena L Rodriguez, William Poole and Richard M Murray

Talks Session 3: Measurement

(1) *FPCountR: improved analytical methods enable absolute protein quantification.* Eszter Csibra and Guy-Bart Stan
(2) *magmiX – An automated magnetic bio-separator for sustainable biomedical research.* Christoph Sadee, Julian Alexander Zagalak, George Konstantinou and Jernej Ule
(3) *Towards an automated assay for the quantification of secreted proteins.* Sara Napolitano, Sebastián Sosa Carrillo, François Bertaux, Hélène Philippe and Gregory Batt
(4) *Rapid gene circuits prototyping with JUMP assembly.* Rizki Mardian, Marcos Valenzuela-Ortega, Jin Wong and Christopher French

Talks Session 4: Software and Pipelines

(1) *GUARDIAN: Ensemble Detection of Engineering Signatures.* Aaron Adler, Joel Bader, Brian Basnight, Jitong Cai, Elizabeth Cho, Joseph Collins, Yuchen Ge, John Grothendieck, Kevin Keating, Tyler Marshall, Anton Persikov, Helen Scott, Roy Siegelmann, Mona Singh, Allison Taggart, Benjamin Toll, Daniel Wyschogrod, Fusun Yaman, Eric Young and Nicholas Roehner
(2) *SIMPLIFE: An automated pipeline for inserting functional domains into globular proteins.* Georgie Hau Sorensen, Fabio Parmeggiani and Thomas Gorochowski
(3) *Galaxy-SynBioCAD: Automated Pipeline for Industrial Biotechnology.* Joan Hérisson, Thomas Duigou, Kenza Bazi-Kabbaj, Mahnaz Sabeti Azad, Manish Kushwaha and Jean-Loup Faulon
(4) *Implementing Cross-Platform Protocol Execution with the Laboratory Open Protocol language.* Bryan Bartley, Jacob Beal, Daniel Bryce, Alexis Casas, Jeremy Cahill, Timothy Fallon, Robert Goldman, Luiza Hesketh, Tim Dobbs and Alejandro Vignoni

Efficient Droplet Microfluidic Characterization for Design Automation

Diana Arguijo
dma25@bu.edu
Department of Biomedical Engineering, Boston University
Boston, MA

Douglas Densmore
dougd@bu.edu
Department of Electrical Engineering, Boston University
Boston, MA

1 INTRODUCTION

Droplet microfluidics provides a tool for the acceleration of synthetic biology research by increasing screening throughput and reproducibility [1]. Precise biological screening utilizing small reagent volumes requires equally precise designs that meet the intended performance of the user. One method for implementing microfluidic devices with exact performance is to design, fabricate, and test devices that iterate over the various geometric microfluidic parameters [3]. Without expert knowledge, this method is resource and time intensive, which increases the barrier to entry for first-time users. To address this challenge, machine learning is used for droplet microfluidic design automation to efficiently map performance metrics to specific designs [2]. However, large data sets are required to initialize or train a model. Here, we propose a workflow for microfluidic device design automation that combines active machine learning, camera-free droplet monitoring, and automatic device characterization. Specifically, through the use of impedance sensing, droplets in a microfluidic device can be monitored automatically.

2 PERFORMANCE CHARACTERIZATION

Droplet generators have multiple geometric parameters that can be varied during the design process, as shown in Figure 1A. Each droplet generation parameter affects the microfluidic performance metrics such as droplet size and generation rate [3]. Additionally, the impedance sensing parameters impact the impedance signal profile.

Electrodes embedded in a microfluidic device can be used to measure the impedance of droplets in oil [5]. As a droplet passes through the sensing area between the electrodes, there is a peak in the impedance signal due to the capacitance difference between water and oil, as shown in Figure 1B. Using peak detection, the theoretical impedance data is analyzed to determine the droplet generation rate. A second microfluidic performance metric, droplet diameter (d) , is calculated using Equation 1. Here, Q_w is the volumetric flow rate for water and F is the droplet generation rate.

$$d = 2\left(\frac{3Q_w}{4F\pi}\right)^{1/3} \tag{1}$$

Under certain flow conditions, the size distribution of droplets is either monodisperse or polydisperse [6]. By analyzing the impedance profile under certain flow rates, the data can be sorted and the specific conditions that yield a monodisperse size distribution can be recorded for microfluidic device characterization (Figure 1B).

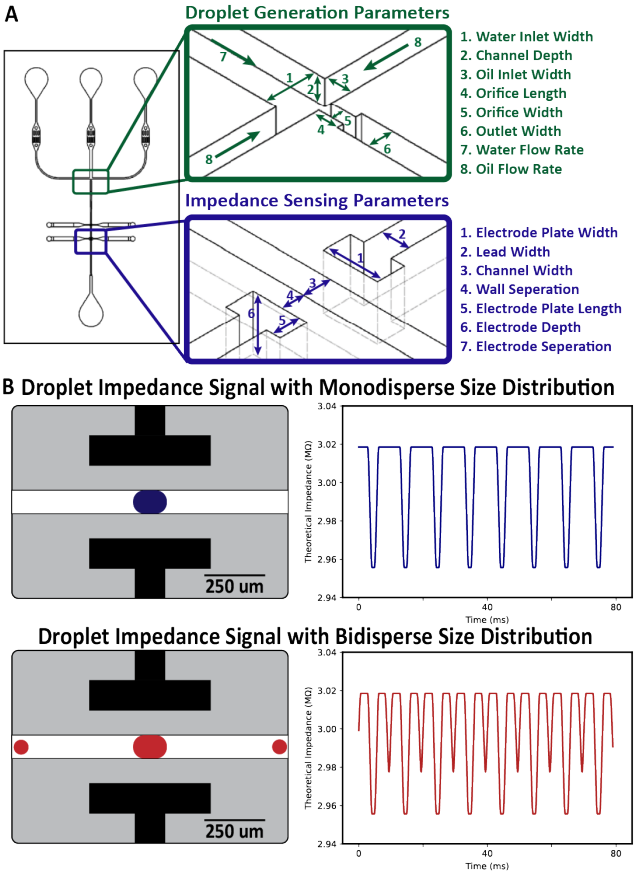


Figure 1: Droplet generation and impedance sensing. A) Geometric and flow parameters for a droplet generator and impedance sensing electrodes. B) Droplets in the impedance sensing area and the theoretical impedance signal droplets with a monodisperse (top) and bidisperse (bottom) size distribution.

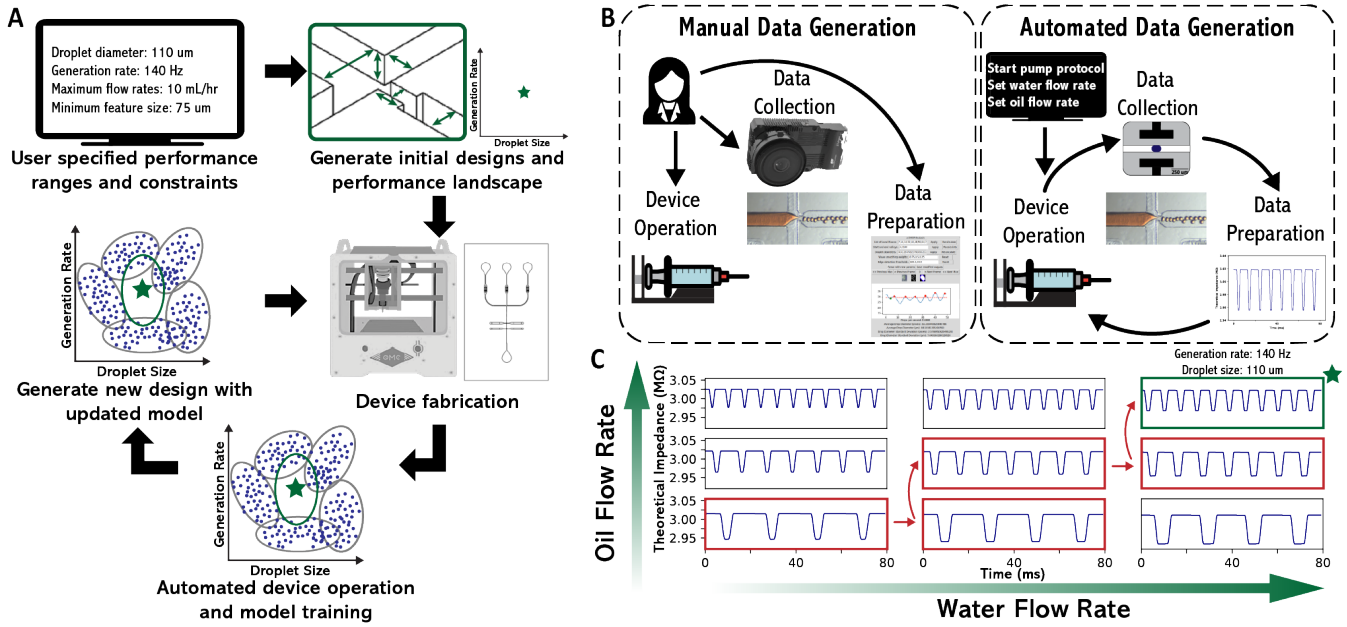


Figure 2: Droplet generation characterization for design automation. A) Active learning design automation workflow. B) Comparison of manual and automated data generation. C) Data set generation by varying oil and water flow rates.

3 MICROFLUIDIC DESIGN AUTOMATION

Existing machine learning models are constrained to droplet generation using fluids with specific properties. To train a model on different fluids large data sets are required. One solution is to utilize active learning to train a model while limiting the number of microfluidic devices designed, fabricated, and characterized [4].

As shown in Figure 2A, the proposed algorithm starts with the user specifying performance metrics and constraints. The machine learning algorithm will identify several initial designs to seed the model [4]. After these devices are fabricated using rapid prototyping [3], the model will be trained on the data generated from the automated device operation. Once the model has been trained on the small data set, a new design is generated with a predicted performance that approaches the target performance. With each round of design generation and characterization, the model accuracy improves [4].

To efficiently generate the data for training the model, this workflow utilizes automated device operation and data collection, Figure 2B. Instead of the microfluidic user manually varying the flow rates to generate each data point for the model, the automation protocol connects to the syringe pumps and changes the flow rates automatically. Additionally, the proposed protocol will connect to the impedance sensors to collect and analyze the impedance data in real-time, replacing the need for a high-speed camera and video processing to extract the droplet generation rate and size.

4 CONCLUSION AND FUTURE WORK

Through the combination of active learning, impedance sensing of droplets, and automatic device characterization, the workflow proposed in Figure 2 will enable efficient data generation for microfluidic device design automation. Integrating and testing a robust impedance sensor with fast sampling rates is required to implement the framework described here with droplet generation rates in the kHz range. Additionally, the workflow can be extended to include other sensor types to monitor droplets.

REFERENCES

[1] Gach, P. C., Iwai, K., Kim, P. W., Hillson, N. J., and Singh, A. K. Droplet microfluidics for synthetic biology. *Lab on a Chip* (2017).

[2] Lashkaripour, A., Rodriguez, C., Mehdipour, N., Mardian, R., McIntyre, D., Ortiz, L., Campbell, J., and Densmore, D. f. d. g. Machine learning enables design automation of microfluidic flow-focusing droplet generation. *Nature Communications* (2021).

[3] Lashkaripour, A., Silva, R., and Densmore, D. Desktop micromilled microfluidics. *Microfluidics and Nanofluidics* (2017).

[4] McIntyre, D., Lashkaripour, A., and Densmore, D. Active learning for efficient microfluidic design automation. *International Workshop on Bio-Design Automation (IWBDA)* (2020).

[5] McIntyre, D., Lashkaripour, A., and Densmore, D. Rapid and inexpensive microfluidic electrode integration with conductive ink. *Lab on a Chip* (2020).

[6] Rosenfeld, L., Lin, T., Derda, R., and Tang, S. Review and analysis of performance metrics of droplet microfluidics systems. *Microfluidics and Nanofluidics* (2014).

Low-cost Open Source Benchtop Bioreactor

Vitor Frost Marchesan[1], Lívia Batista Galinari[2], Luiza de Oliveira Possa[2],Tiago Antônio de Oliveira Mendes [2], João Vitor Dutra Molino[3], Livia Seno Ferreira-Camargo[1]
[1]Universidade Federal do ABC, [2]Universidade Federal de Viçosa, [3]University of California - San Diego
{vitor.marchesan,livia.camargo}@ufabc.edu.br, {livia.galinari,luiza.possa,tiagoaomendes}@ufv.br, jdutramolino@ucsd.edu

1 INTRODUCTION

To achieve maximum efficiency in a biological compound production, researchers need to screen multiple parameters, to identify the bio products viability, especially on engineered cells [2]. There are a multitude of tools to automate and screen important parameters on the micro-scale. Yet, there is still a bottleneck, the validation of the acquired data in the benchtop scale, due to cost and low customization[10]. For example, automated used bioreactor hardware costs at least $6.000,00 [4], which results in labs limiting the number of reactors to very few or using only non-automated cultures. Today, most innovative bioreactors can read online data and estimate biomass growth using a built-in fluorometer and densitometer [7]. There are a few options of open-source bioreactors available, but most needs the user to program parameters in C++ language [5], or they lack flexibility [8].

This work aims to show a flexible, low-cost benchtop bioreactor capable of supplying multiple features. It can work in several regimens, read online and inline sensors, and active actuators to create the desired environment to best address cell needs.

2 BIOREACTOR DESIGN

First, we estimated the most common laboratory equipment of biolabs, as the bare minimum to be able to use this reactor: A laminar flux cabinet, Autoclave, and a magnetic stirrer. Then, we selected building materials that are cheap, easy to acquire, and broadly available. When an item needed custom solutions, a Filament Deposition 3D printer was used. Table 1, lists the most relevant items and costs. The materials cost was estimated for running 12 experiments as a PhotoBioReactor (PBR), due to the silicone lid wear.

Bioreactor Control

The Open-Source Single Board Computer, Raspberry Pi, has been chosen as the processing and control unit due to its flexibility of multiple input and output ports, enabling the connection of all the sensors and actuators. It runs the Linux Raspberry Pi OS together with the Mycodo software for the interface and control. Mycodo is an Open-Source system that allows adding multiple sensors, and creating actuator patterns, for example, adding fresh medium upon reaching an established pH or color reading threshold, and it can activate or deactivate the LED strips that provide the light source according to a programmed light cycle. All of this is done through its web interface, enabling real-time data visualization.

Bioreactor Construction

The standard labware reagent bottle has been chosen as the bioreactor vessel to fit all criteria, constraining on wide mouth types, enabling multiple sensors and actuators connection. The reactor cap is designed to hold the sensors and tubing, and it is made of polymerizable silicon rubber. To create this lid, a 3D mold matrix was designed and printed. The sensors implemented were: digital temperature sensor DS18B20; pH sensor for DIY makers; a digital RGB color sensor APDS9960 to estimate biomass concentration using real-time color readings of the reactor culture color [1]. To provide light energy for light-dependent cells, a 3D structure was designed

*All authors contributed equally to this research. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, as well as by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – Finance code 2016/12992-6.

and printed to hold common LED light strips, which can provide up to 100 µmol photons.m^{-2}.s^{-1} on the reactor surface. The agitation was done by a magnetic stirrer at the bottom. Detailed steps of construction are available at: http://github.com/VitorFrost/photobioreactor .

3 BIOREACTOR TESTING

To challenge and validate the bioreactor viability, a model organism[9] *Chlamydomonas reinhardtii* expressing heterologous fluorescent protein mCherry[6] was used, and the chosen reactor configuration was a single batch PBR with 500mL working volume. Strains were grown in Erlenmeyer flasks containing 100 mL of TAP media [3] on an orbital shaker at 110 RPM, and under constant light (100 µmol photons.m^{-2}.s^{-1}) for 3 days at room temperature, in which an inoculation of 5mL of this culture was added to the reactor containing 500mL of TAP. The experiment was run in triplicate, a sample was daily taken from days zero to five, and an extra sample was taken on day 7 to verify reactor stability. *Figure 1* shows a working reactor and some details.

4 DISCUSSION AND FUTURE WORK

Results showed high correlation between online readings of the green color with offline Chlorophyll samples reading, as seen in *Figure 2*. The pH sensors also had a high correlation, but it displayed a -0.5 offset from the samples. The increase in chlorophyll, pH, and mCherry fluorescence (figure 2-a,b,c) indirectly indicates cell growth, and that it was possible to create a working Bioreactor, capable of providing adequate conditions to cell growth, along with online data collection, allowing faster detection of growth curve phase changes. The next step is to test the dynamics system, especially a Fed-Batch Bioreactor, and to test with another organism.

REFERENCES

[1] Benavides, M. et al. 2015. Design and test of a low-cost RGB sensor for online measurement of microalgae concentration within a photo-bioreactor. *Sensors (Switzerland)*. 15, 3 (Feb. 2015), 4766–4780. DOI:https://doi.org/10.3390/S150304766.

[2] de Bournonville, S. et al. 2019. Towards Self-Regulated Bioprocessing: A Compact Benchtop Bioreactor System for Monitored and Controlled 3D Cell and Tissue Culture. *Biotechnology Journal*. 14, 7 (Jul. 2019), 1800545. DOI:https://doi.org/10.1002/BIOT.201800545.

[3] Gorman, D.S. and Levine, R.P. 1965. Cytochrome f and plastocyanin: their sequence in the photosynthetic electron transport chain of Chlamydomonas reinhardi. *Proceedings of the National Academy of Sciences of the United States of America*. 54, 6 (Dec. 1965), 1665–1669. DOI:https://doi.org/10.1073/pnas.54.6.1665.

[4] LabX Classifieds: 2022. *https://www.labx.com/fermenters-bioreactors*. Accessed: 2022-10-06.

[5] Microbial Bioreactor. 2018. *https://www.hackster.io/open-bioeconomy-lab/microbial-bioreactor-d7f61b*. Accessed: 2022-10-6.

[6] Molino, J.V.D. et al. 2018. Comparison of secretory signal peptides for heterologous protein expression in microalgae: Expanding the secretion portfolio for Chlamydomonas reinhardtii. *Plos one*. 13, 2 (Feb. 2018), e0192433–e0192433. DOI:https://doi.org/10.1371/JOURNAL.PONE.0192433.

[7] Nedbal, L. et al. 2008. A photobioreactor system for precision cultivation of photoautotrophic microorganisms and for high-content analysis of suspension dynamics. *Biotechnology and Bioengineering*. 100, 5 (Aug. 2008), 902–910. DOI:https://doi.org/10.1002/BIT.21833.

[8] Pioreactor: 2022. *https://github.com/pioreactor/*. Accessed: 2022-10-06.

[9] Scranton, M.A. et al. 2015. *Chlamydomonas* as a model for biofuels and bio-products production. *The Plant Journal*. 82, 3 (May 2015), 523–531. DOI:https://doi.org/10.1111/tpj.12780.

[10] Walls, L.E. et al. 2022. Definitive screening accelerates Taxol biosynthetic pathway optimization and scale up in Saccharomyces cerevisiae cell factories. *Biotechnology Journal*. 17, 1 (Jan. 2022), 2100414. DOI:https://doi.org/10.1002/BIOT.202100414.

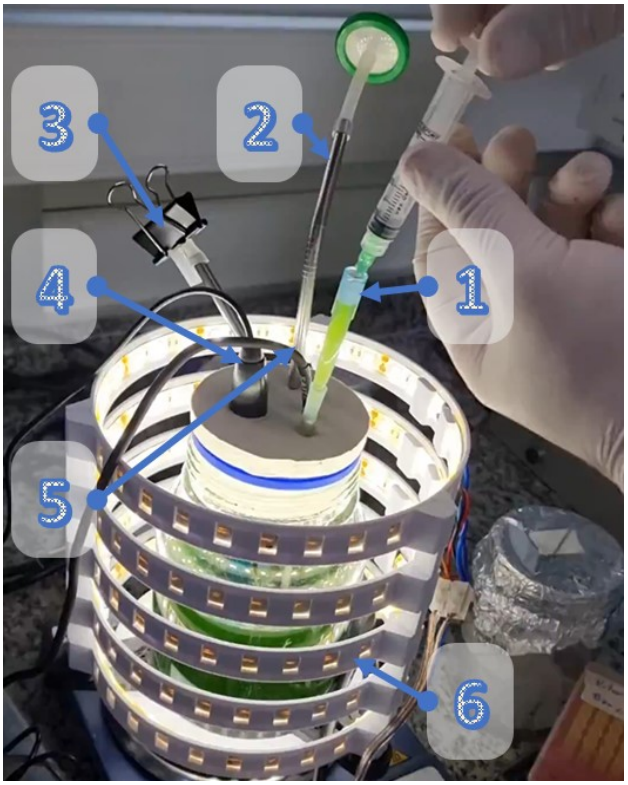


Figure 1: Bioreactor operating as Photobioreactor (1) sampling port, (2) gas outlet, (3) closed gas inlet or input, (4) pH probe, (5) temperature sensor, and (6) color sensor.

***Table 1*: Bioreactors parts and cost for running 12 experiments.**

Bioreactor parts	Requirements	Price/unit (USD)	Total price
Platinum silicone rubber Shore 20A	0.275kg	$60/kg	$16.50
Raspberry Pi 3 B+ or Raspberry Pi 4 B 1 GB	1 unit	$36.38	$36.38
Temperature sensor ds18b20	1 unit	$8.99	$8.99
pH sensor kit	1 unit	$29.50	$29.50
APDS9960 color sensor	1 unit	$7.50	$7.50
Metal straws	4 units	$0.60	$2.40
GLS 80 glass (1L or 500 mL)	1 unit	$27.51	$27.51
5v-3.3v level shifter	1 unit	$3.50	$3.50
PLA filament	0.235 kg	$56.95/kg	$13.38
PETG filament	0.032 kg	$56.99/kg	$1.82
Single-use syringe filter 0.22 micron sterile with 33mm diameter	24 units	$4.32	$103.68
12V 10A power supply	1 unit	$17.99	$17.99
5050 white led strip lights	2.5 meters	$2.53/meter	$11.33
400W DC Mosfet module	3 units	$4.20	$12.60
Peristaltic Pump	1 unit	$9.98	$9.98
Prototype Breakout PCB Shield Hat for Raspberry Pi	1 unit	$3.25	$3.25
Silicone Tube 2mm ID x 4mm OD	2 meters	$5.89/meter	$11.78
Some wires, electrical connectors, and soldering	Multiple items	-	$50
			TOTAL = ~ $370.00

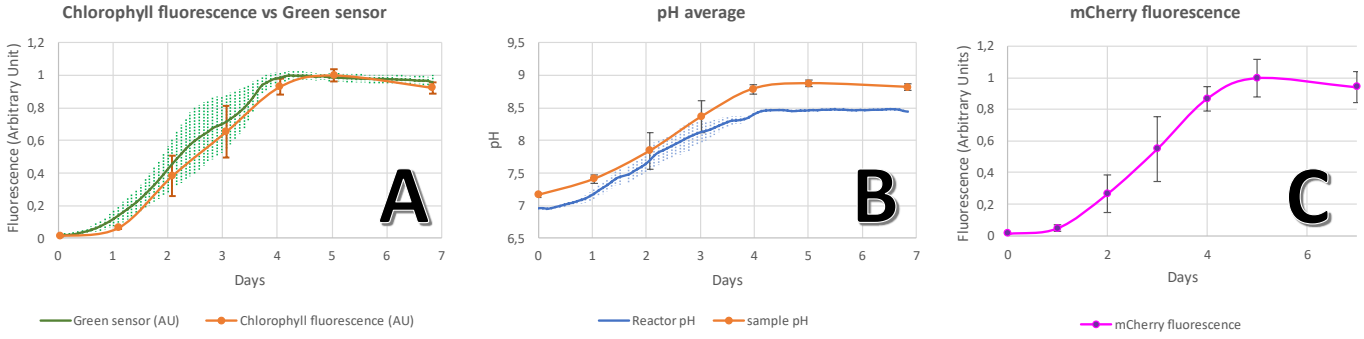


Figure 2: Graphical comparison of automated data collection and offline samples during the time of C. reinhardtii cultivation. Graph A shows in orange chlorophyll fluorescence offline sample readings and, in green, inline bioreactor readings of the green color channel measuring color development. On B, in orange, pH offline sample readings, and, in blue, online pH readings. On C, Fluorescence readings of offline samples. All fluorescence readings are in Arbitrary Units

Harnessing Biofoundries for the forward engineering of strains, with a focus on increased cis, cis-muconic acid titers in yeast.

Kealan Exley[1] Linas Tamošaitis[2] Lars K. Nielsen[1, 3]
Zofia D. Jarczynska[1,*] Giovanni Schiesaro[2] Vijayalakshmi Kandasamy[1]

keexley@biosustain.dtu.dk

1. CfB Biofoundry, Novo Nordisk Foundation Center for Biosustainabilty, DTU, Denmark
2. Cell Architecture Novo Nordisk Foundation Center for Biosustainabilty, DTU, Denmark
3. Australian Institute for Bioengineering and Nanotechnology, UQ, Australia

1. INTRODUCTION

The CfB biofoundry capabilities includes an Inscripta Onyx platform to generate massively parallel genome edited strains to accelerate the DBTL cycle. The Inscripta's automated bench-top appliance for genome engineering facilitates rapid large-scale CRISPR editing of S. cerevisiae or E. coli to introduce up to 10,000 edits [1]. Using proprietary technology, the MAD7 nuclease can introduce targeted single-nucleotide polymorphisms (SNPs), insertions, and/or deletions, into the host genome [2].

With the Inscripta Onyx platform's ability to generate thousands of strain variants within days, it is advantageous to match this pace of library generation to the identification of useful strain variants. Biofoundries possess infrastructure for the execution of high-throughput automated methods and are indispensable for the rapid phenotypical screening and sequencing of strain libraries [3]. By combining the capabilities of the Inscripta Onyx and the high-throughput platforms at the CfB Biofoundry, for targeting and identifying novel genes respectively, an optimized yeast strain producing cis,cis-muconic acid (ccM) with increased yields was generated within one month. Muconic acid is an important commodity chemical for nylon production. A GFP-biosensor coupled with the ccM-producing yeast strain enabled the online monitoring of ccM production [4]. Using fluorescent assisted selection equipment, such as the PIXL microbial colony picker and FACS, GFP fluorescent ccM-producing yeast strain were isolated. Subsequent in-house sequencing of barcoded editing plasmid DNA from Inscripta tracked the abundance of each edit in the Inscripta library and revealed a number of unique gene targets that improved ccM yields.

The Inscripta Onyx in conjunction with the integrated infrastructure of a Biofoundry enables the rapid and expansive completion of many iterations of a DBTL cycle. For instance, after analyzing single advantageous gene targets, determined by an Inscripta library, the CfB Biofoundry infrastructure enables multiplex-CRISPR targeting to generate variants with multiple edits for further strain optimization. The Inscripta-generated libraries aid genome discovery and enable forward engineering of strains to improve product titers [1].

*Both authors contributed equally to this research. The research was funded by a Novo Nordisk Foundation

REFERENCES

[1] www.inscripta.com

[2] Rojek J, Basavaraju Y, Nallapareddy S, et al. Mad7: An IP friendly CRISPR enzyme. *Authorea.* 2021.

[3] Hillson, N., Caddick, M., Cai, Y. *et al.* Building a global alliance of biofoundries. *Nat Commun.* 2019.

[4] Guokun Wang G, Süleyman Øzmerih S , Rogério Guerreiro R, et al. Improvement of *cis*,*cis*-Muconic Acid Production in *Saccharomyces cerevisiae* through Biosensor-Aided Genome Engineering *ACS Synthetic Biology.* 2020

The iBioFoundry: Automated, Low-Cost, High-Throughput Experimentation

Camillo Moschner[1], **Charlie Wedd**[1], **Georgeos Hardo**[1], **Somenath Bakshi**[1]
[1]Control Group, Department of Engineering, University of Cambridge, Cambridge, UK
cm967@cam.ac.uk,cdw42@cam.ac.uk,gh464@cam.ac.uk,sb2330@cam.ac.uk

1 INTRODUCTION

A fundamental limitation to the engineering and discovery of novel biological systems is low experimental throughput due to time and effort constraints associated with manual sample handling. To address this need, BioFoundries, highly-automated biological facilities, have emerged that can be hired for particular experiments and method development [5, 8]. Many labs, however, do not have access to these facilities, either due to geographical or financial limitations. Furthermore, many laboratories still prefer complete control over their experimental setups.

Here, we present the "in-house BioFoundry" (iBioFoundry), an open-source, low-cost automation pipeline written in Python and designed for maximal flexibility in DNA assembly and experiment preparations. The iBioFoundry represents a throughput-adaptive workflow that can be applied to a variety of different liquid handling systems. To enable easy access to this lab automation pipeline we showcase its use on the low-cost Opentrons OT-2 liquid handling robot.

2 THE IBIOFOUNDRY

The iBioFoundry follows the implementation steps of a given application, and in version 1.0 focuses on method-agnostic DNA assembly. Applications are divided into separate, consecutive operations (Figure 1). Each operation requires CSV-based input spreadsheets and a Python Jupyter Notebook, and generates a range of CSV-based output spreadsheets for precise sample tracking.

A DNA assembly workflow starts on day 1 with DNA extracts automatically being molarity-adjusted ("molaritised") into a 96- or 384-well plate, called "ark plate", based on a CSV-input of the DNA to be used. Next, the operator generates a reagents definition and a design spreadsheet. The reagents definition file can be adjusted to accommodate any DNA assembly method (e.g. Golden Gate, Gibson or Gateway cloning) while the design file allows one to choose between two design modes: "Defined" designs require each DNA piece of an assembly to be be precisely defined in the same row of the spreadsheet. "Factorial" designs allow for more flexibility and automated design. Each column represents a particular part position (e.g. promoter, ribosomal binding site (RBS), coding sequence (CDS), terminator, destination vector), and rows have to be filled with all the variants of each part which one intends to test. The iBioFoundry then automatically generates a fully factorial assembly design of the given parts to create all combinations of possible assemblies (e.g. 5x promoters, 2x RBSs, 3x CDSs, 1x terminator, 3x destination vectors generates 90 assembly combinations). Automatically generated assemblies are indexed, and their part composition and movement between tubes precisely recorded, allowing for sample tracking throughout the entire automation pipeline. Dynamic liquid handling algorithms allow the factorial design to be column-adjusted, i.e. simply modifying the number of columns in the design spreadsheet allows for expansion or reduction of the number of part types to be used. Finally, to increase accuracy and reduce time during liquid handling, the iBioFoundry calculates the most effective pipetting strategy based on repeated usage of the same DNA parts.

Consecutive thermocycling, specific to the assembly method chosen, is performed off-deck to allow for increased usage of the robotic platform. Further Jupyter Notebooks calculate and execute liquid handling for chemical transformations, cell spreading on multi-well plates, colony PCRs on day 2, and automated liquid culture inoculation with PCR-verified colonies in 96-deep well plates. Finally, on day 3, sample tracking allows for fast creation of 96-well-based glycerol stocks. The entire overnight growth plate can then be sent directly for sequencing or can be used for DNA extraction in the lab.

This DNA assembly automation workflow enables low-cost creation of up to 192 plasmids by one person in a period of 3 days, starting with purified DNA and finishing with glycerol stocks of PCR-verified assemblies, and sequencing-ready samples. DNA assembly efficiencies are dependent on the assembly method chosen and the parts being assembled, and have been shown to be equivalent to manual liquid handling.

3 DISCUSSION

Computer-aided design and execution of biological experiments is becoming increasingly important [1]. In particular considering the vast sequencing space of biomolecules (DNA, RNA, proteins) an inconceivably large design space would have to be created and tested to generate a complete genotype-phenotype mapping for a given application [3]. Exploring even a small fraction of this design space in a

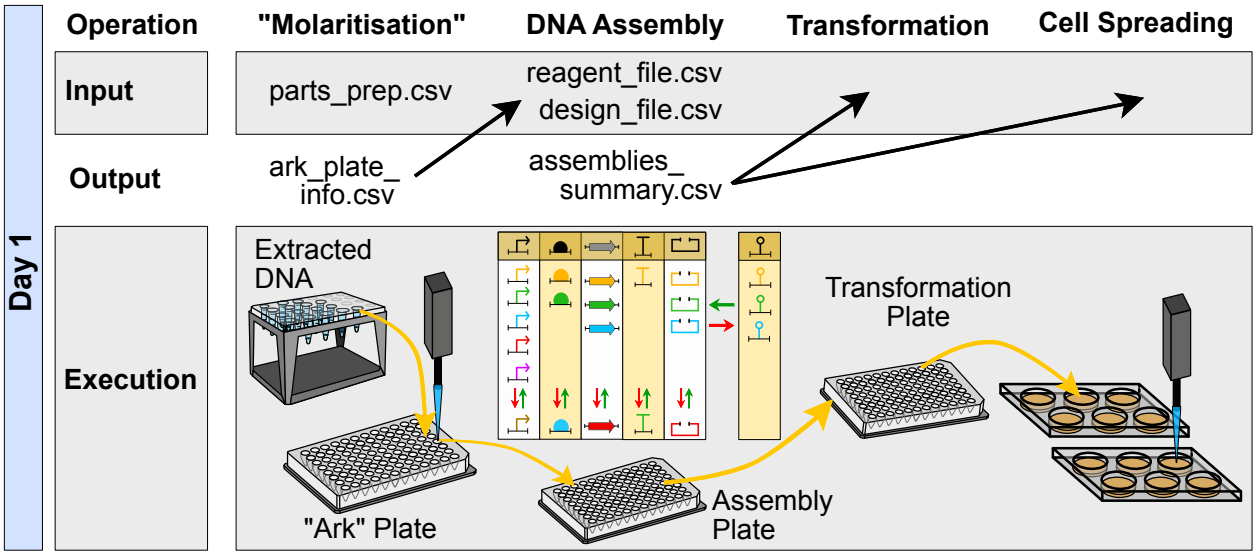


Figure 1: Typical iBioFoundry Workflow for Automated Molecular Cloning on Day 1

timely manner is only feasible using lab automation. However, most academic laboratories still do not have access to these automation techniques. The iBioFoundry represents an automation pipeline, designed to democratize lab automation, both in terms of financial investment and usability.

Two essential components in achieving this goal are a simple design and execution interface, and the possibility of low-cost implementation. The usage of Jupyter Notebooks allows for human-in-the-loop (HITL) design and liquid handling, enabling real-time feedback to the operator. Furthermore, the ability to generate both defined and factorial designs during assemblies permits simple, rapid experiment modifications, and can be seen as an introduction of a Design of Experiments (DoE) approach for a broad, non-specialised audience. DoE has recently been recognised as a transformative strategy to design and investigate biological systems [4].

Multiple pipelines for biology automation have recently been published [1, 2, 6, 7]. However, most of them are implemented on high-end, expensive liquid handling systems, and low-cost implementations have thus far shown limited capabilities to automatically adapt to different methods and changing throughput between experiments. As a result, the required additional time and modification of existing workflows has been prohibitive for widespread adoption [9]. The iBioFoundry counters this trend using a throughput-adaptive workflow, implemented on the low-cost Opentrons OT-2 robot.

We believe that this technology has the power to truly democratize lab automation, and accelerate design and discovery of novel biological systems.

REFERENCES

[1] Chory, E. J., Gretton, D. W., DeBenedictis, E. A., and Esvelt, K. M. Enabling high-throughput biology with flexible open-source automation. *Molecular Systems Biology 17*, 3 (2021), 1–10.

[2] Enghiad, B., Xue, P., Singh, N., Boob, A. G., Shi, C., Petrov, V. A., Liu, R., Peri, S. S., Lane, S. T., Gaither, E. D., and Zhao, H. PlasmidMaker is a versatile, automated, and high throughput end-to-end platform for plasmid construction. *Nature Communications 13*, 1 (2022).

[3] Gaeta, A., Zulkower, V., and Stracquadanio, G. Design and assembly of DNA molecules using multi-objective optimization. *Synthetic Biology 6*, 1 (2021), 1–9.

[4] Gilman, J., Walls, L., Bandiera, L., and Menolascina, F. Statistical Design of Experiments for Synthetic Biology. *ACS Synthetic Biology 10*, 1 (2021), 1–18.

[5] Hillson, N., Caddick, M., Cai, Y., Carrasco, J. A., Chang, M. W., Curach, N. C., Bell, D. J., Le Feuvre, R., Friedman, D. C., Fu, X., Gold, N. D., Herrgård, M. J., Holowko, M. B., Johnson, J. R., Johnson, R. A., Keasling, J. D., Kitney, R. I., Kondo, A., Liu, C., Martin, V. J., Menolascina, F., Ogino, C., Patron, N. J., Pavan, M., Poh, C. L., Pretorius, I. S., Rosser, S. J., Scrutton, N. S., Storch, M., Tekotte, H., Travnik, E., Vickers, C. E., Yew, W. S., Yuan, Y., Zhao, H., and Freemont, P. S. Building a global alliance of biofoundries. *Nature Communications 10*, 1 (2019), 1038–1041.

[6] Ortiz, L., Pavan, M., McCarthy, L., Timmons, J., and Densmore, D. M. Automated robotic liquid handling assembly of modular DNA devices. *Journal of Visualized Experiments 2017*, 130 (2017), 1–7.

[7] Storch, M., Haines, M. C., and Baldwin, G. S. DNA-BOT: A low-cost, automated DNA assembly platform for synthetic biology. *Synthetic Biology 5*, 1 (2020), 1–7.

[8] Tellechea-Luzardo, J., Otero-Muras, I., Goñi-Moreno, A., and Carbonell, P. Fast biofoundries: coping with the challenges of biomanufacturing. *Trends in Biotechnology 40*, 7 (2022), 831–842.

[9] Walsh, D. I., Pavan, M., Ortiz, L., Wick, S., Bobrow, J., Guido, N. J., Leinicke, S., Fu, D., Pandit, S., Qin, L., Carr, P. A., and Densmore, D. Standardizing Automated DNA Assembly: Best Practices, Metrics, and Protocols Using Robots. *SLAS Technology 24*, 3 (2019), 282–290.

Model-driven analysis and debugging of synthetic logic circuits with new CRISPRi components

Davide De Marchi[1], Roman Shaposhnikov[1], Paolo Magni[1], Lorenzo Pasotti[1,2]

[1]Department of Electrical, Computer and Biomedical Engineering, University of Pavia, IT; [2]Institut Pasteur, Paris, FR

{davide.demarchi, paolo.magni, lorenzo.pasotti}@unipv.it, roman.shaposhnikov01@universitadipavia.it

1 INTRODUCTION

Engineering-inspired layered design of synthetic circuits enables the decoupling of environment-specific sensing (inputs) and actuation (outputs) devices with an information processing layer, which provides the required complexity for the designed functions [1]. Information processing requires toolkits of components that mimic digital or analog behaviors like logic gates, amplifiers, and switches. CRISPR interference (CRISPRi) modules are used to construct logic gates and other networks in many organisms due to their regulation programmability [2]. In addition to the traditionally adopted *Streptococcus pyogenes* dCas9 (SpdCas9), genes from other organisms will expand the available toolkit of orthogonal parts and are expected to overcome SpdCas9 limitations like size, toxicity and off-targeting [3].

Our limited ability to predict the output of even simple circuits currently hampers the design of synthetic circuits, motivating efforts in developing predictive models for interconnected systems. Among the many circuit-, host- and environment-borne unpredictability sources, the stress by heterologous expression is a major cause of unexpected behaviors and tools are needed to characterize, predict and mitigate its effects [4,5].

In this work, we showcase the *Staphylococcus aureus* dead-Cas9 (SadCas9) to design synthetic circuits in engineered bacteria. This regulator was recently used in other organisms motivated by its smaller size than SpdCas9 and its more restrictive PAM sequence (NNGRRT) with a theoretical off-target reduction [6,7]. We show that SadCas9 is suitable for NOT gate design, but quantitatively unexpected transfer function features were observed. Mathematical modeling was then adopted to drive the debugging of this circuitry.

2 CIRCUIT DESCRIPTION AND CONSTRUCTION

The analyzed circuitry is illustrated in Fig.1. SadCas9 was obtained from Addgene plasmid 113718 [8], sgRNA by de-novo synthesis and the other parts were from iGEM Distributions. BioBrick Standard Assembly was used to construct circuits, with HSL-inducible SadCas9 cassette in low copy, RFP target cassette (driven by different constitutive promoters–P_{con}) in medium copy and IPTG-inducible sgRNA cassette in high copy vectors. Plasmids were co-transformed in the TOP10F' *Escherichia coli* strain. The sgRNA sequence was customized to target a 21-bp sequence that was cloned downstream of P_{con}.

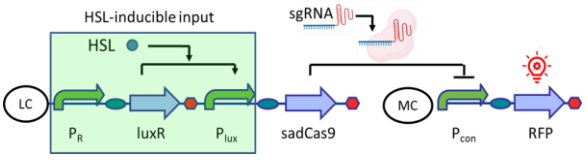


Figure 1: SadCas9-based NOT gate with 3-oxo-C6 homoserine lactone (HSL) as input and RFP as output.

3 QUANTITATIVE EXPERIMENTS

Fluorescence (*F*) and absorbance (*A*) were measured via microplate reader (Infinite F200Pro, Tecan). Cultures were grown in M9 with glycerol and casamino acids in 96-well plates, incubated at 37°C, 3-mm linear shaking, 5-min sampling time. A signal proportional to RFP synthesis rate per cell was obtained as $S_{cell}(t) = dF^*(t)/dt \,/\, A^*(t)$, where *F** and *A** are the background-subtracted *F* and *A*. The resulting $S_{cell}(t)$ was averaged in exponential growth phase as steady-state output.

4 EXPERIMENTAL RESULTS

The circuits were studied using constitutive promoters of diverse strengths (J23118–medium, J23119–strong) to drive RFP. We expected output curves with the same shape and different amplitudes due to the same regulation mechanism on promoters with different activities, as observed in previous work [9]. Both circuit versions qualitatively resulted in functional NOT gates, as shown by the decreasing trend as a function of HSL (Fig.2). As expected, the stronger promoter produced higher RFP levels. However, the shape of the curves was unexpectedly different in terms of quantitative features as activity range, switch point and steepness. We assumed that the expression load caused by RFP could explain the experimental trends observed in the data.

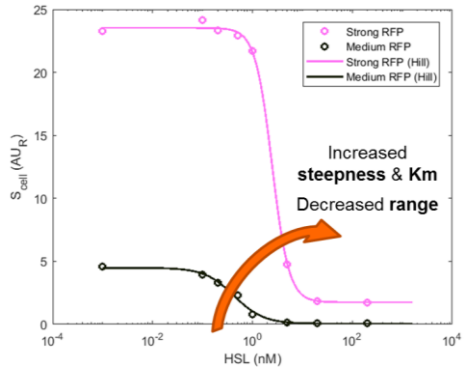


Figure 2: Experimental data (average of 3 replicates) and fitting by a Hill equation to summarize the input-output trends.

5 MATHEMATICAL MODEL DEFINITION

We simulated the HSL-RFP transfer function at steady-state by Hill equation models with cell load. The resulting model capturing the shape variation observed *in vivo* is:

$$S_{max,luxR} = \alpha_{PR}$$

$$Q = \frac{LuxR}{1 + \left(\frac{K_{lux}}{H}\right)^{\eta_{lux}}}$$

$$S_{max,cas} = \delta_{lux} + \frac{\alpha_{lux}}{1 + \left(\frac{K_{lux}}{Q}\right)^{\eta_{lux}}}$$

$$S_{max,rfp} = \delta_{con} + \frac{\alpha_{con}}{1 + \frac{Cas}{K_{cas}}}$$

$$S_i = \frac{S_{max,i}}{1 + J_{rfp} \cdot S_{max,rfp}}$$

$$LuxR\ or\ Cas\ or\ RFP = S_i/\mu$$

In this set of equations, $S_{max,i}$ represents the maximum synthesis rate of the *i*-th protein of the circuit (LuxR, SadCas9 or RFP) achievable in absence of cell load, S_i is the actual synthesis rate per cell, and *LuxR*, *Cas* and *Q* represent the intracellular levels of LuxR, SadCas9:sgRNA and HSL:LuxR active complex. Hill equations have the α, δ, K, η parameters indicating maximum rate, basic activity, half-induction concentration and Hill coefficient, respectively. Cell load was described as a scale factor between $S_{max,i}$ and S_i, with J_{rfp} being the resource usage parameter [4]. Finally, μ is growth rate. Simulations were run via Matlab R2017b (MathWorks) with parameters fixed to values selected from previous works [9,10].

6 SIMULATIONS FOR MODEL-BASED DEBUGGING

The simulation of S_{rfp} for different values of J_{rfp} successfully predicted the observed trend (Fig.3). The model confirms that hidden interactions occur in the circuit by turning the target gene into a global repressor of all the three genes, thus generating a feedback loop. Wide variations are predicted to occur as a function of the resource usage (index of cell load) of the target gene.

7 CONCLUSIONS

This work showed unexpected behaviors in the output of a synthetic circuit affected by cell load. If not considered, load effects may lead to wrong biological conclusions (e.g., in terms of Hill equation parameters) or low generalization power (e.g., when reusing components in circuits driving different genes). We used SadCas9 as a new model system, also demonstrating it is suitable for circuits design and it is being successfully applied in our lab to construct other logic gates and biological devices.

REFERENCES

[1] Wang B et al. A modular cell-based biosensor using engineered genetic logic circuits to detect and integrate multiple environmental signals. *Biosens Bioelectron* 2013, 40:368-76.
[2] Santos-Moreno J, Schaerli Y. CRISPR-based gene expression control for synthetic gene circuits. *Biochem Soc Trans* 2020, 48:1979-93.
[3] Wang J et al. The rapidly advancing Class 2 CRISPR-Cas technologies: A customizable toolbox for molecular manipulations. *J Cell Mol Med* 2020, 24:3256-70.
[4] Qian Y et al. Resource competition shapes the response of genetic circuits. *ACS Synth Biol* 2017, 6:1263-72.
[5] Ceroni F et al Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nat Methods* 2015, 12:415-8.
[6] Ran FA et al. In vivo genome editing using Staphylococcus aureus Cas9. *Nature* 2015, 520:186-91.
[7] Friedland AE et al. Characterization of Staphylococcus aureus Cas9: a smaller Cas9 for all-in-one adeno-associated virus delivery and paired nickase applications. *Genome Biol* 2015, 16:257.
[8] Savic N et al. Covalent linkage of the DNA repair template to the CRISPR-Cas9 nuclease enhances homology-directed repair. *Elife* 2018, 7:e33761.
[9] Bellato M et al. CRISPR interference modules as low-burden logic inverters in synthetic circuits. *Front Bioeng Biotechnol* 2022, 9:743950.
[10] Pasotti L et al. Re-using biological devices: a model-aided analysis of interconnected transcriptional cascades designed from the bottom-up. *J Biol Eng* 2017, 11:50.

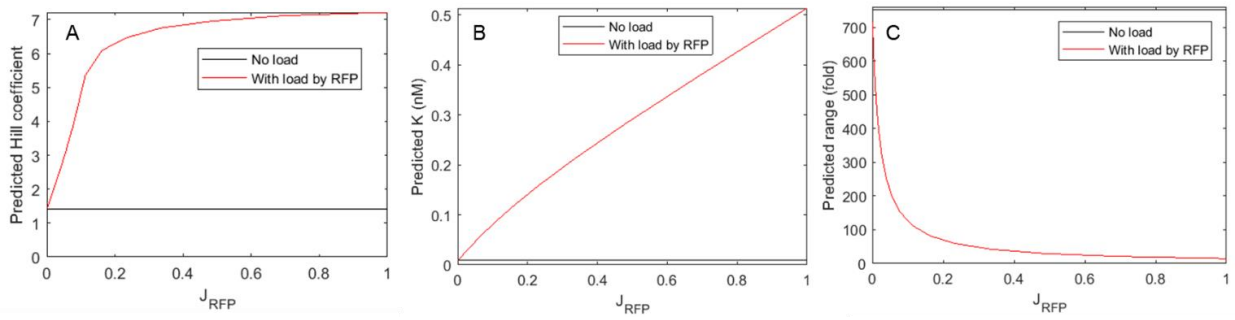


Figure 3: Simulated Hill coefficient (A), switch point (B) and activity range (C) of the NOT gate as a function of resource usage (J_{rfp}).

From Specification to Implementation: Assume-Guarantee Contracts for Synthetic Biology

Ayush Pandey[1*], **Inigo Incer**[1,2*], **Alberto Sangiovanni-Vincentelli**[2], **Richard M. Murray**[1]
[1]California Institute of Technology [2]University of California, Berkeley

1 INTRODUCTION

Mathematical modeling has played a key role in the foundations of synthetic biology and since then has been extensively used to study the design of engineered biological systems [3]. From a design standpoint, the development of models is necessary to decide when to use the circuits described by these models. As system complexity increases, we believe that it is necessary to develop a complete design methodology that begins with a top-level description of the system's objective and guides the designer in the generation of an implementation that can be proven to meet the specification. This is the main contribution of this paper. We present a methodology that decouples reasoning about component specifications from reasoning about the modeling details of each component. This methodology allows designers to focus on particular aspects of the design process at various levels of detail while ensuring that other aspects of the design are not forgotten.

Contract-based design

The design methodology presented here is centered on contract-based design [5]. At the heart of contract-based design are assume-guarantee contracts, which are formal specifications that distinguish between the responsibilities of a component and the assumptions made on its environment. A contract is thus as pair $C = (a, g)$ of assumptions a and guarantees g. Contracts come with a rich algebra [2, 4] that allows us, for example, to compose contracts in order to find system specifications from subsystem specifications and to find missing specifications to complete a design through the operation of quotient. We demonstrate the application of our methodology with the design of a biological AND gate. We use the contract algebra to derive the specification of the system from its components. We discuss how contracts help us to meet a system-wide specification when we have a partial implementation of the system available. Finally, we show how our methodology seamlessly connects the specification of a component with its models and implementations.

2 BIOLOGICAL AND GATE

Consider the design of an AND gate with inputs u_1 and u_2 and one output y. This design (built in [1]) is a composition of three subsystems as shown in Fig. 1, where the inputs are chemical inducer signals (salicylate and arabinose respectively), and the output y is the fluorescence of green fluorescent protein (GFP). Let the output of subsystem Σ_1 be x_1 and the output of Σ_2 be x_2. According to [1], x_1 models an engineered tRNA called supD and x_2 models an mRNA that codes for T7 RNA polymerase. For the translation of this mRNA, x_1 must be present, hence the AND logic. When both x_1 and x_2 are present, Σ_3 is activated to express GFP.

We can write the specification for the subsystem Σ_1 by describing the assumptions and guarantees of the design. We assume that at time $t = \tau_{u_1}$, we have $u_1 \geq u_{1_{\min}}$, and u_1 stays over this threshold. The contract $C_1 = (a_1, g_1)$ for Σ_1 guarantees that $x_1 \geq x_{1_{\min}}$ at time $t \leq \tau_{u_1} + t_1$. To write this specification as a contract, we split it into two viewpoints. First, we have a functionality viewpoint that $x_1 \geq x_{1_{\min}}$ follows from $u_1 \geq u_{1_{\min}}$. The other is a timing viewpoint that the event τ_{x_1}, defined as the time when $x_1 \geq x_{1_{\min}}$, happens at most t_1 time after the event τ_{u_1}, defined as the time when $u_1 \geq u_{1_{\min}}$. That is, we have the following two contract viewpoints: $C_1^f = (u_1 \geq u_{1_{\min}}, x_1 \geq x_{1_{\min}})$ and $C_1^t = (1, \tau_{x_1} \leq \tau_{u_1} + t_1)$, where 1 represents the boolean value "true".

For Σ_2, we have the input u_2 that activates x_2. For this subsystem, if we assume that at $t = \tau_{u_2}$, u_2 crosses the threshold $u_2 \geq u_{2_{\min}}$, then the subsystem specification guarantees that $x_2 \geq x_{2_{\min}}$ at time $t \leq \tau_{u_2} + t_2$. The functionality and timing contracts $C_2 = (a_2, g_2)$ for Σ_2 are $C_2^f = (u_2 \geq u_{2_{\min}}, x_2 \geq x_{2_{\min}})$ and $C_2^t = (1, \tau_{x_2} \leq \tau_{u_2} + t_2)$, where τ_{x_2} is, as before, the event when x_2 crosses the threshold $x_2 \geq x_{2_{\min}}$.

Similarly for Σ_3, under the assumptions that $x_1 \geq x_{1_{\min}}$ and $x_2 \geq x_{2_{\min}}$ starting at some $t = \max(\tau_{x_1}, \tau_{x_2})$, Σ_3 guarantees that the output y is at least $F > 1$ fold-change higher than the leaky expression output y_ϵ at time $\tau_y \leq t_3 + \max\{\tau_{x_1}, \tau_{x_2}\}$. Hence, the contracts for Σ_3 are $C_3^f = (x_1 \geq x_{1_{\min}} \wedge x_2 \geq x_{2_{\min}}, y \geq F y_\epsilon)$ and $C_3^t = (1, \tau_y \leq \max\{\tau_{x_1}, \tau_{x_2}\} + t_3)$. Note that for brevity we have not considered other viewpoints which would describe all of the AND logic conditions.

Generating specifications of the system

Now that we have the specifications for the three elements of the system, we seek the specification of the entire system. First, we use the operation of composition to obtain the specification of the subsystem consisting of components Σ_1 and Σ_2. We compute the compositions $\bar{C}_{12}^f := C_1^f \parallel C_2^f$ and

[*] Authors contributed equally.

$C_{12}^t := C_1^t \parallel C_2^t$. Then we abstract this operation to obtain a specification whose assumptions only depend on inputs and whose guarantees involve both inputs and outputs. We obtain $C_{12}^f = \left((u_1 \geq u_{1_{\min}} \wedge u_2 \geq u_{2_{\min}}), (x_1 \geq x_{1_{\min}} \wedge x_2 \geq x_{2_{\min}})\right)$. Now we obtain the specification of the entire system by composing these contracts with those of Σ_3, i.e., we compute $\bar{C}_{123}^f = C_{12}^f \parallel C_3^f$ and $\bar{C}_{123}^t = C_{12}^t \parallel C_3^t$ and then abstract these contracts to obtain $C_{123}^f = \left((u_1 \geq u_{1_{\min}} \wedge u_2 \geq u_{2_{\min}}), y \geq Fy_\epsilon\right)$ and $C_{123}^t = (1, \tau_y \leq \max\{\tau_{u_1} + t_1, \tau_{u_2} + t_2\} + t_3)$. These contracts give us a specification for the entire system. They only refer to variables that lie at the interface between the system and its environment, namely u_1, u_2, and y; there is no mention of x_1 and x_2. This allows us to "black-box" the system so that it can be used as a component of a more involved system.

Design synthesis of missing subsystem

In the discussion so far, we went from component specifications to the specification of the entire system. Now we will start from a top-level specification and the specification of a subsystem, and we will look for the specification of the missing subsystem needed to satisfy the top-level specification. Therefore, suppose that we are given a specification for the entire system: $C_s^f = \left((u_1 \geq u_{1_{\min}} \wedge u_2 \geq u_{2_{\min}}), y \geq y_{\min}\right)$ and $C_s^t = \left(1, \tau_y \leq \max\{\tau_{u_1}, \tau_{u_2}\} + t_3\right)$.
Suppose we also have available the specification of a subsystem, say the composition of Σ_1 and Σ_2 — C_{12}^f and C_{12}^t as computed above. The question is, what is the specification of an element that we have to add to C_{12}^f and to C_{12}^t so that the resulting implementation meets the system-level specifications, C_s^f and C_s^t? The largest specification with this property is given by the contract quotient. We compute the quotient and refine by removing references to the inputs u_1 and u_2: $C_q^f = \left(x_1 \geq x_{1_{\min}} \wedge x_2 \geq x_{2_{\min}}, y \geq y_{\min}\right)$ and $C_q^t = \left(1, \tau_y \leq \max\{\tau_{x_1} - t_1, \tau_{x_2} - t_2\} + t_3 \quad\right)$. Any implementation of this contract is guaranteed to satisfy the system-level specification when it operates in conjunction with an implementation of C_{12}. We now look for an implementation of C_q. We can propose a first-order model using the following expression:

$$y(t) = k_3 \left(1 - e^{-\frac{t-\tau_x}{t_3}}\right) \cdot s(t - \tau_x), \tag{1}$$

where $s(\cdot)$ is the step function and we define $\tau_x := \max(\tau_{x_1} - t_1, \tau_{x_2} - t_2)$ and $k_3 = k_o \frac{e(y_{\min} - y(0))}{e-1}$. The dynamical model is given by $\dot{y} = \left(k_y - d_y y\right) \cdot s(t - \tau_x)$, where $k_y = \frac{k_3}{t_3}$ and $d_y = \frac{1}{t_3}$. With $k_o = 1$, this model satisfies the guarantees such that $y = y_{\min}$ at the required timing guarantee $t = \tau_x + t_3$. This synthesized dynamical implementation model can be expanded further to include the modeling details specific to a synthetic biology implementation by using a Hill activator function instead of the step function. The biological implementation with a Hill function has an additional constraint on the Hill coefficient. To offset this, we set $k_o > 1$ so that the model in (1) satisfies $y > y_{\min}$ at a time $t < \tau_x + t_3$. In this way, the biological implementation with the Hill functions also satisfies the guarantees as shown in Fig. 2.

3 CONCLUDING REMARKS

We presented a contract-based design framework for synthetic biology. Current modeling practices in synthetic biology are limited to system analysis and inverse problems for system identification. Our results are a step towards a design framework that reasons about system properties using contracts and is capable of correlating implementations with specifications. Two key insights are discussed below.
Speeding up the experimental design process. Usually, biological circuit design entails multiple experimental iterations to optimize parameters such as promoter strengths, input levels, and physical conditions like temperature. In our approach, since the parameters of the implementations are mapped to the system objectives, these controllable aspects in the experimental design can be manipulated accordingly. Hence, a formal design framework serves to minimize the experimental trial-and-error steps.
Resource loading effects on system design. Engineered genetic circuits are dependent on resources such as RNA polymerase, ribosome, and ATP. As resources are used up by a subsystem, the performance of other subsystems is affected due to loading effects. The contract-based design framework proposed in this paper can be scaled to describe such environmental assumptions by adding resource viewpoints in addition to those we discussed.
We anticipate diverse future directions stemming from the research presented in this paper as we build towards model-guided design for large-scale synthetic biological systems.

REFERENCES

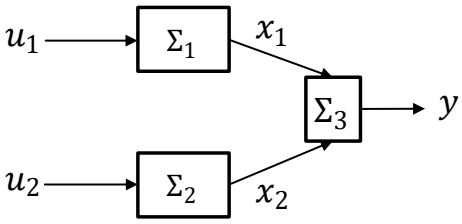
[1] Anderson, J. C., Voigt, C. A., and Arkin, A. P. Environmental signal integration by a modular and gate. *Molecular Systems Biology 3*, 1 (2007), 133.

[2] Benveniste, A., Caillaud, B., Nickovic, D., Passerone, R., Raclet, J.-B., Reinkemeier, P., Sangiovanni-Vincentelli, A. L., Damm, W., Henzinger, T. A., and Larsen, K. G. Contracts for system design. *Foundations and Trends® in Electronic Design Automation 12*, 2-3 (2018).

[3] Hsiao, V., Swaminathan, A., and Murray, R. M. Control theory for synthetic biology: recent advances in system characterization, control design, and controller implementation for synthetic biology. *IEEE Control Systems Magazine 38*, 3 (2018), 32–62.

[4] Incer, I., Sangiovanni-Vincentelli, A. L., Lin, C.-W., and Kang, E. Quotient for assume-guarantee contracts. In *16th ACM-IEEE International Conference on Formal Methods and Models for System Design* (October 2018), MEMOCODE'18, pp. 67–77.

[5] Sangiovanni-Vincentelli, A., Damm, W., and Passerone, R. Taming Dr. Frankenstein: Contract-based design for cyber-physical systems*. *European Journal of Control 18*, 3 (2012), 217–238.



u_1	u_2	y
0	0	y_ϵ
$u_{1_{\min}}$	0	y_ϵ
0	$u_{2_{\min}}$	y_ϵ
$u_{1_{\min}}$	$u_{2_{\min}}$	y^*

For $u_1 \geq u_{1_{\min}}, u_2 \geq u_{2_{\min}}$:

$$y^* \geq F y_\epsilon, \qquad t \leq \tau_y$$

Figure 1: The AND gate schematic shows composition of three subsystems to achieve AND logic implementation in an engineered biological system. The static AND gate specifications are such that when both inputs u_1 and u_2 are greater than their specified minimum values, we have $y^* \geq F y_\epsilon$, where $F > 1$ is the desired fold change in output compared to the leaky output y_ϵ. The dynamic specifications add that the output achieves the desired fold-change in time $t \leq \tau_y$.

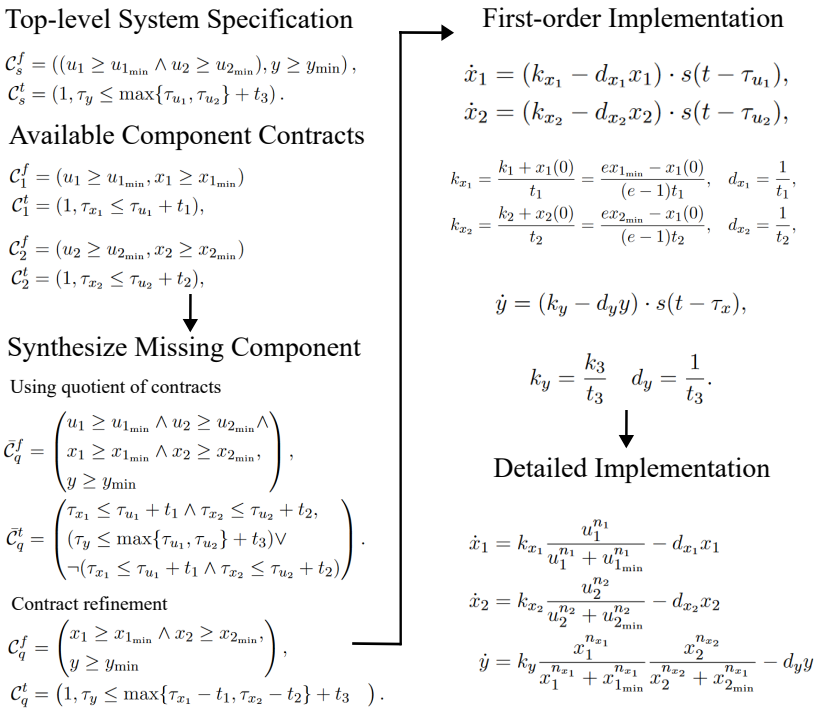


Figure 2: Given the top-level system specification and the contracts for Σ_1 and Σ_2, we synthesize the missing subsystem specification (Σ_3) using the quotient operation on contracts. Then, we propose a first-order implementation from the synthesized contract and a detailed implementation that expands the step functions into Hill functions to model the biological activation mechanisms. All parameters in the implementations are mapped to the specifications as shown. In the simulation, the dotted and dashed lines show the contract assumptions and guarantees, respectively, while solid lines show implementations. Here τ_u is defined as $\tau_u := \max\{\tau_{u_1}, \tau_{u_2}\}$. Python code to generate the simulations is available online.

A Bounded Model Checking Framework for the Analysis of Chemical Reaction Network Models

Mohammad Ahmadi[1], **Lukas Buecherl**[2], **Zhen Zhang**[3], **Chris Myers**[2], **Chris Winstead**[3], **Hao Zheng**[1*]
[1]University of South Florida , [2]University of Colorado Boulder, [3]Utah State University
{mahmadi,haozheng}@usf.edu,{chris.myers,lubu1090}@colorado.edu,{zhen.zhang,chris.winstead}@usu.edu

1 INTRODUCTION

In order to model stochastic behavior of systems, probabilistic formalisms such as Markov chains are used. Continuous Time Markov Chains (CTMCs) can be used to model chemical reaction networks' (CRNs) stochastic temporal behavior. Given a CRN modeled as a CTMC, probabilistic model checkers such as PRISM [4] can calculate the probability of certain events such as "population of species X exceeding 100 molecules within 20 seconds starting from the initial conditions of the model".

Probabilistic model checkers require the model's state-space to be finite, meaning that all species populations should be bounded. As a result, they cannot be used in scenarios where there is no upper-bound on the values the species populations can take. Tools such as STAMINA [6] are developed to handle infinite-state models. Instead of a single value, STAMINA reports a range (a lower-bound and an upper-bound) for the probability of an event.

Here, a new approach based on Bounded Model Checking (BMC) [2] is proposed to analyze infinite-state CRN models. Given a CRN C and an event $X \xrightarrow{t \leq T} \theta$ (population of species X reaching θ within T seconds), this framework intends to find a set of traces satisfying the event. A trace satisfying the event $X \xrightarrow{t \leq T} \theta$ is a valid sequence of states that start in the initial state of the model and end in a state where $X = \theta$. We call such a trace a *witness*. The problem of finding a witness of length k is encoded as a satisfiability modulo theories (SMT) problem, $BMC(k)$, and is solved using a SMT solver. Starting from $k = 0$, this framework tries to find witnesses of length k. If there are no witnesses for a particular bound k, or if all witnesses of bound k are already found, k is incremented by 1 in order to look for longer witnesses. The outcome of this approach is a set, $\mathcal{W}$, of witnesses corresponding to a finite, partial state-space of the model constructed by overlaying these witnesses on top of each-other. A probabilistic model checker can be used to calculate the probability of the event of interest on this partial state-space. The resulting probability is a lower-bound for the probability of the event on the original model. Moreover, the set of witnesses can be further analyzed to facilitate debugging of the model. For example, if a property like "the probability of event $X \xrightarrow{t \leq T} \theta$ is less than p" is falsified on the model by calculating a lower-bound greater than p, the set of witnesses $\mathcal{W}$ can be analyzed to extract information, such as *"the relative frequency of reactions among traces leading to a state where $X = \theta$"*, in order to gain insight into why the property has been falsified and to help with the refinement of the model.

*The authors of this work are supported by the National Science Foundation Grant Nos. 1856733, 1856740, and 1900542. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

2 RESULTS

The proposed framework is tested on three CRNs. The BMC method used for analyzing these models is described in [1].

Enzymatic Futile Cycle: This model uses six-reactions to represent a biochemical futile cycle [3]. The network is illustrated in Figure 1A. The event of interest is $S_5 \xrightarrow{t \leq 100s} 40$.

Table 1 shows the results of applying the proposed BMC framework on this model. For each threshold value p, the framework tries to generate a set of witnesses satisfying the following condition: the probability of the event of interest on the partial state-space generated by that set is greater than the threshold p. Figure 1B shows the growth in probability and the size of the partial state-space constructed by the witness set as time progresses.

In order to speed up the process of the expansion of the witness set, we use a technique called *scaffolding* on all the experiments presented here. When a witness is found, it is guaranteed that the target state is reachable from every state of that witness. Therefore, a trace sharing a suffix of a witness is also a witness. The scaffolding method exploits the existing witnesses, instead of solving a BMC encoding for a large bound k, to find new witnesses that share the same suffix with any of the witnesses already found. More information on scaffolding method and how it's encoded as a SMT problem can be found in [1]. For all experiments presented here, scaffolding method is used to generate (at most) 50 new witnesses exploiting the current set of found witnesses $\mathcal{W}$, after every 3 new witnesses are found using the original BMC approach.

Modified Yeast Polarization: This model is a CRN consisting of 7 species reacting through 8 reaction channels [3]. The network is illustrated in Figure 2A. The event of interest

is $G_{bg} \xrightarrow{t \leq 20s} 50$ ($G_{bg} = 0$ in the initial state of the model). Table 2 shows the results for applying the proposed BMC framework on this model. Figure 2B shows the growth in probability and the size of the partial state-space constructed by the witness set as time progresses. In order to reduce the complexity of the BMC encoding, we use the divide-and-conquer technique to generate witnesses for this model. With divide-and-conquer, instead of finding a witness starting from the initial state and ending in a state where $G_{bg} = 50$, we first find a trace from the initial state to a state where $G_{bg} = 10$. Then, this newly found state is used as the initial state to find a trace to a state where $G_{bg} = 20$. This procedure is continued until a trace ending in a state where $G_{bg} = 50$ is found. The new trace constructed by following these 5 found traces is a witness for the event of interest. Utilizing divide-and conquer significantly reduces the complexity of the BMC encoding, resulting in a much better performance in scenarios where witnesses for the event of interest are relatively long. More information on divide-and-conquer method can be found in [1].

Circuit 0x8E: This example is a genetic circuit model that consists of 15 reactions including 79 reaction rates [5]. The model is illustrated in Figure 3A. The event of interest is $S_8 \xrightarrow{t \leq 1000s} 30$ ($S_8 = 0$ in the initial state of the model). Table 3 shows the results of applying the proposed BMC framework on this model. Figure 3B shows the growth in probability and the size of the partial state-space constructed by the witness set as time progresses. We use the divide-and-conquer technique with a step of 10 on this model, i.e. a witness to the event is constructed by sequencing three witnesses, one from the initial state to a state where $S_8 = 10$, one from the newly found state to a state where $S_8 = 20$ and finally a trace ending in a state where $S_8 = 30$.

3 DISCUSSION

The described BMC framework shows promise for analyzing stochastic behavior of CRNs. It is able to generate a lower-bound for the probability of certain events happening on a CRN model with potentially infinite state-space. An important byproduct of this framework is the generated witness set. We argue that reporting the mere probability of an event, although important, is not helpful enough for debugging a model. The generated witness set tends to be small (compared to the potentially infinite state-space of the model) and thus can be analyzed to extract information such as the relative frequency of reactions in error-traces. This information can be utilized to gain insight into the root cause of the erroneous behavior of the model. An example of how the witness set can be utilized to extract such information is shown in Table 4.

The current framework does not produce an upper-bound for the probability. We expect that modifying the encoding to find strongly connected components in the model's state-space instead of witnesses can be helpful in producing the upper-bound.

Tables 2 and 3 show that the BMC framework is currently not able to produce a witness set for higher threshold values within reasonable time frame. We expect that modifying the encoding and introducing a probability measure for the traces to search for witnesses with higher probability first would significantly improve the performance of the framework. Currently, the framework searches for the shorter witnesses first. If there are many such relatively short witnesses with abysmal probability, a large amount of CPU time is spent on finding those witnesses first, ignoring longer witnesses that could potentially have higher probabilities.

Figure 1B shows that the probability stays almost the same after the size of the partial state-space has grown larger than a certain amount. Running the framework for a longer period will result in finding longer witnesses and longer witnesses often have lower probabilities. Therefore, the conclusion that this lower-bound probability is close to the actual probability of the event can be made with high confidence. In figures 2 and 3B we can observe that the probability has not plateaued, suggesting that running the framework for a longer period will likely result in a higher lower-bound.

The current BMC framework is able to generate a lower-bound for the probability of events happening on a CRN with potentially infinite state-space. The described optimization techniques *scaffolding* and *divide-and-conquer* significantly improve the performance of the framework. We expect future improvements such as introducing the encoding for strongly connected components and to prioritize the search for high probability witnesses to improve the applicability of this framework. All scripts are accessible at https://github.com/fluentverification/bmc_counterexample/tree/main/iwbda.

REFERENCES

[1] Ahmadi, M., Zhang, Z., Myers, C., Winstead, C., and Zheng, H. Counterexample generation for infinite-state chemical reaction networks, 2022. https://arxiv.org/abs/2207.05207.

[2] Clarke, E., Biere, A., Raimi, R., and Zhu, Y. Bounded model checking using satisfiability solving. *FMSD 19*, 1 (2001), 7–34.

[3] Daigle Jr, B. J., Roh, M. K., Gillespie, D. T., and Petzold, L. R. Automated estimation of rare event probabilities in biochemical systems. *The Journal of chemical physics 134*, 4 (2011), 01B628.

[4] Kwiatkowska, M., Norman, G., and Parker, D. Prism 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification* (2011), Springer, pp. 585–591.

[5] Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., and Voigt, C. A. Genetic circuit design automation. *Science 352*, 6281 (2016).

[6] Roberts, R., Neupane, T., Buecherl, L., Myers, C. J., and Zhang, Z. Stamina 2.0: Improving scalability of infinite-state stochastic model checking. In *International Conference on Verification, Model Checking, and Abstract Interpretation* (2022), Springer, pp. 319–331.

Table 1: Results for applying the BMC approach to the enzymatic futile cycle model. The event of interest is $S_5 \xrightarrow{t \leq 100s} 40$. First column shows a probability threshold the framework tries to surpass. Second column shows the probability of the partial state-space generated by the witness set. Third column reports the time it took for the framework to finish in seconds. Size of the state-space is defined as the sum of the number of states and transitions.

threshold	probability	time	partial state-space size
1×10^{-30}	8.04×10^{-29}	2.8	39
1×10^{-20}	2.87×10^{-2}	13.2	158
4×10^{-2}	4.13×10^{-2}	29.2	190

Table 2: Results for applying the BMC approach to the modified yeast polarization model. The event of interest is $G_{bg} \xrightarrow{t \leq 20s} 50$. First column shows a probability threshold the framework tries to surpass. Second column shows the probability of the partial state-space generated by the witness set. Third column reports the time it took for the framework to finish in seconds. Any experiment that took more than 1800s is marked with -TO-. Size of the state-space is defined as the sum of the number of states and transitions.

threshold	probability	time	partial state-space size
1×10^{-90}	2.54×10^{-90}	9.4	333
1×10^{-80}	1.08×10^{-80}	25.2	681
1×10^{-70}	1.09×10^{-70}	122.7	1620
1×10^{-60}	1.11×10^{-60}	1207.6	4593
1×10^{-50}	-TO-	-TO-	-TO-

Table 3: Results for applying the BMC approach to the genetic circuit 0x8E model. The event of interest is $S_8 \xrightarrow{t \leq 1000s} 30$. First column shows a probability threshold the framework tries to surpass. Second column shows the probability of the partial state-space generated by the witness set. Third column reports the time it took for the framework to finish in seconds. Any experiment that took more than 1800s is marked with -TO-. Size of the state-space is defined as the sum of the number of states and transitions.

threshold	probability	time	partial state-space size
1×10^{-12}	1.08×10^{-12}	0.28	7
1×10^{-11}	1.94×10^{-11}	1.73	101
1×10^{-10}	1.02×10^{-10}	47.29	1131
1×10^{-9}	-TO-	-TO-	-TO-

Table 4: Example of how the witness set can be analyzed to gain insight into the behavior of the enzymatic futile cycle model. First row shows the relative frequency of each reaction among the total fired reactions after running 1000 SSA (Stochastic Simulation Algorithm) simulations of 100 time units each. Second row shows the relative frequency of each reaction among the total reactions in a witness set with probability 4.13×10^{-2} generated for the event $S_5 \xrightarrow{t \leq 100s} 40$. It can be concluded from this table that R_6 fires more frequently and R_5 fires less frequently among traces that satisfy the event $S_5 \xrightarrow{t \leq 100s} 40$ compared to normal behavior of the system.

	R_1/Total	R_2/Total	R_3/Total	R_4/Total	R_5/Total	R_6/Total
1000 SSA Simulations	0.025	0.227	0.022	0.251	0.227	0.022
Witness set $\mathcal{W}$ with $P(\mathcal{W}) = 4.13 \times 10^{-2}$	0.118	0.096	0.081	0.331	0.092	0.283

A)

$$S_1 + S_2 \xrightarrow{k_1} S_3, \quad S_3 \xrightarrow{k_2} S_1 + S_2,$$
$$S_3 \xrightarrow{k_3} S_1 + S_5, \quad S_4 + S_5 \xrightarrow{k_4} S_6,$$
$$S_6 \xrightarrow{k_5} S_4 + S_5, \quad S_6 \xrightarrow{k_6} S_4 + S_2.$$

$$k_1 = k_2 = k_4 = k_5 = 1$$
$$k_3 = k_6 = 0.1$$

Initial state:
$$S_1(0) = S_4(0) = 1, \quad S_2(0) = S_5(0) = 50,$$
$$S_3(0) = S_6(0) = 0.$$

B)

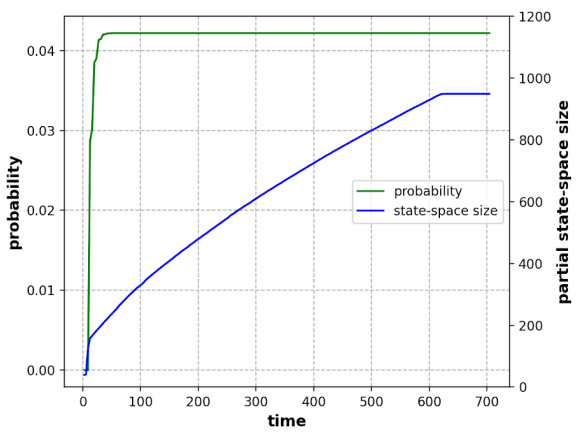


Figure 1: A) Enzymatic Futile Cycle model B) Growth in probability and the size of the partial state-space constructed by the witness set as time progresses. Size of the partial state-space is defined as the sum of the number of states and transitions. Time is measured in seconds.

A)

$$\emptyset \xrightarrow{k_{R_s}} R, \quad R \xrightarrow{k_{R_d}} \emptyset,$$
$$L + R \xrightarrow{k_{RLL}} RL + L,$$
$$RL \xrightarrow{k_R} R, \quad G_a \xrightarrow{k_{G_d}} G_d,$$
$$RL + G \xrightarrow{k_{G_a}} G_a + G_{bg},$$
$$G_d + G_{bg} \xrightarrow{k_G} G, \quad \emptyset \xrightarrow{k_{RL}} RL$$

$$k_{R_s} = 0.0038, \qquad k_{R_d} = 4.00 \times 10^{-4},$$
$$k_{RLL} = 0.042, \qquad k_R = 0.0100,$$
$$k_{G_a} = 0.011, \qquad k_{G_d} = 0.100,$$
$$k_G = 1.05 \times 10^3, \quad k_{RL} = 3.21.$$

Initial state:
$$R(0) = G(0) = 50, \; L(0) = 2$$
$$RL(0) = G_a(0) = G_{bg}(0) = G_d(0) = 0.$$

B)

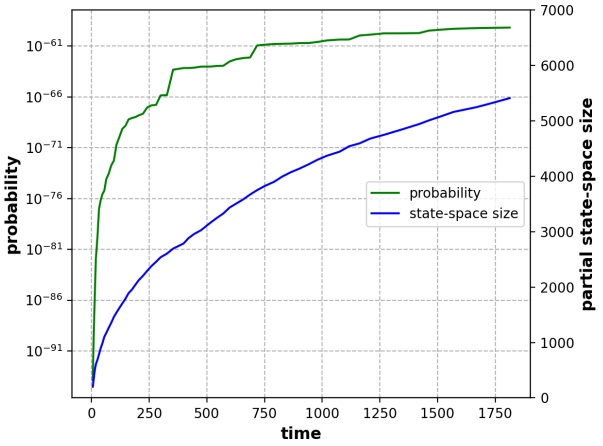


Figure 2: A) Yeast Polarization model B) Growth in probability and the size of the partial state-space constructed by the witness set as time progresses. Probability axis is in logarithmic scale. Size of the partial state-space is defined as the sum of the number of states and transitions. Time is measured in seconds.

A)

$$
\begin{aligned}
&10\,S_1 \xrightarrow{k_1} \emptyset\\
&10\,S_3 \xrightarrow{k_1} \emptyset\\
&10\,S_4 \xrightarrow{k_1} \emptyset\\
&10\,S_6 \xrightarrow{k_1} \emptyset\\
&10\,S_8 \xrightarrow{k_1} \emptyset\\
&S_9 + S_2 \xrightarrow{k_2} S_9 + S_2 + 10\,S_1\\
&S_{10} + S_4 \xrightarrow{k_3} S_{10} + S_4 + 10\,S_1\\
&S_{11} + S_4 \xrightarrow{k_4} S_{11} + S_4 + 10\,S_3\\
&S_{12} + S_7 \xrightarrow{k_5} S_{12} + S_7 + 10\,S_3\\
&S_{14} + S_2 \xrightarrow{k_6} S_{14} + S_2 + 10\,S_4\\
&S_{13} + S_7 \xrightarrow{k_7} S_{13} + S_7 + 10\,S_4\\
&S_{16} + S_5 \xrightarrow{k_8} S_{16} + S_5 + 10\,S_6\\
&S_{15} + S_1 \xrightarrow{k_9} S_{15} + S_1 + 10\,S_6\\
&S_{18} + S_6 \xrightarrow{k_{10}} S_{18} + S_6 + 10\,S_8\\
&S_{17} + S_3 \xrightarrow{k_{11}} S_{17} + S_3 + 10\,S_8
\end{aligned}
$$

Initial State:

$$
\begin{aligned}
&S_1(0) = S_3(0) = S_6(0) = 70, \quad S_2(0) = 60\\
&S_4(0) = S_5(0) = S_7(0) = S_8(0) = 0\\
&S_9(0) = \ldots = S_{18}(0) = 2.0
\end{aligned}
$$

B)

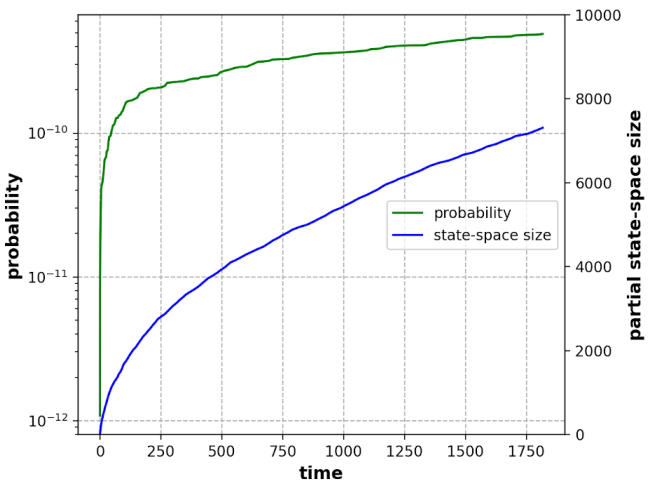


Figure 3: A) Circuit 0x8E model. Reaction rates can be found at https://github.com/fluentverification/bmc_counterexample/blob/main/iwbda/circuit_description.txt B) Growth in probability and the size of the partial state-space constructed by the witness set as time progresses. Probability axis is in logarithmic scale. Size of the partial state-space is defined as the sum of the number of states and transitions. Time is measured in seconds.

Characterization of integrase and excisionase activity in cell-free protein expression system using a modeling and analysis pipeline

Ayush Pandey[1], **Makena L. Rodriguez**[2], **William Poole**[3], **Richard M. Murray**[1,2]
[1]Control and Dynamical Systems, [2]Biology and Biological Engineering, California Institute of Technology. [3]Altos Labs

1 THE PIPELINE

Over the past few years, we have seen a widespread adoption of software tools in synthetic biology research for modeling, simulation, analysis, data exchange, and design optimizations. The focus on bio-design automation and rational design in synthetic biology has led to this enthusiastic acceptance of software tools. A few examples include COPASI [5] (modeling and simulation), iBioSim [7] (CAD-style circuit modeling), Tellurium [1] (text scripting to model circuits), RBS Calculator [12] (prediction of translation initiation rates), and automated design recommendations [11]. The rise in tools for specific tasks has led to their integration into automated pipelines like Infobiotics [6] and Galaxy SynBioCAD [4]. However, there is still need for user-friendly modeling and analysis pipelines that work alongside experimental data in a design-build-test-learn cycle. Towards that end, we present an automated Python pipeline for iterative modeling, model reduction, analysis, and parameter identification of synthetic biological circuits. We further develop on BioCRNpyler [10] (to build models), AutoReduce [9] (to obtain reduced models), and Bioscrape [14] (for simulations, analysis, and Bayesian inference using Emcee [3]) to create this computational framework shown in Figure 1. We apply the proposed pipeline to characterize an integrase and excisionase-mediated DNA recombination circuit in TX-TL[1].

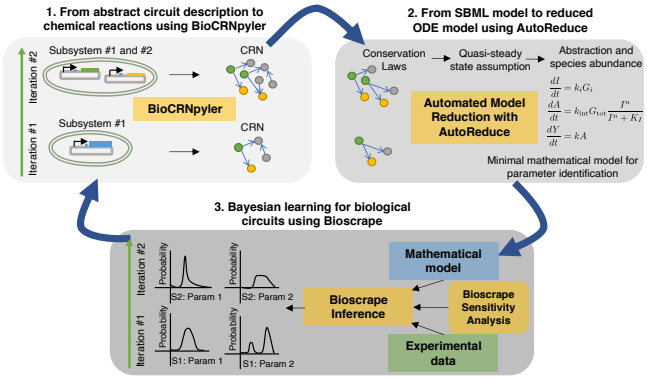


Figure 1: An iterative Python pipeline for modeling, analysis, and learning of biological circuits. In the first iteration, we build the CRN model for subsystem #1 then obtain the minimal representation suitable for parameter identification. Bayesian inference is used to find parameter distributions. The second iteration uses the previously identified context and circuit parameters to inform the model.

2 MODELING INTEGRASE ACTIVITY IN TX-TL

To characterize the integrase activity independent of the excisionase, we design a two plasmid system — (1) Bxb1 integrase expressing plasmid fused with CFP to measure integrase expression, and (2) a YFP plasmid that gets activated on integrase action (shown in Figure 2A). Using this circuit, we characterize the integrase expression in TX-TL and its flipping action on a promoter to control the fluorescent reporter expression (data shown in Figure 3A).

Towards this first goal, we model the two plasmid system in TX-TL using a detailed mechanistic chemical reaction network (CRN) with mass-action kinetics using BioCRNpyler.

We simulate this CRN model using Bioscrape (shown in Figure 2B) to explore the design space and the resource-loading effects. However, the detailed model is infeasible to fit to the experimental data due to the problem of unidentifiability [2] and high-dimensionality. Hence, we use AutoReduce to automatically derive potential reduced models for this system. We choose a reduced model that recovers the desired properties (integrase flipping, fluorescent reporter levels, and any other important context effects), shown in Figure 2C as M-3. To obtain a further reduced model, we abstract the context by switching off resource-dependent mechanisms for transcription and translation in BioCRNpyler. Then, we reduce the model using quasi-steady state approximation (QSSA) and assuming abundance of certain species to obtain a minimal ODE model (M-4). It is evident from Figure 2C that the model M-4 recovers the commonly used Hill function, however, no heuristics were used in deriving this model. For this model, we use Bayesian inference to fit the TX-TL data (Figure 3B). The parameter inference algorithm is implemented in Bioscrape as a black-box Python wrapper for the emcee package. Hence, the "full-stack" Python pipeline of modeling, design-space exploration, sensitivity analysis, model reduction, and

[1]Throughout this paper, we refer to cell-free protein expression system as TX-TL

parameter inference gives us a validated mathematical model for integrase activity in TX-TL.

3 MODELING EXCISIONASE ACTIVITY IN TX-TL

Similar to the integrase model, we construct an integrase-excisionase model and its simulations (shown in Figure 4A and 4B). Then, in multiple model reduction steps, we derive a minimal ODE model (M-8 in Figure 4C) suitable for parameter identification. Using the identified parameters, we build and predict excisionase activity in reversing the integrase action. The experimental data and the parameter identification steps for excisionase characterization are shown in Figure 5.

4 CONCLUSION

We present an automated and iterative pipeline for modeling, analysis, and parameter identification of biological circuits by building on existing Python tools — BioCRNpyler, AutoReduce, and Bioscrape. Using this pipeline, we characterize the expression and activity of enzyme-mediated DNA recombination in cell-free protein synthesis system. We build detailed chemical reaction network models from high-level description of the biological circuit and the context using BioCRNpyler. We show that many sensitive parameters in this detailed model affect the output. However, for feasible parameter identification, we use AutoReduce to automatically obtain reduced models that have fewer parameters. We derive a hierarchy of reduced models under different assumptions to finally derive a minimal ODE model for each circuit. Then, we run sensitivity analysis-guided Bayesian inference using Bioscrape for each circuit to identify the parameters of the models. We characterize the strength of integrase to flip a promoter direction as well as the excisionase mechanisms to reverse it. This characterization of the integrase-excisionase activity in cell-free opens up a new paradigm of complex circuit designs that depend on accurate control over protein expression levels independent of induction or degradation.

REFERENCES

[1] Choi, K., Medley, J. K., König, M., Stocking, K., Smith, L., Gu, S., and Sauro, H. M. Tellurium: an extensible python-based modeling environment for systems and synthetic biology. *Biosystems 171* (2018), 74–79.

[2] Davidescu, F. P., and Jørgensen, S. B. Structural parameter identifiability analysis for dynamic reaction networks. *Chemical Engineering Science 63*, 19 (2008), 4754–4762.

[3] Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific 125*, 925 (2013), 306.

[4] Hérisson, J., Duigou, T., Du Lac, M., Bazi-Kabbaj, K., Azad, M. S., Buldum, G., Telle, O., El-Moubayed, Y., Carbonell, P., Swainston, N., et al. Galaxy-synbiocad: Automated pipeline for synthetic biology design and engineering. *bioRxiv* (2022).

[5] Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. Copasi—a complex pathway simulator. *Bioinformatics 22*, 24 (2006), 3067–3074.

[6] Konur, S., Mierla, L., Fellermann, H., Ladroue, C., Brown, B., Wipat, A., Twycross, J., Dun, B. P., Kalvala, S., Gheorghe, M., et al. Toward full-stack in silico synthetic biology: Integrating model specification, simulation, verification, and biological compilation. *ACS Synthetic Biology 10*, 8 (2021), 1931–1945.

[7] Myers, C. J., Barker, N., Jones, K., Kuwahara, H., Madsen, C., and Nguyen, N.-P. D. ibiosim: a tool for the analysis and design of genetic circuits. *Bioinformatics 25*, 21 (2009), 2848–2849.

[8] Pandey, A. https://github.com/ayush9pandey/integrase-excisionase-characterization.

[9] Pandey, A., and Murray, R. M. Robustness guarantees for structured model reduction of dynamical systems with applications to biomolecular models. *International Journal of Robust and Nonlinear Control* (2022).

[10] Poole, W., Pandey, A., Tuza, Z., Shur, A., and Murray, R. M. Biocrnpyler: Compiling chemical reaction networks from biomolecular parts in diverse contexts. *BioRxiv* (2020).

[11] Radivojević, T., Costello, Z., Workman, K., and Garcia Martin, H. A machine learning automated recommendation tool for synthetic biology. *Nature communications 11*, 1 (2020), 1–14.

[12] Salis, H. M. The ribosome binding site calculator. In *Methods in enzymology*, vol. 498. Elsevier, 2011, pp. 19–42.

[13] Shin, J., and Noireaux, V. An e. coli cell-free expression toolbox: application to synthetic gene circuits and artificial cells. *ACS Synthetic Biology 1*, 1 (2012), 29–41.

[14] Swaminathan, A., Poole, W., Pandey, A., Hsiao, V., and Murray, R. M. Quantitative modeling of integrase dynamics using a novel python toolbox for parameter inference in synthetic biology. *bioRxiv* (2022).

MATERIALS AND METHODS

Cell-free experiments

Experimental characterization of both integrase and excisionase activity were done in TX-TL – an *in vitro*, cell-free protein expression system. As depicted in Figure 2A, the TX-TL reactions created are composed of S30 *E. Coli* cell extract, energy buffer, and plasmid DNA [13]. Each reaction contains a different concentration of the DNA plasmids in our synthetic circuits added to a common TX-TL master mix (cell extract and energy buffer). Then, these reactions are analyzed through the fluorescent protein expression levels on a Biotek plate reader.

Data analysis and inference

Sensitivity analysis tools in Bioscrape [14] were used to find identifiable parameters from the data. Python emcee's [3] Markov chain monte carlo (MCMC) sampler was used for the Bayesian inference algorithm implemented in Bioscrape. Code for all data analysis and parameter inference is available on Github [8].

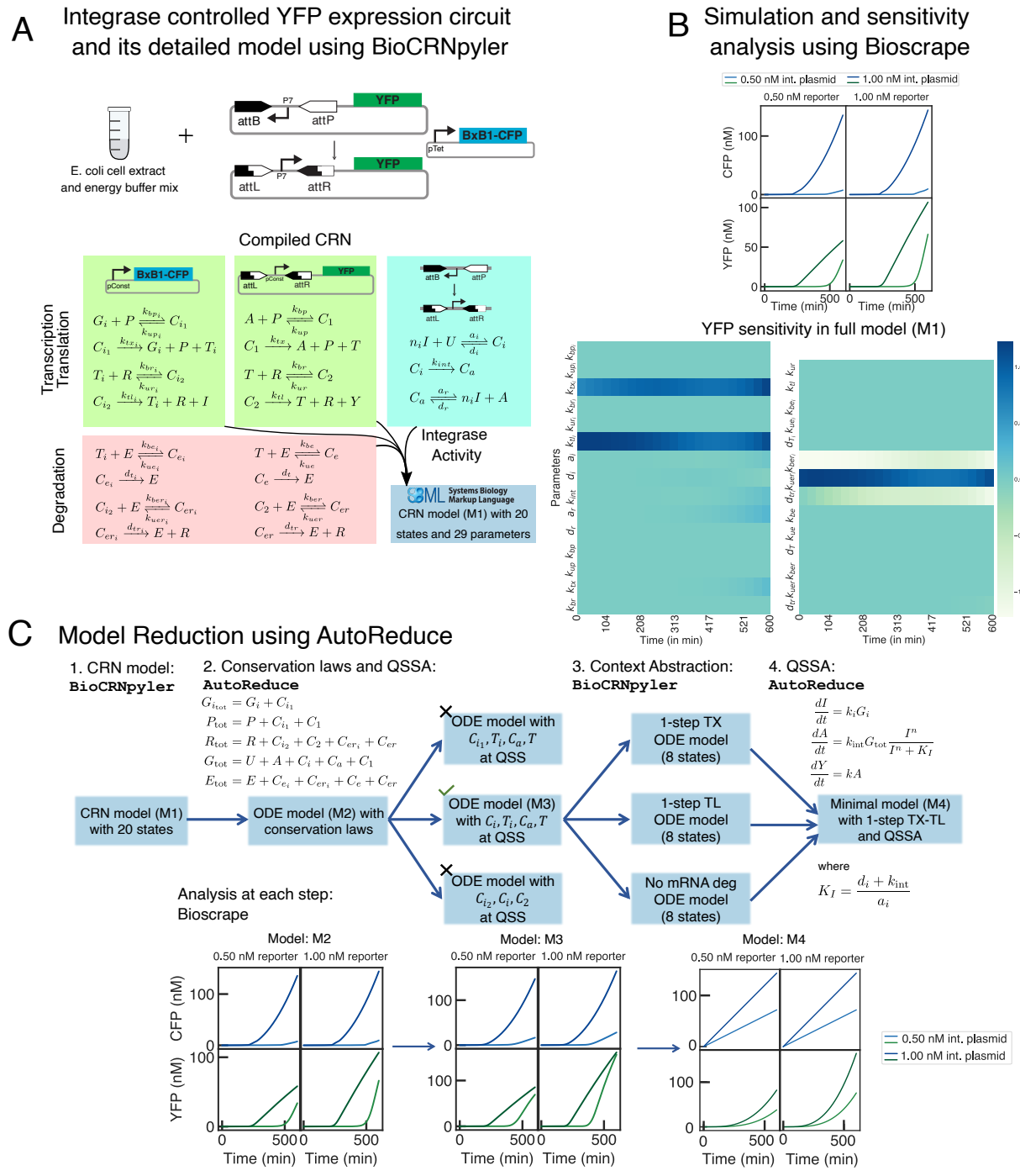


Figure 2: Modeling and analysis of integrase expression and activity in TX-TL. (A) The circuit consists of two plasmids, one expressing the Bxb1 integrase and the other with a reversed promoter upstream of YFP reporter. The integrase activity flips the promoter so that YFP is expressed. BioCRNpyler is used to convert this abstract system description into a CRN model written as an SBML file. (B) shows the simulation of the detailed model and the sensitivity analysis that shows the most influential parameters for the time-course of YFP expression. (C) The CRN model is reduced in multiple steps with AutoReduce. First, the conservation laws are determined (as shown in C-2) and a reduced model is obtained symbolically. This reduced model is further reduced using QSSA which gives multiple possibilities out of which one model (M3) is selected based on error performance metric as computed by AutoReduce. Then, a minimal model is obtained by abstracting the details and further reducing the model using QSSA and species abundance assumptions. The minimal ODE model (M4) is shown in C-4. The simulations for each reduced model using Bioscrape are shown at the bottom.

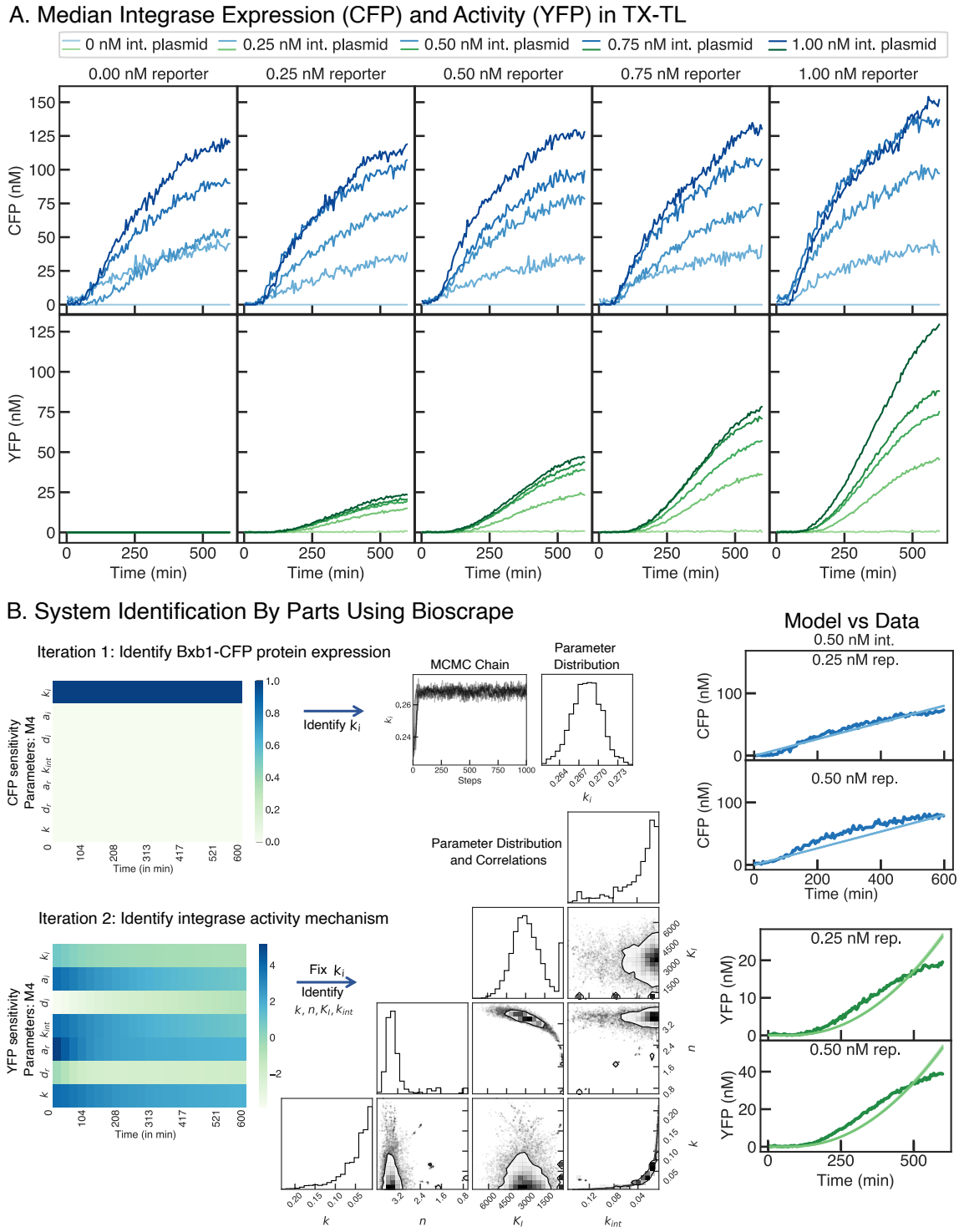


Figure 3: Experimental data and system identification of integrase expression and activity in TX-TL. (A) Median background-subtracted fluorescence data for the integrase circuit in TX-TL. (B) We identify the system by parts, that is, we first select the integrase expression part of the circuit and run sensitivity analysis to find out its identifiable parameters. We observe that k_i is the only sensitive parameter and hence, we run Bayesian inference to identify the posterior parameter distribution for k_i. The model fit alongside the data is shown in the rightmost column. Once we have idenitifed this part, we fix the corresponding parameter, k_i, and run the sensitivity analysis for YFP output. We identify all parameters that YFP is sensitive to. The corner plot shows the posterior distributions of each parameter alongside their correlations with a 75% confidence contours.

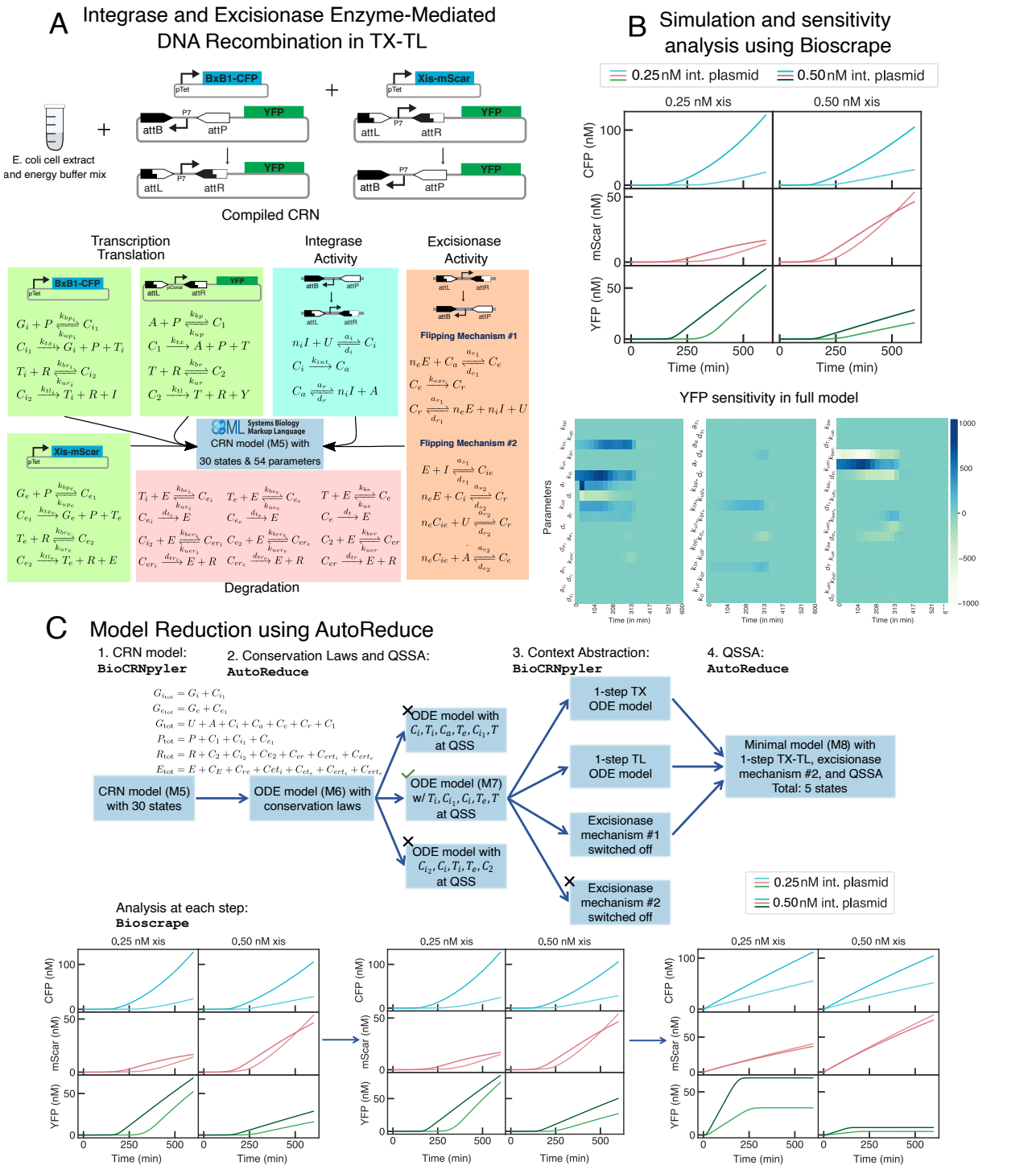


Figure 4: Mathematical models for excisionase expression and activity in TX-TL. (A) Modeling excisionase action using BioCRNpyler. We obtain a CRN for the circuit with both integrase and excisionase in TX-TL by describing the circuit specifications in BioCRNpyler. (B) shows the simulation for the CRN model. We use the identified integrase parameters to predict the excisionase activity. We observe that as more excisionase is expressed, YFP expression falls down. The sensitivity of each parameter in the CRN model with time is shown in the sensitivity analysis heatmaps. (C) Using AutoReduce, we obtain different levels of reduced models for the integrase-excisionase system.

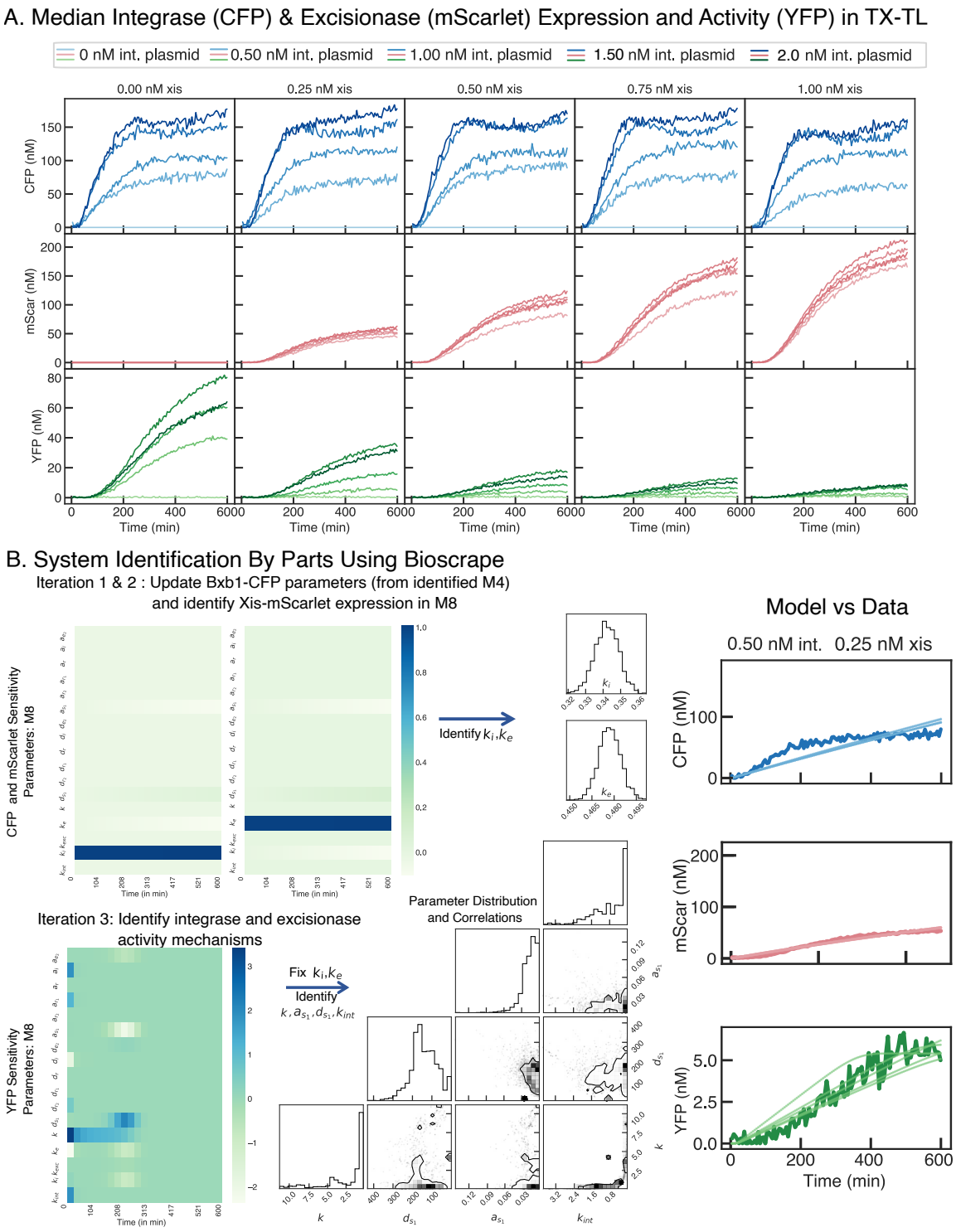


Figure 5: Experimental data and system identification by parts of integrase and excisionase expression and activity in TX-TL. (A) Median background-subtracted fluorescence data for integrase-excisionase circuit in TX-TL. We observe that as more integrase is added, more YFP is expressed until the maximum possible expression is reached at 1.5nM integrase. With higher excisionase levels, we observe a decreasing gradient of YFP levels. (B) To identify the model parameters, we set the previously identified integrase mechanism parameters and update those accordingly to account for various loading effects. In the second iteration, we infer the parameters for the mScarlet expression. Finally, we identify the sensitive set of parameters for YFP. In the right panels, we show the identified parameter distributions and the data plotted alongside the model simulations.

FPCountR: improved analytical methods enable absolute protein quantification

Eszter Csibra
Guy-Bart Stan
e.csibra@imperial.ac.uk
g.stan@imperial.ac.uk
Imperial College London
London, United Kingdom

1 INTRODUCTION AND AIMS

Our aim for this work is to develop a generalisable method for fluorescent protein (FP) calibration, that could be used by any group wishing to calibrate fluorescence readings on microplate readers, from arbitrary or relative units to molecular units. While methods for conducting calibrations with small molecule fluorophores are available, for this work we were interested in using fluorescent proteins directly as calibrants. We reasoned it should be possible to develop a simplified protocol for fluorescent protein purification, that allows users to directly calibrate their instruments using the same FPs as present in their cellular assays, in order to ensure that the values obtained from calibrated experimental measurements directly reflect the number of protein molecules present per cell.

2 METHODS AND RESULTS

An overview of the FPCountR fluorescent protein calibration protocol is illustrated in Fig. 1. Calibrants are produced by a bacterial overexpression and a subsequent preparation step, followed by the production of a dilution series subjected to two assays. The calibration of plate readers with fluorescein has traditionally been conducted using a dilution series of known concentrations of fluorescein, subjected to a fluorescence assay in the plate reader whose calibration is desired (measurement of relative fluorescence units, RFU). The results are used to relate fluorescein molecule number to RFU to obtain a conversion factor, which can in turn be used to convert RFU readouts from experimental data into MEFL units [1]. For protein calibrants produced in-house, the process is identical, but one additional assay is required, for protein concentration determination.

Calibrant preparation does not require protein purification

Initial versions of our FPCount method included a straightforward protocol for the purification of FPs. However, if the requirement for protein purification could be eliminated, the majority of the cost and labour of calibrant preparation could be removed, and it would enable those without prior experience of protein techniques to feel confident in producing reliable calibrants. However, most assays developed for protein concentration determination are general assays that detect all proteins. The requirement for protein concentration determination therefore appeared to necessitate the purification of FPs.

As light absorption is crucial to their performance, all widely-used FPs possess a literature-recorded extinction coefficient ('EC') measurement corresponding to light absorption at their peak ('max') excitation wavelength (Fig. 2A). Suspecting this might be adapted as a novel 'ECmax' assay for protein concentration, we established using purified FPs that this was in fact the best performing assay in terms of both accuracy and sensitivity. Indeed, it had one extra potential advantage over traditional methods: it did not in principle require the FPs to be pure. To investigate this, we lysed cells expressing different FPs (mCherry and mTagBFP2), separated the soluble fractions and concentrated them. Putting these through an ECmax assay and fluorescence assay, we observed it was possible to quantify FPs in crude lysates with high sensitivity, and to obtain almost identical conversion factor values as from our purified FPs (Fig. 2B). Calculated conversion factors were within 20% of expected values from purified FPs and showed high precision (coefficients of variation between 0.06-0.12).

Automated analysis of fluorescent protein calibration

We have developed an automated analysis pipeline to accompany the experimental protocol (based in part on the flopR package [5]). The analytical steps involved in the calculation of conversion factors are illustrated in Fig. 1. After a preparation of a serial dilution, two measurements are taken of the calibrants, one to determine protein concentration and the second to determine fluorescence. In order to obtain protein concentrations using the ECmax assay, the collection of data on the entire absorbance spectrum (200-1000nm), rather than just at the peak absorbance wavelength, is required. This allows for the correction of path length and the removal of light scatter. The raw data is processed by two consecutive R functions: plot_absorbance_spectrum() and get_conc_ECmax().

Helpfully for automation purposes, a database of FPs has recently been developed ([6], https://www.fpbase.org/). As each FP in FPbase is associated with a structured set of properties, including its extinction coefficient (ECmax), we can automate its retrieval using the FPbase API.

Quantification of proteins in *E. coli* as molecules per cell

Using the above calibration, along with a few amendments to take into account autofluorescence and cell-based fluorescence attenuation, it was possible to convert plate reader data from engineered *E. coli* growth assays into molecules per cell. For example, using a p15A vector, we found that mCherry abundance varies between about 900 to 70,000 molecules per cell. While the FP abundances were in the same order of magnitude, mTagBFP2 accumulated to higher levels per cell than mCherry by 3.8-fold on average. Using OD as a measure of the cumulative cellular volume in a culture [7], FP abundance could even be calculated in molar units instead of molecules per cell.

3 CONCLUSION AND SUMMARY

We have shown that it is possible to develop a simplified method for direct fluorescent protein calibration without requiring protein purification, by using a simple absorbance-based assay that allows accurate FP quantification in crude cell lysates. This method, FPCountR, allows for the quantitative characterisation of synthetic *E. coli* parts and circuits using any FP, in molecules per cell or even as molar concentrations. Further details of the method development are available in our preprint [4]. We have made both the experimental protocol [3] at https://www.protocols.io/view/fpcount-protocol-in-lysate-purification-free-proto-bzudp6s6 and analytical workflow [2] at https://github.com/ec363/fpcountr (including a worked example of the package functions), available as open resources.

4 ACKNOWLEDGEMENTS

GBS and EC acknowledge funding from the Royal Academy of Engineering (RAEng CiET 1819\5).

REFERENCES

[1] Beal, J., Haddock-Angelli, T., Baldwin, G., Gershater, M., Dwijayanti, A., Storch, M., Mora, K. d., Lizarazo, M., Rettberg, R., and Contributors, w. t. i. I. S. Quantification of bacterial fluorescence using independent calibrants. *PLOS ONE 13*, 6 (June 2018), e0199432. Publisher: Public Library of Science.

[2] Csibra, E. FPCountR: Fluorescent protein calibration for plate readers, Dec. 2021.

[3] Csibra, E., and Stan, G.-B. FPCount protocol - in-lysate (purification free) protocol. *protocols.io* (2021).

[4] Csibra, E., and Stan, G.-B. FPCountR: Absolute quantification of fluorescent proteins for synthetic biology. Tech. rep., Dec. 2021. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article.

[5] Fedorec, A. J. H., Robinson, C. M., Wen, K. Y., and Barnes, C. P. FlopR: An Open Source Software Package for Calibration and Normalization of Plate Reader and Flow Cytometry Data. *ACS Synthetic Biology 9*, 9 (Sept. 2020), 2258–2266.

[6] Lambert, T. J. FPbase: a community-editable fluorescent protein database. *Nature Methods 16*, 4 (Apr. 2019), 277–278.

[7] Volkmer, B., and Heinemann, M. Condition-Dependent Cell Volume and Concentration of Escherichia coli to Facilitate Data Conversion for Systems Biology Modeling. *PLoS ONE 6*, 7 (July 2011), e23126.

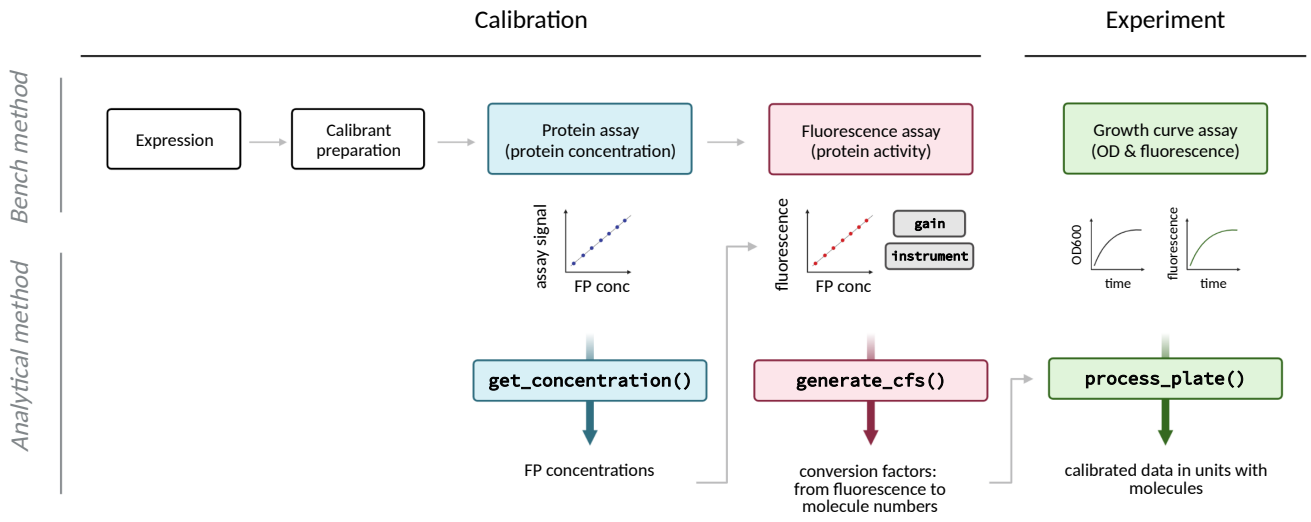


Figure 1: Overview of fluorescent protein calibration workflow using FPCountR. A calibration workflow is described (left), followed by a demonstration of how this calibration can be used to convert experimental data from arbitrary fluorescence units per optical density into molecules per cell (right). The calibration workflow consists of a wet lab protocol (top, available on protocols.io) and an analysis package (bottom, available on GitHub). Figure created with Biorender.com.

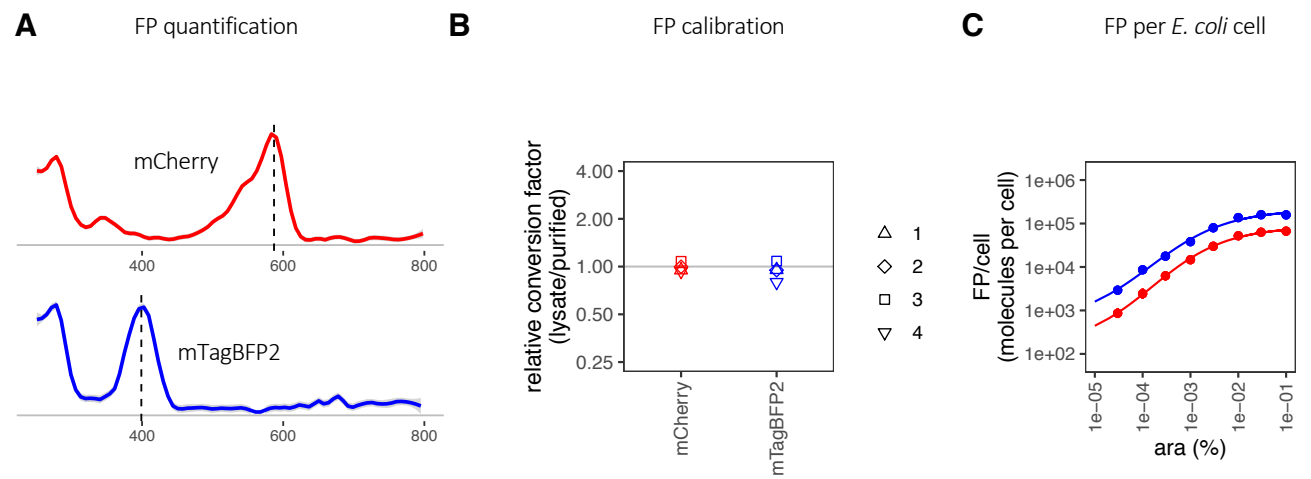


Figure 2: Absolute FP calibration in lysates. A. Absorbance spectra of mCherry (red) and mTagBFP2 (blue) showing FP-specific absorbance at their respective ECmax wavelengths. B. Performance of ECmax assay in calibrations using cell lysates, as compared to using purified proteins. C. mCherry and mTagBFP2 expression, induced at a range of arabinose concentrations, and quantified in a calibrated plate reader. Data was processed with FPCountR.

Towards an automated assay for the quantification of secreted proteins

Sara Napolitano[1,2,†], **Sebastián Sosa-Carrillo**[2,1,†], **François Bertaux**[1,2,3], **Hélène Philippe**[1,4], **Gregory Batt**[2,1*]
[1]Institut Pasteur, Université Paris Cité, F-75015 Paris, France; [2]Inria, 2 rue Simone Iff, F-75012 Paris, France; [3]Lesaffre International, 101 rue de Menin, Marcq-en-Baroeul, France; [4]AgroParisTech, 75005, Paris, France
sara.napolitano@pasteur.fr;{sebastian-ramon.sosa-carrillo,gregory.batt}@inria.fr
f.bertaux@lesaffre.com;helene.philippe@agroparistech.fr

1 INTRODUCTION

One major aim of the Fourth Industrial Revolution or Industry 4.0 is viable and sustainable manufacturing. In this context, bioproduction is playing an important role [5, 8]. It is a field of production which applies biology to produce goods of interest using biological systems as factories [4]. Among all the bioproducts, proteins represent an important class. Indeed, they can be used in many fields of our daily life, including medicine or pharmacy, but also in industry where enzymes can be used to transform a wide range of substrates into a variety of products [8]. Proteins can be easily produced in yeasts. Yeasts provide unique advantages, including the ability to secrete proteins [3, 7]. Protein secretion is a feature of great interest for bioproduction purposes because it could simplify the downstream processes and purification steps [7]. In this framework, however, the real-time tracking of the quantity of secreted proteins is desirable to have good control of the bioprocess, leading to the necessity to automate measurements.

Here, we present an innovative and general pipeline for quantifying secreted proteins based on the use of magnetic beads which allow a simplified method to separate secreted proteins from cells. Specifically, it is based on fusing the POI (Protein Of Interest) with a purification Tag, that binds its partner present on the magnetic beads. Then, the level of bounded protein could be measured either by using fluorescent reporters fused to the POI or through other labeling assays. Thus, by mixing the beads with the cell culture, only the protein dissolved in the culture medium will be captured by the immuno-beads. Then, applying a magnetic field, the beads linked to the POI will be trapped in the vial, whereas the cells will be free to be moved into another vial or wasted. After this washing, the beads will be resuspended and sent to a flow cytometer, which is now able to recognize the particles and read their fluorescence.

*Corresponding author. † Both authors contributed equally to this research.

2 A BEADS ASSAY FOR SECRETED PROTEIN QUANTIFICATION

Principle of the method

A common approach to measure secreted proteins is to fuse them with a fluorescent reporter [2, 6]. However, in these techniques, cells will produce the POI already fused with the reporter becoming fluorescent as well. Therefore, separating cells from the supernatant in which the secreted proteins are present is a key step in these methods. Another weakness of this practice is the detection: the single proteins are too small to be sensed by common instruments, such as the flow cytometer, which we chose for its sensitivity.

To make this technique feasible through the use of the flow cytometer, we developed a protocol in which we make use of agarose-based microbeads covered with antibodies that recognize and bind to a common epitope in the different POIs (Figure 1.B). To this end, we built a genetic construct in which the POI is fused, at the C-terminal end, to a fluorescent reporter (mNeonGreen), in turn, fused, at the C-terminal, to three consecutive copies of the FLAG tag, which is the common region we selected, as shown in Figure 1.A. Subsequently, cells secreting this construct are grown and a sample of this culture is incubated with the magnetic beads covered with anti-FLAG antibodies. During the incubation, the secreted construct will make contact with the agarose-based beads allowing the binding between the FLAG tag and the anti-FLAG antibody, while the proteins still inside the cells will not be visible by the beads. At the end of the incubation time, using a magnetic field, the beads are separated from the cells and passed through the flow cytometer, as shown in Figure 1.C (steps from 2 to 5).

Gating criteria

Despite the washing step to separate beads and cells, a mixture of both will be sent to the instrument for detection. However, the two types of particles have different light scattering properties. Therefore, they can be separated computationally by gating on the forward/side scatter, as shown in Figure 1.C (step 6).

Ratiometric measurement

The binding capacity of the bead might vary among different batches, and even among single beads in the same sample. This can add unwanted variability and heterogeneity to the measurements, reducing the reproducibility among different experiments. Another source of variability among the experiments is the incubation time which might influence the binding of the beads with the proteins into the supernatant. Finally, also the manipulation by the operator can affect the outcome of the protocol, also because of the viscosity of the beads mixture.

To overcome these drawbacks, we implemented a strategy termed ratiometric measurement. It consists of mixing the sample to be measured with a known concentration of a second fluorescent reporter (Figure 1.C - step 1). This reporter should be built exactly as the POI construct to have the same binding properties. Thus, the ratio between the POI fluorescence and the known fluorophore is computed. In this way, due to the fact that all the samples contain the same concentration of the reference protein, the differences between samples are due only to the variation of the POI.

Automation

Since the pipeline has been developed, we are integrating it into the ReacSight strategy [1]. Furthermore, we are automatizing the protocol by using the automatic pipette of the Opentrons OT-2 robot equipped with a magnetic module that allows the separation of the magnetic beads from the cells. This integration will enable us to perform online secretion measurements throughout the experiments.

To this end, it is important not only to develop the code which controls the OT-2 robot but also to adapt the protocol to the requirements of the robot itself. Specifically, the washing step performed by the robot is less efficient than that one performed by a human operator. This means that more cells will be sent to the flow cytometer together with the beads, decreasing the total number of beads analyzed and worsening the statics. Moreover, as explained in [1], our cells are mostly optogenetically-induced and, therefore, the incubation step is performed in the dark, to avoid further secretion during this time. This is not completely possible in the robot. Therefore, we analyzed the effective needs of the incubation for this pipeline. The preliminary results we have on this side are very promising.

3 CONCLUSIONS

We developed a method to measure secretion levels in a quantitative and standard manner, by using fluorescent reporters, monoclonal antibodies bound to magnetic beads and a flow cytometer. This method will allow us to measure secretion levels directly from cell culture, without an *a priori* separation of supernatant. We also assessed the issue of the variability inherent to the characteristics of the beads by performing ratiometric measurements. Finally, we start to automatize the protocol by dealing with the problems of bead concentration and incubation time.

This innovative pipeline can become a useful tool for both research and industrial bioproduction.

REFERENCES

[1] Bertaux, F., Sosa-Carrillo, S., Gross, V., Fraisse, A., Aditya, C., Furstenheim, M., and Batt, G. Enhancing bioreactor arrays for automated measurements and reactive control with reacsight. *Nature communications 13*, 1 (2022), 1–12.

[2] Besada-Lombana, P. B., and Da Silva, N. A. Engineering the early secretory pathway for increased protein secretion in saccharomyces cerevisiae. *Metabolic engineering 55* (2019), 142–151.

[3] Borodina, I., and Nielsen, J. Advances in metabolic engineering of yeast saccharomyces cerevisiae for production of chemicals. *Biotechnology journal 9*, 5 (2014), 609–620.

[4] Camarasa, C., Chiron, H., Daboussi, F., Della Valle, G., Dumas, C., Farines, V., Floury, J., Gagnaire, V., Gorret, N., Léonil, J., et al. Inra's research in industrial biotechnology: For food, chemicals, materials and fuels.

[5] Carvalho, N., Chaim, O., Cazarini, E., and Gerolamo, M. Manufacturing in the fourth industrial revolution: A positive prospect in sustainable manufacturing. *Procedia Manufacturing 21* (2018), 671–678. 15th Global Conference on Sustainable Manufacturing.

[6] Huang, D., Gore, P. R., and Shusta, E. V. Increasing yeast secretion of heterologous proteins by regulating expression rates and post-secretory loss. *Biotechnology and bioengineering 101*, 6 (2008), 1264–1275.

[7] Love, K. R., Dalvie, N. C., and Love, J. C. The yeast stands alone: the future of protein biologic production. *Current opinion in biotechnology 53* (2018), 50–58.

[8] Yu, L.-P., Wu, F.-Q., and Chen, G.-Q. Next-generation industrial biotechnology-transforming the current industrial biotechnology into competitive processes. *Biotechnology Journal 14*, 9 (2019), 1800437.

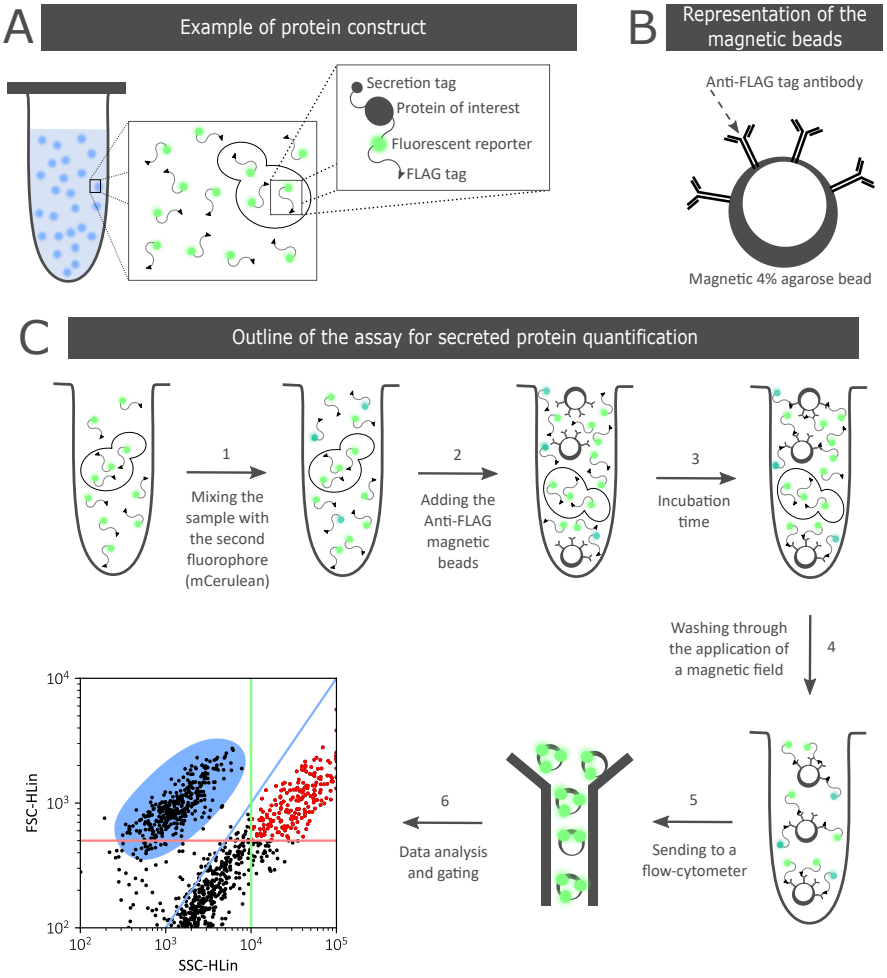


Figure 1: An assay to quantify secreted proteins directly from the culture media. (A) An example of protein construct compatible with this assay. The secreted POI is fused to a fluorescent reporter followed by the FLAG tag purification peptide. (B) The Anti-FLAG antibodies attached to a 4% agarose magnetic bead will allow the capture of the secreted POI. (C) The developed assay is performed according as follows. (i) A sample of the culture medium is mixed with a known concentration of a second fluorophore (i.e. mCerulean) with the same binding capacity. (ii) The solution containing the magnetic beads is added to the mixture. (iii) The 1-hour incubation will allow the beads to capture the secreted proteins. (iv) Beads are partially washed by applying a magnetic field. (v) The washed beads are passed through a flow cytometer. (vi) The data analysis pipeline allows us to further filter the dataset by separating beads from cells, by considering the different light scatter properties. In the forward light scatter (FCS-HLin) versus the side light scatter (SSC-HLin) graph here represented, the blue line separates the ratio that distinguishes beads from cells. The green line restricts the gating to those events higher than the maximal SSC observed for cells. The red line shows the threshold set for the binding capacity of the beads, below which beads show reduced fluorescence. The events within the blue area correspond to the majority of cells, while the events in red are considered as beads.

Rapid gene circuits prototyping with JUMP assembly

Rizki Mardian, Marcos Valenzuela-Ortega, Jin Wong, Christopher French
The University of Edinburgh
{Rizki.Mardian,M.Valenzuela-Ortega,Jin.Wong,C.French}@ed.ac.uk

1 INTRODUCTION

A synthetic biology project, in particular gene circuit design, typically requires the capability to build an extensive library of DNA constructs in order to explore the genetic design space. Recently, Modular Cloning (MoClo) or Type-IIS assembly method has received serious attention in the synthetic biology community for its capacity to facilitate rapid assembly of hundreds of genetic constructs in a standardized and automated fashion.

While there has been a variety of different MoClo standards introduced in the past few years [1, 4, 5], most of them have varying degrees of compatibility and are aimed at specific organisms. Here, we introduce a MoClo-based vector design named JUMP (Joint Universal Modular Plasmids) that aims to improve the existing MoClo standards, by combining the compatibility of commonly adopted vector standards, such as PhytoBricks, BioBricks, and Standard European Vector Architecture standard (SEVA). JUMP is a flexible framework because it is easy to add new features at any assembly stages via two secondary sites flanking the main insertion chassis. As JUMP is built upon SEVA backbones, it is also easy to exchange its origin-of-replications (OriV) and antibiotic resistance (AbR) genes, making it simple to generate a wide variety of vectors with various OriV and AbR combinations. In total, we have created a library of 10 OriV and 5 AbR, and this ever-growing toolkit can still be expanded.

Additionally, we have also developed an open-source software package that can be used to design and create an assembly plan with the simple click of a button, further assisting the quick assembly of a large number of genetic constructs using JUMP. We demonstrate this capability with a case study of building combinatorial assembly of a synthetic gene circuit.

2 RESULTS

Detailed design for the JUMP vectors can be found from [3]. In short, JUMP is a Modular Cloning standard where multiple genetic parts can be assembled through different levels of hierarchy facilitated by standardized 4-base overhang conventions, and alternating Type-IIS restriction enzymes and antibiotics resistance markers, in this case using the combination of BsaI and kanamycin for the odd-level, and BsmBI and spectinomycin for the even-level. Basic JUMP-compatible genetic parts, termed level-0 parts, are stored in a backbone

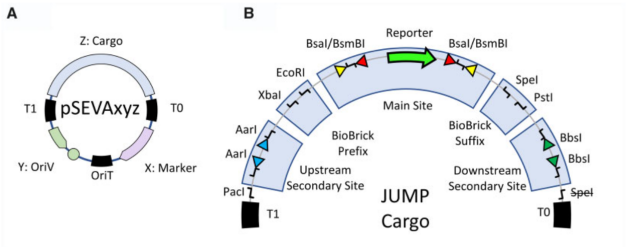


Figure 1: Schematic of JUMP vector. A. Basic SEVA vector consisting of cargo, antibiotic resistance gene, and origin of replication in a modular format. B. JUMP extends a SEVA backbone by having a main modular insertion cassette flanked by two additional secondary sites, separated by BioBricks affix, makes it compatible to BioBricks and SEVA standards. Digestion into the cloning sites can be done with BsaI/BsmBI for the main site, AarI for the upstream site, and BbsI for the secondary site. All 4-base overhangs that dictate the assembly position follows PhytoBrick standard. Image is adapted from [3]

with ampicillin resistance. The secondary sites are located downstream and upstream of the main insertion site, and are divided by the BioBrick prefix and suffix. Cloning into these region can be done at any level using other Type-IIS enzymes, i.e., AarI and BbsI, respectively. Figure 1 illustrates the schematic of the JUMP vector.

On top of a modular vector design, JUMP is accompanied by a dedicated software package that helps facilitate rapid assembly of large-scale genetic constructs. In general, there are 3 supported functionalities:

(1) Domestication of genetic parts from template plasmids or synthetic DNA to make JUMP compatible level-0 parts using a universal acceptor vector. This includes removal of any internal restriction sites for all enzymes required for using JUMP. Users need to provide the sequence annotation of parts they want to create, and any template files if they want to amplify those parts out of other plasmids, otherwise synthetic DNA fragments will be recommended. If parts amplification is preferred, a list of primers and PCR parameters will be generated.
(2) Simulation of JUMP assembly in-bulk. Users need to provide a tabular file consisting the list of genetic constructs assembly plan, and their corresponding plasmid

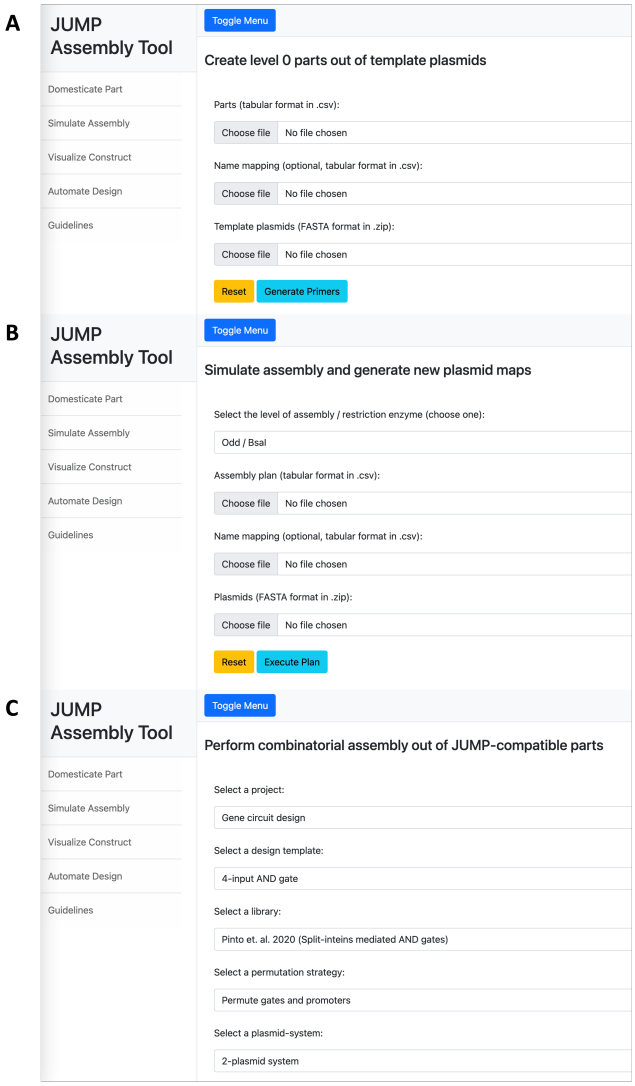


Figure 2: JUMP assembly software user interface. A. Domesticating genetic parts B. Simulating JUMP assembly plan. C. Design automation of a more complex gene circuit. Shown here is an example of automating the design of a 4-input AND gate using split-intein mediated logic AND gate library from [2].

maps. The software runs exhaustive check to match the part overhangs given the selection of a restriction enzyme (i.e., BsaI for the odd-level, and BsmBI for the even-level). If desired, new plasmid maps and their visual SBOL abstraction can be automatically generated, given the perfect match of the part overhangs.

(3) Design automation of higher degree of synthetic gene networks, e.g., combinatorial assembly of gene circuits design. Users can choose a selection of predefined gene circuits (e.g., multiple input AND/OR/NAND/NOR logic gates, full-adder, MUX-and-DEMUX, etc.), gate libraries, and combinatorial assembly strategy (e.g., whether or not to permute the gene positions and/or promoters). The software then will generate an assembly plan that can accommodate required genetic constructs to explore the desired genetic design space.

For each functionality, the JUMP software can generate an experimental plan that can either be performed by hand (e.g., a spreadsheet containing primer design, PCR reaction parameters, pipetting plan, etc.) or be executed by an automated platform (e.g., Python script for Opentrons OT-2). Figure 2 illustrates the user-interface of the JUMP assembly software.

3 DISCUSSION

This work aimed to facilitate rapid assembly of genetic constructs for a large scale synthetic biology project via a modular and flexible vector design paired with a design automation software. We have developed a broad range of vector library (with various OriV and AbR) that are compatible with the PhytoBricks, BioBricks, and SEVA standards. Along with JUMP, we created an open-source tool that can enable instant experimental plans that can be executed directly in the lab, either by hand or via automated liquid handling platforms. We demonstrated the practicality of this approach through a real project in designing a large number of genetic constructs to investigate the design space of synthetic gene circuits. In the future, it will be interesting to see this approach to be adopted in different synthetic biology projects, such as metabolic engineering, biosensors, etc.

REFERENCES

[1] Bernardo Pollak, Tamara Matute, I. N. A. C. C. L. V. V. A. K. V. B. P. v. D. C. L. D. F. F. Universal loop assembly: open, efficient and cross-kingdom dna fabrication. *Synthetic Biology 5* (2020).

[2] Filipe Pinto, Ella Lucille Thornton, B. W. An expanded library of orthogonal split inteins enables modular multi-peptide assemblies. *Nature Communications 11* (2020).

[3] Marcos Valenzuela-Ortega, C. F. Joint universal modular plasmids (jump): a flexible vector platform for synthetic biology.

[4] Simon J. Moore, Hung-En Lai, R. J. R. K. S. M. C. D. J. B. K. M. P., and Freemont, P. S. Ecoflex: A multifunctional moclo kit for e. coli synthetic biology. *ACS Synthetic Biology 5* (2016), 1059–1069.

[5] Sonya V. Iverson, Traci L. Haddock, J. B., and Densmore, D. M. Cidar moclo: Improved moclo assembly standard and new e.coli part library enable rapid combinatorial design for synthetic and traditional biology. *ACS Synthetic Biology 5* (2016), 99–103.

GUARDIAN: Ensemble Detection of Engineering Signatures

Aaron Adler[1], Joel S. Bader[2], Brian Basnight[1], Jitong Cai[2], Elizabeth Cho[2], Joseph H. Collins[4], Yuchen Ge[2], John Grothendieck[1], Kevin Keating[4], Tyler Marshall[1], Anton Persikov[3], Helen Scott[1], Roy Siegelmann[2], Mona Singh[3], Allison Taggart[1], Benjamin Toll[1], Daniel Wyschogrod[1], Fusun Yaman[1], Eric M. Young[4], and Nicholas Roehner[1]

[1]Raytheon BBN, [2]Johns Hopkins University, [3]Princeton University, [4]Worcester Polytechnic Institute
nicholas.roehner@raytheon.com

1 INTRODUCTION

Synthetic biology is pushing the boundaries of what is possible with genetic engineering and has the potential to revolutionize human health and industry. This potential for great benefit, however, is accompanied by the potential for harm due to accidental or malicious release of genetically engineered organisms.

Prior to the IARPA FELIX program, there existed no dedicated biosurveillance tools for detecting genetic engineering, and efforts to evaluate whether DNA was engineered relied heavily on time-consuming analysis by subject matter experts (SMEs). While there were tools for screening DNA synthesis orders for pathogenic or toxin-encoding sequences, these approaches focus on detecting known dangerous sequences [1] and cannot detect engineering that is novel or not directly harmful.

As part of the Guard for Uncovering Accidental Release, Detecting Intentional Alterations, and Nefariousness (GUARDIAN) project, we have developed and integrated tools that use a variety of artificial intelligence (AI) and machine learning (ML) techniques to screen sequence data and individual cells for signatures of engineering. Our whole-genome sequencing analysis system uses an ensemble approach based on the guiding principle that no single approach is likely to be able to detect engineering with perfect accuracy. Critically, ensembling enables GUARDIAN to detect foreign sequence inserts in 13 target organisms with a high degree of specificity that requires no SME review.

2 RESULTS

As part of an independent Test & Evaluation (T&E) during FELIX, we used GUARDIAN to analyze 100 samples listed in Table 1. We calculate that GUARDIAN's ensembled detection of samples with foreign inserts has a sensitivity of 0.62 and a specificity of 0.95, as compared to a slightly higher sensitivity of 0.65 and a considerably worse specificity of 0.8 for SME review (see Figure 1). SME review consisted of team members evaluating the outputs of GUARDIAN's modules and calling a sample engineered on that basis, while ensembling involved automated comparison of module outputs and calling a sample engineered based on a voting algorithm (see Methods). SME review took 5 days to complete, whereas our ensembled approach took less than 2 hours. Finally, rather than sample species or another factor, it appears that insert length and the proportion of engineered cells had the greatest effect on GUARDIAN's performance. If we exclude 19 out of 23 false negative negatives based on GUARDIAN's apparent limits of detection (insert length >1000 bp, dilution $>5.35*10^{-5}$), then its sensitivity rises from 0.62 to 0.9.

3 METHODS

As shown in Figure 2, GUARDIAN ensembles output from six modules. BGAF assembles and taxonomically classifies genomes from Illumina short-read and Nanopore long-read sequencing data, then analyzes these genomes and passes them on to HMM, N-Gram, and BED-DD for their analyses. The outputs of all four assembly-first modules are then ensembled with the outputs of the reads-first modules JHUARDIAN and Targeted Search. GUARDIAN's approach to ensembling involves grouping modules' output regions of interest (ROIs) for a sample based on their pairwise sequence similarity. ROIs are DNA sub-sequences that have been identified by a module as potentially engineered. If two ROIs are sufficiently similar, then their groups are merged. GUARDIAN then calls a sample engineered if it has at a group of at least two ROIs produced by different modules.

JHUARDIAN

JHUARDIAN is a bioinformatics pipeline that includes taxonomic classification with Kraken [7], read mapping with Bowtie [3], read assembly with MEGAHIT [4], and annotation with BLAST. JHUARDIAN can be used to filter out reads that map to the reference genomes for organisms present in a sample, or it can be used to filter in reads that map to plasmids in UniVec. JHUARDIAN calls annotated regions engineered based on taxonomic mismatches with the host and keywords such as "synthetic," "artificial," or "cloning."

Targeted Search

Targeted Search is a read analysis pipeline that uses the Burrows-Wheeler Aligner (BWA) [5] to map reads to a curated list of target sequences commonly used in genetic engineering and to the reference genomes for our target organisms (Table 1). Targeted Search calls a sample engineered if it contains sufficient reads mapping to its target sequences.

BGAF

The BBN Genetic Anomaly Filter (BGAF) pipeline includes whole genome assembly with Abyss [6] from short reads and Prymetime [2] from short and long reads, taxonomic classification and read mapping with BBN's FAST-NA tool, and annotation with BLAST. FAST-NA uses a probabilistic data structure known as a Bloom filter to determine whether short sub-sequences in sample contigs are definitely unnatural or may map to the reference genome for a target organism in Table 1. BGAF then constructs ROIs from any unnatural subsequences and calls them engineered based on their BLAST results against NCBI (looking for suspicious keywords) and the target sequences used by Targeted Search.

HMM

The HMM pipeline uses Hidden Markov Models (HMMs) constructed from the aligned reference genomes of our target organisms (Table 1) to compute HMM scores for sample assemblies and flag ROIs with high scores. The HMM pipeline calls samples engineered based on their overall HMM score and the results of BLASTing their ROIs against UniVec, filtering hits against an exclusion list of plasmids that matched the genomes of our target organisms.

N-Gram

The N-Gram pipeline uses n-gram language models constructed from the reference genomes of our target organisms (Table 1) to compute sequence entropy scores for sample assemblies and flag ROIs with high scores. N-Gram filters ROIs by BLASTing them against NCBI and removing any that match the sample host taxon. N-Gram then calls the remaining ROIs engineered based on their entropy score and their BLAST results against UniVec and the target sequences used by Targeted Search.

System Requirements

We created Docker containers for each of our system modules to make the system easier to run. Most modules are written in Python, though some modules such as BGAF use libraries written in C or Perl. To process 100 samples of varying size, the system required 14,900 CPU hours (< 72 hr elapsed time with parallel processing of samples) and 17.9 TB of disk space. A significant portion of this was required for genome assembly (~11,600 CPU hr and ~4.6 TB of disk space).

4 DISCUSSION

Complex metagenomic samples in which engineering signatures were highly diluted by diverse natural DNA sequences proved to be the most difficult for GUARDIAN to detect. In this case, the most effective modules were JHUARDIAN and Targeted Search, in part because these modules could analyze raw sequencing reads directly without requiring a potentially intractable genome assembly as a pre-processing step. To better handle complex samples in the future, new tools are needed that can similarly analyze reads first. In addition, it would be beneficial to develop enhanced signature-based detection of engineering, since the large space of unsequenced natural DNA can limit the effectiveness of anomaly-based approaches to detect unnatural sequences.

Beyond engineering detection, our work could also be extended to "reverse compile" engineering signatures into putative design specifications. Such a biological decompiler would seek to match known design motifs to predicted sequence features such as promoters and protein coding sequences. This could enable detection of engineering based on the relative order and organization of different types of sequence features, even if their individual engineering signatures are small. It would also enable investigation into the purpose or attribution of an organism's engineering.

Acknowledgements

This work was supported by the IARPA Finding Engineering-Linked Indicators (FELIX) award HR0011-15-C-0084. DISTRIBUTION STATEMENT A: Approved for public release. Distribution is unlimited. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

REFERENCES

[1] Carter, S., and Friedman, R. DNA synthesis and biosecurity: lessons learned and options for the future. *J Craig Venter Institute, La Jolla, CA* (2015), 1–28.

[2] Collins, J. H., et al. Engineered yeast genomes accurately assembled from pure and mixed samples. *Nat. Commun. 12* (2021), 1485.

[3] Langmead, B., and Salzberg, S. L. Fast gapped-read alignment with bowtie 2. *Nat. Methods 9* (2012), 357–359.

[4] Li, D., Liu, C.-M., Luo, R., Sadakane, K., and Lam, T.-W. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics 31*, 10 (01 2015), 1674–1676.

[5] Li, H., and Durbin, R. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics 25*, 14 (2009), 1754–1760.

[6] Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J., and Birol, I. Abyss: a parallel assembler for short read sequence data. *Genome research 19*, 6 (2009), 1117–1123.

[7] Wood, D. E., and Salzberg, S. L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol. 15* (2014), R46.

Table 1: IV&V Organisms and # of Samples

Organism	w/ Insert	w/o Insert
A. thaliana	1	1
B. subtilis	7	3
C. freundii	2	1
E. coli	11	3
Influenza A	1	2
O. oncorhynchi	0	1
O. sativa	3	1
P1 Phage	0	1
P. aeruginosa	2	4
P. putida	3	1
Rabies lyssavirus	0	1
R. toruloides	2	1
S. cerevisiae	10	4
S. enterica	8	1
T7 Phage	0	1
Y. lipolytica	2	1
Bacteria mixture	0	3
Bacteria+yeast mixture	0	6
Metagenomic soil	4	2
Metagenomic gut	4	2

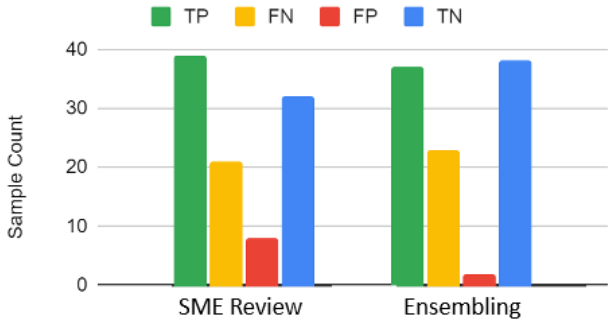


Figure 1: True positive (TP), false negative (FN), false positive (FP), and true negative (TN) sample counts and for detecting samples containing sequence inserts using SME review (left) and automated ensembling (right).

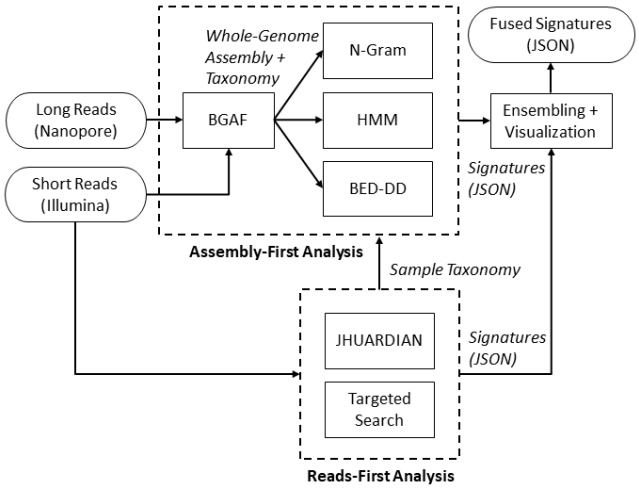


Figure 2: Overview of the GUARDIAN system.

SIMPLIFE: An automated pipeline for inserting functional domains into globular proteins

Georgie Hau Sorensen
School of Biological Sciences,
University of Bristol,
Bristol, United Kingdom
georgie.hausorensen@bristol.ac.uk

Fabio Parmeggiani
Schools of Biochemistry and
Chemistry, University of Bristol,
Bristol, United Kingdom
fabio.parmeggiani@bristol.ac.uk

Thomas E. Gorochowski
School of Biological Sciences,
University of Bristol,
Bristol, United Kingdom
thomas.gorochowski@bristol.ac.uk

INTRODUCTION

Structure-guided design holds much promise for engineering biology with software suites like Rosetta [6] having already proven useful for the *de novo* design of a range of different proteins. This includes antibodies with engineered specificity towards targets of interest [1] and custom three-dimensional protein structures [9]. A particularly interesting application of structure-guided protein design is finding optimal locations for the insertion of functional domains, or even whole enzymes, into the backbones of molecular machines like polymerases. Such an approach has already been employed to augment the function of Cas enzymes, improving the editing capabilities of CRISPR-Cas systems [7].

Historically, structure-guided design has been limited by a dependence on experimental structure data, such as from Cryogenic Electron Microscopy (Cryo-EM) or X-ray crystallography, which is expensive to produce [4]. A solution to this issue is to use recent advances in computational protein structure prediction. For example, AlphaFold [4] is now able to accurately infer the structure of many proteins from sequence alone. Using AlphaFold-derived structures for rational design of proteins could vastly expand the range of possible designs and is especially valuable when working with proteins that are difficult to crystallise.

Here, we present the Structurally Informed Modifications of Protein Loops for Insertions of Functional Entities (SIMPLIFE) workflow which integrates AlphaFold-predicted structures with the protein grafting capabilities of Rosetta to help automate the augmentation of proteins with new functions.

THE SIMPLIFE WORKFLOW

SIMPLIFE was created to aid in the insertion of functional domains into a globular protein backbone, while retaining function of both insert and backbone. Designing an insertion would usually require some parameter optimisation specific for each input structure, as well as some prior knowledge on which parts of the backbone can be altered. With SIMPLIFE, prior knowledge is not needed because the workflow involves a step to rapidly graft and score the insert domain into every single loop of the backbone structure. SIMPLIFE is therefore an agnostic screening tool that returns only loops for which the insertion is energetically favourable. SIMPLIFE is implemented by four shell scripts that run XML scripts for Rosetta or Python code for AlphaFold for a particular input. The major steps in this process are shown in Figure 1. Two files are required as input, one containing the backbone protein, and the other the domain to insert. Either file can be submitted in a PDB format (if structural data is available) or as a FASTA protein sequence file. For FASTA formatted inputs, AlphaFold is used to predict a 3D structure of the input. All structures go through a minimization and relaxation step, along with some Rosetta-specific preparation of the input structures. The user is then prompted to specify which region of the input insert domain structure should be grafted into the backbone protein. Grafting is then performed by the CCDEndsGraftMover [1] via Rosettascript [3]. Filters are applied to the resulting pose to ensure the insert has favourable surface availability and overall energy scores. Structures passing this stage are further refined by remodelling the region of the grafted structure most proximal to the insert, using FastDesignMover [2]. Since the grafting process is generally unable to obtain loop closures, the Rosetta tool ChainBreakFilter are used to score the bond-lengths between backbone and insert. These Rosetta-derived structures are then submitted to AlphaFold to predict a final structure without chain breaks. The AlphaFold structure is also compared to the Rosetta generated structure to validate the output.

APPLICATION TO T7-RNA POLYMERASE

T7 RNA polymerase (T7 RNAP) is commonly used for transcribing heterologous genes due to its high processivity and orthogonality to the endogenous gene expression machinery of production hosts. For these reasons, we chose T7 RNAP as an input structure to demonstrate the SIMPLIFE workflow. PDB entry '1h38' was used as a high-resolution crystal structure for T7 RNAP [8], and as an insert domain we selected DogTag, a protein tag that has been optimised for strong in-loop protein-protein interactions to the DogCatcher binding protein [5].

Of the 883 residues in the '1h38' structure, SIMPLIFE identified 297 residues as potential insertion sites that were not annotated as part of any secondary structure. Inserting the

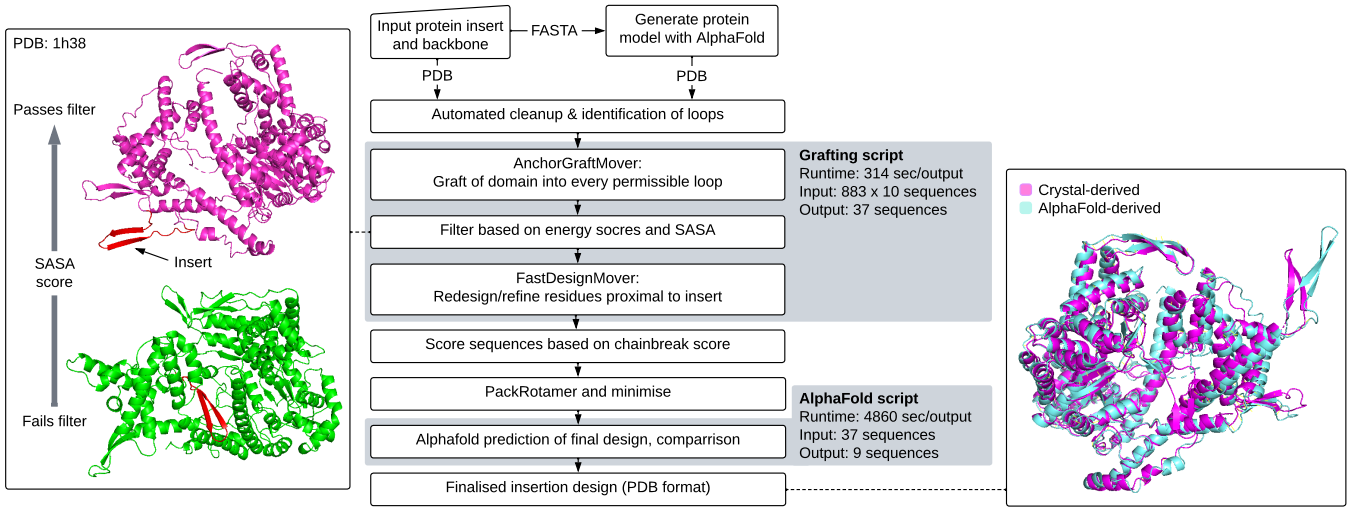


Figure 1: Overview of the SIMPLIFE workflow. (Middle) Schematic of the major workflow steps. The two main scripts for the workflow are shown in grey boxes and their respective runtimes for Oracle Cloud Infrastructure shown. (Left) At the filter stage of SIMPLIFE, scoring the Solvent-Accessible Surface Area (SASA) can be used to evaluate the degree to which the DogTag domain (marked in red) is buried within the polymerase structure itself. A passing structure from the filter (purple) will result in the DogTag domain being able to freely interact on the surface of the polymerase, compared to a non-passing structure (green). (Right) The highest-scoring structure using a crystal structure-derived input protein (magenta) and AlphaFold-derived input (cyan). The prediction in both cases favour insertion at residue 97.

DogTag domain into each site yielded 37 different structures across 19 different insert positions that passed the filter criteria. An optional filter was used to score the solvent-accessible surface area (SASA) of the insert (filter passed if score > 2200), which was implemented to ensure that the position of the DogTag would be on the surface of the engineered polymerase to ensure it was available for binding to DogCatcher. These 37 structures were modelled using AlphaFold, yielding a total of 9 final output structures from the workflow with an acceptable similarity between the Rosetta and AlphaFold outputs (RMSD < 5).

SIMPLIFE was run on a high performance computing (HPC) cluster using the Oracle Cloud Infrastructure (OCI), provided and funded by the Oracle for Research program to allow for GPU acceleration. Using the HPC cluster, the total runtime for each successful structure was around 5400 seconds, with specific runtimes for most time-consuming scripts shown in Figure 1.

Running the SIMPLIFE workflow on either crystal-derived or AlphaFold-derived input structures led to near identical results. For 1h38, the same insert sites were suggested for both inputs, with a slightly lower energy score of -2681.7 for the best candidate from the AlphaFold-derived model. Furthermore, the AlphaFold-derived structure did not contain any residue gaps, unlike the crystal-derived structure. These features make the AlphaFold-derived input structure a favourable alternative.

CONCLUSION

We have shown that SIMPLIFE is able to identify multiple permissible sites for domain insertion into T7 RNA polymerase and that both crystal and AlphaFold-derived input structures yield similar results. We are currently experimentally verifying these computationally generated designs.

REFERENCES

[1] Adolf-Bryfogle, J., Kalyuzhniy, O., Kubitz, M., et al. Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS Computational Biology 14* (4 2018).

[2] Bhardwaj, G., Mulligan, V. K., Bahl, C. D., et al. Accurate de novo design of hyperstable constrained peptides. *Nature 538* (2016), 329–335.

[3] Fleishman, S. J., Leaver-Fay, A., Corn, J. E., et al. Rosettascripts: A scripting language interface to the rosetta macromolecular modeling suite. *PLoS ONE 6* (2011).

[4] Jumper, J., Evans, R., Pritzel, A., et al. Highly accurate protein structure prediction with alphafold. *Nature 596* (8 2021), 583–589.

[5] Keeble, A. H., Yadav, V. K., Ferla, M. P., et al. Dogcatcher allows loop-friendly protein-protein ligation. *Cell Chemical Biology* (7 2021).

[6] Leman, J. K., Weitzner, B. D., Lewis, S. M., et al. Rosetta: recent methods and frameworks. *Nature Methods* (2020), 665–680.

[7] Meaker, G. A., Hair, E. J., and Gorochowski, T. E. Advances in engineering CRISPR-Cas9 as a molecular Swiss Army knife. *Synthetic Biology* (2020).

[8] Tahirov, T. H., Temiakov, D., Anikin, M., et al. Structure of a t7 rna polymerase elongation complex at 2.9 a resolution, 2002.

[9] Yeh, C. T., Brunette, T. J., Baker, D., et al. Elfin: An algorithm for the computational design of custom three-dimensional structures from modular repeat protein building blocks. *Journal of Structural Biology* (2018), 100–107.

Galaxy-SynBioCAD: Automated Pipeline for Industrial Biotechnology

Joan Hérisson*
Genomics Metabolics, Genoscope, François Jacob Institute, CEA, CNRS, Univ Evry, Université Paris-Saclay

Thomas Duigou*
Kenza Bazi-Kabbaj, Mahnaz Sabeti Azad, Manish Kushwaha, Jean-Loup Faulon
Université Paris-Saclay, INRAE, AgroParisTech, Micalis

1 INTRODUCTION

Computation has become an essential tool in life science research. Synthetic biology, metabolic engineering and industrial biotechnology make no exception to that trend. As for genetic circuits, there are plenty of software tools to assist the biosynthetic pathway design process [20]. Briefly, from a given target compound and a given chassis strain, the first step consists of doing retrosynthesis [2, 4, 8, 9, 17, 19, 29] to find metabolic reactions that link the target compound to the native metabolites of the host strain. The result of retrosynthesis is a metabolic map and there is a need in a second step to enumerate the pathways linking the chassis metabolites to the target [5, 15, 21, 22, 24]. The third step is to find the most promising enzyme sequences catalyzing the metabolic reactions. Once the pathways have been annotated with enzyme sequences, they can be ranked in a fourth step with some metrics [4, 7, 16, 19]. In addition to the enzyme identities, there are multiple layout solutions and settings to engineer the top-ranked pathways. The fifth step deals with promoters, order and RBS strength combinatorics by making use of tools such as the RBS calculator [26] to compute RBS sequences for different strengths, and design of experiments (DoE) [6, 23] to sample the space of possible constructs, which can be very large. Several computational tools can be used to perform a sixth and last step of DNA assembly design before constructing the pathways [1, 14, 18, 28]. Engineered pathways are generally evaluated using HPLC or mass spectrometry analyses [11, 25]. Considering the above, we are clearly at a stage where the pathway engineering process is not that far from being fully driven by computer software products. However, there are several hurdles that prevent this from happening even for tools covering pathway design only. First, the tools are not easily findable, they are stored in different places and the keywords to search online are not obvious. Secondly, some of the tools are difficult to access, some requiring registration, purchase, or access fees. Thirdly, almost none of the tools are interoperable and cannot be chained one after another to ensure that computational experiments are communicated well, and hence reproducible. Lastly, and perhaps most problematic for wider acceptance, the tools can be difficult to comprehend, requiring a level of expertise that limits their use by a large community. Scientific workflows help to address these issues by providing an open, web-based platform for performing findable and accessible data analyses linked to experimental protocols for all scientists irrespectively of their informatics expertise, along with interoperable and reproducible computations regardless of the particular platform that is being used [31]. Indeed, without programming skills, scientists that need to use computational approaches are impeded by difficulties ranging from tool installation to determining which parameter values to use, to efficiently combining and interfacing multiple tools together in an analysis chain. Among existing workflow platforms, Galaxy is a system originally developed for genome analysis [12] which now includes more than 8500 tools that can be found in the public ToolShed [3].

2 RESULTS

Here, we introduce the Galaxy-SynBioCAD web portal, the first Galaxy set of tools for synthetic biology, metabolic engineering and industrial biotechnology, fully integrated into automatic pipelines. The portal is a growing community effort where developers can add new tools and users can evaluate the tools performing design for their specific projects. The tools and workflows currently shared on the Galaxy-SynBioCAD portal cover an end-to-end metabolic pathway design and engineering process from the selection of strain and target to automated DNA parts assembly and strain transformation. All workflows are available from our instance (galaxy-synbiocad.org) or any instance of Galaxy (by installing tools from the ToolShed). To develop an integrated ecosystem, we selected software applications from among the computational tools mentioned above. Several criteria were used for this selection: (i) the tools needed to be relevant for pathway design and engineering, (ii) be published, (iii) open-source under MIT, GNU GPL, or related licenses, (iv) well documented and deposited in GitHub, (v) make use of standard input/output, and (vi) exist as a standalone command-line tool. Within a workflow, each tool connected to one or more tools must share common file format for data exchange, i.e. each output file of a tool has to be compatible with the input file format of downstream tools in the

*Both authors contributed equally to this research.

workflow. The file format relies on the nature of the data (e.g. metabolic model, construct design, ranked properties) and the implementation choice made for each tool. Among the standard formats used, some are rather generic (CVS, TSV, JSON) while others are more specific to a scientific field (e.g. SBOL, SBML). The selected tools can be divided in three categories: (i) those aimed at finding pathways to synthesize heterologous compounds in chassis organisms (RetroRules, RetroPath2.0, RP2Paths, rpCompletion), (ii) thoses aimed at evaluating and ranking pathways (rpThermo, rpFBA, rpReport, rpViz, rpScore) and (iii) those related to genetic design and engineering (Selenzyme, SbmlToSbol, PartsGenie, OptDOE, DNA Weaver, LCR Genie, rpBASICDesign, and DNABot). Following FAIR principles [26], all selected tools are open source with code available on GitHub and installable through the Conda package manager [29]. Therefore, any user can install the tools needed on their own computer and run these as standalone programs or chain them together to process more complex calculations. To go further in chaining tools, three types of Galaxy workflows are available on the Galaxy-SynBioCAD portal: 1) a Retrosynthesis workflow to enumerates the pathways enabling the synthesis of a given target chemical in a host chassis organism (Figure 1); 2) a Pathways analysis workflow to score and rank the pathways produced through Retrosynthesis based on multiple criteria (Figure 2); 3) two Genetic design and engineering workflows (Figures 3, 4) that produce assembly plans (Golden Gate [10], Gibson [13], LCR [30] and BASIC [27]) for plasmids encoding the pathways generated by the Retrosynthesis or Pathway analysis workflows. We illustrated our workflows by designing and engineering a library of 88 pathway variants designed to produce lycopene in E. coli DH5-alpha on Opentrons liquid handlers. The Galaxy workflows were executed from four different workstations (in France and abroad) demonstrating the ability of the Galaxy-SynBioCAD portal to run workflows (including robot drivers with different labwares) at different sites, and consequently the possibility of completing multi-partners design and engineering projects.

REFERENCES

[1] https://synthace.com/antha-platform.

[2] Algfoor, Z. A., et al. Identification of metabolic pathways using pathfinding approaches: a systematic review. 87–98. Number: 2 Place: Oxford Publisher: Oxford Univ Press WOS:000397205400004.

[3] Blankenberg, D., Galaxy Team, et al. Dissemination of scientific software with galaxy ToolShed. 403. Number: 2.

[4] Campodonico, M. A., et al. Generation of an atlas for commodity chemical production in escherichia coli and a novel pathway prediction algorithm, GEM-path. 140–158.

[5] Carbonell, P., et al. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. 66.

[6] Carbonell, P., et al. Efficient learning in metabolic pathway designs through optimal assembling. 7–12. Number: 26.

[7] Carbonell, P., et al. XTMS: pathway design in an eXTended metabolic space. W389–W394. Number: W1.

[8] Delépine, B., et al. RetroPath2.0: A retrosynthesis workflow for metabolic engineers. 158–170.

[9] Duigou, T., du Lac, M., Carbonell, P., and Faulon, J.-L. RetroRules: a database of reaction rules for engineering biology. D1229–D1235. Number: D1 Publisher: Oxford Academic.

[10] Engler, C., et al. A one pot, one step, precision cloning method with high throughput capability. e3647. Number: 11.

[11] Giacomoni, F., et al. Workflow4metabolomics: a collaborative research infrastructure for computational metabolomics. 1493–1495. Number: 9 Publisher: Oxford Academic.

[12] Giardine, B., et al. Galaxy: A platform for interactive large-scale genome analysis. 1451–1455. Number: 10.

[13] Gibson, D. G., et al. Enzymatic assembly of DNA molecules up to several hundred kilobases. 343–345. Number: 5.

[14] Gupta, V., et al. BioBlocks: Programming protocols in biology made easier. 1230–1232. Number: 7 Publisher: American Chemical Society.

[15] Hadadi, N., et al. Enzyme annotation for orphan and novel reactions using knowledge of substrate reactive sites. 7298–7307. Number: 15.

[16] Hafner, J. Modeling, predicting an mining metabolism at atom-level resolution.

[17] Hatzimanikatis, V., et al. Exploring the diversity of complex metabolic networks. 1603–1609. Number: 8.

[18] Keller, B., et al. Aquarium: The laboratory operating system v2.6.0.

[19] Kumar, A., et al. Pathway design using de novo steps through uncharted biochemical spaces. 184. Number: 1 Publisher: Nature Publishing Group.

[20] Lin, G.-M., Warden-Rothman, R., and Voigt, C. A. Retrosynthetic design of metabolic pathways to chemicals not found in nature. 82–107.

[21] Mellor, J., et al. Semisupervised gaussian process for automated enzyme search. 518–528. Number: 6.

[22] Rahman, S. A., et al. EC-BLAST: a tool to automatically search and compare enzyme reactions. 171–174. Number: 2 Publisher: Nature Publishing Group.

[23] Roehner, N., et al. Double dutch: A tool for designing combinatorial libraries of biological systems. 507–517. Number: 6 Publisher: American Chemical Society.

[24] Ryu, J. Y., et al. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. 13996–14001. Number: 28 Publisher: National Academy of Sciences Section: Biological Sciences.

[25] Röst, H. L., et al. pyOpenMS: a python-based interface to the OpenMS mass-spectrometry algorithm library. 74–77. Number: 1.

[26] Salis, H. M. The ribosome binding site calculator. 19–42.

[27] Storch, M., et al. BASIC: A new biopart assembly standard for idempotent cloning provides accurate, single-tier DNA assembly for synthetic biology. 781–787. Number: 7 Publisher: American Chemical Society.

[28] Storch, M., et al. DNA-BOT: a low-cost, automated DNA assembly platform for synthetic biology. Number: 1 Publisher: Oxford Academic.

[29] Tyzack, J. D., et al. Exploring chemical biosynthetic design space with transform-MinER. 2494–2506. Number: 11 Publisher: American Chemical Society.

[30] Wiedmann, M., et al. Ligase chain reaction (LCR)–overview and applications. S51–S64. Number: 4 Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

[31] Wilkinson, M. D., et al. The FAIR guiding principles for scientific data management and stewardship. 1–9. Number: 1.

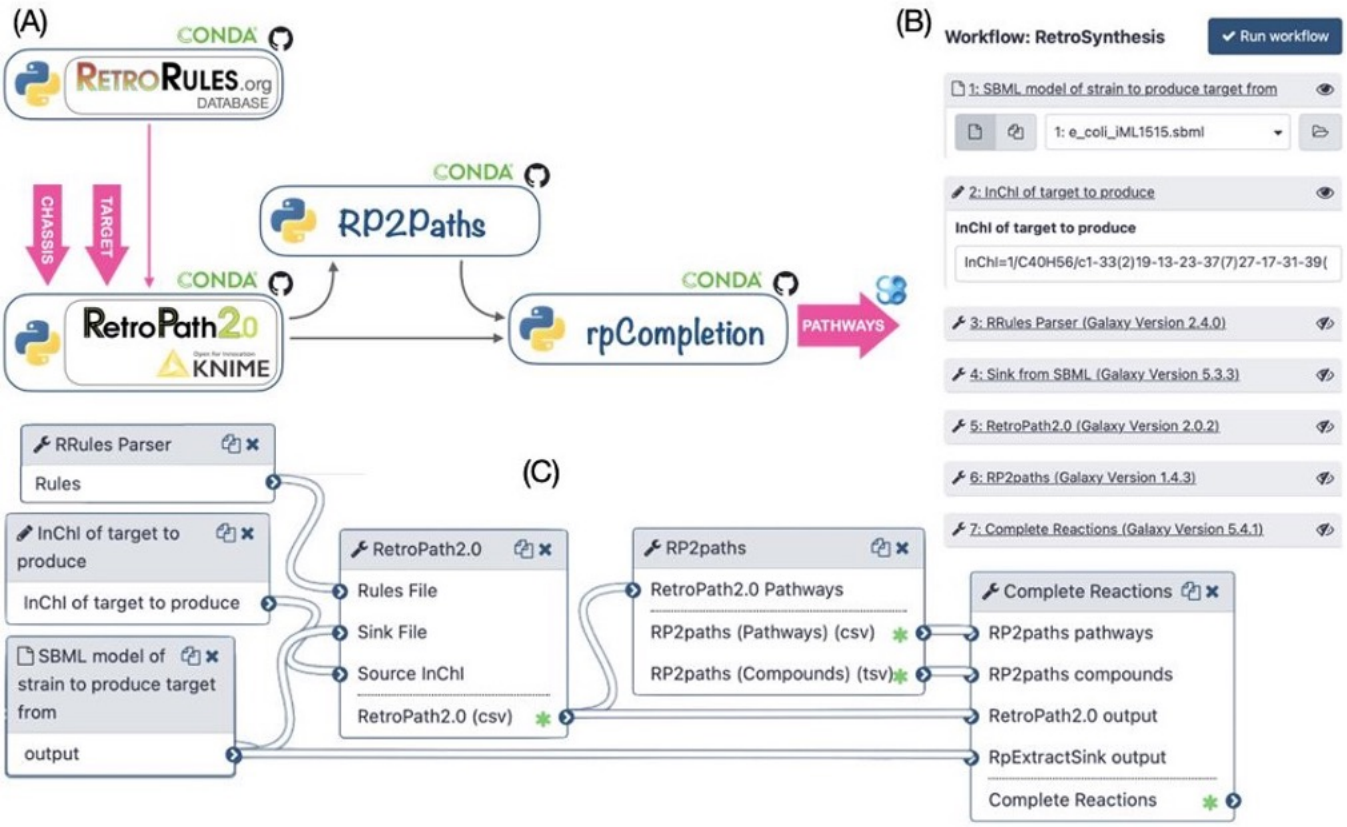


Figure 1: RetroSynthesis and Pathway Enumeration workflow. (A) The workflow of tools for retrosynthesis and pathway enumeration. Tools can be chained manually by running each tool one after the other in a command-line terminal. Outputs of each tool (files) can be directly given as inputs of the others without any other processing. (B) The workflow menu at runtime in the Galaxy interface. The user specifies the genome scale SBML model of the host organism and the InChI structure of the target molecule. The user can also change the default settings for each tool by clicking on its name. The RetroRules entry has been set as default for convenience. The workflow generates a collection of heterologous pathways for target production in separate SBML files. (C) The workflow as displayed in the Galaxy workflow Editor.

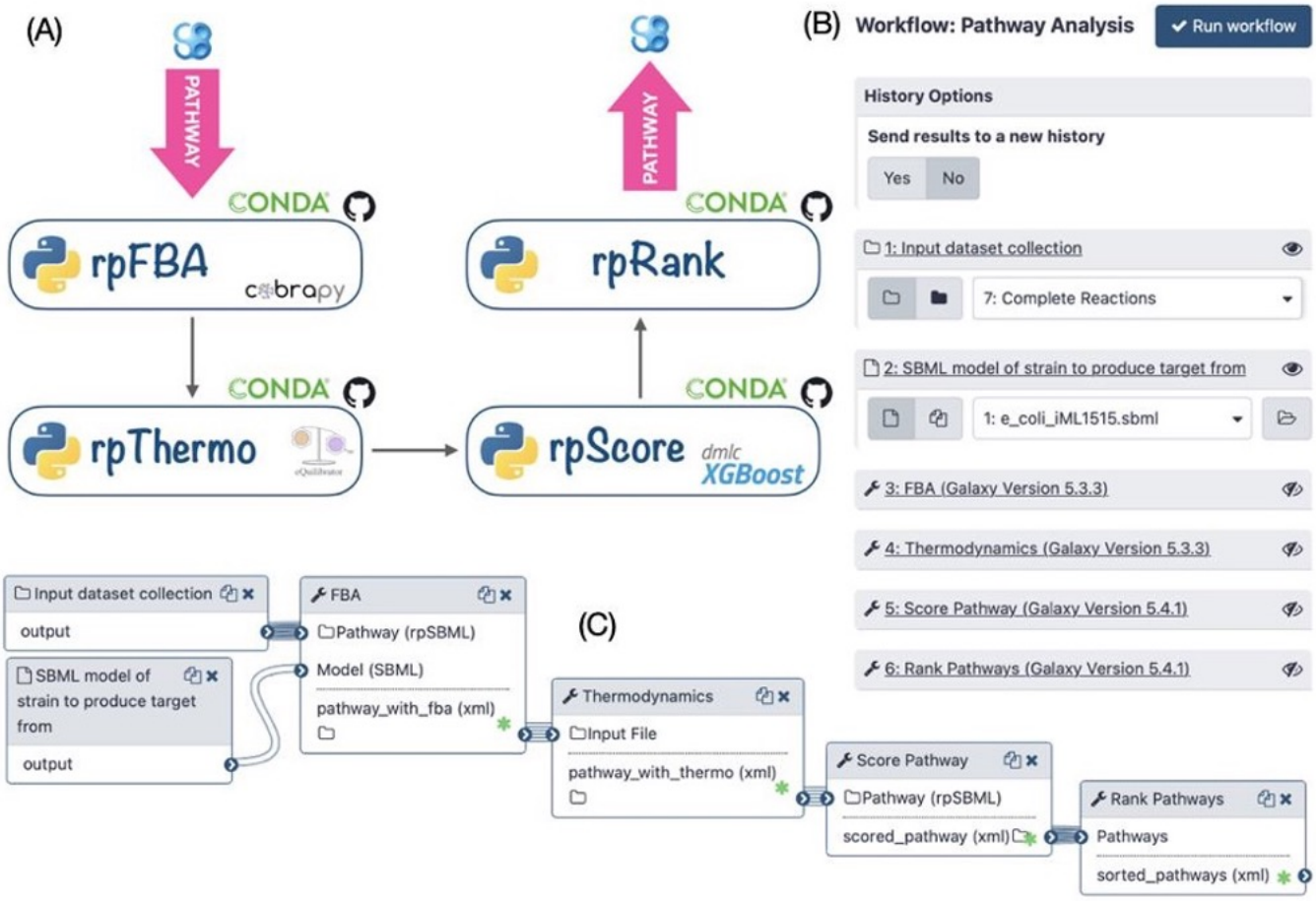


Figure 2: Pathway Analysis and Ranking. (A) Workflow of tools for pathway analysis and ranking. (B) The workflow menu upon executing it within the Galaxy interface. The user specifies the GEM SBML model of the host organism and the set of pathways to rank (here we can choose the output of Retrosynthesis workflow). The user can also modify default parameters of each tool by clicking on its name. The workflow generates a collection of heterologous pathways which are scored and ranked. (C) The workflow as displayed in the Galaxy workflow manager.

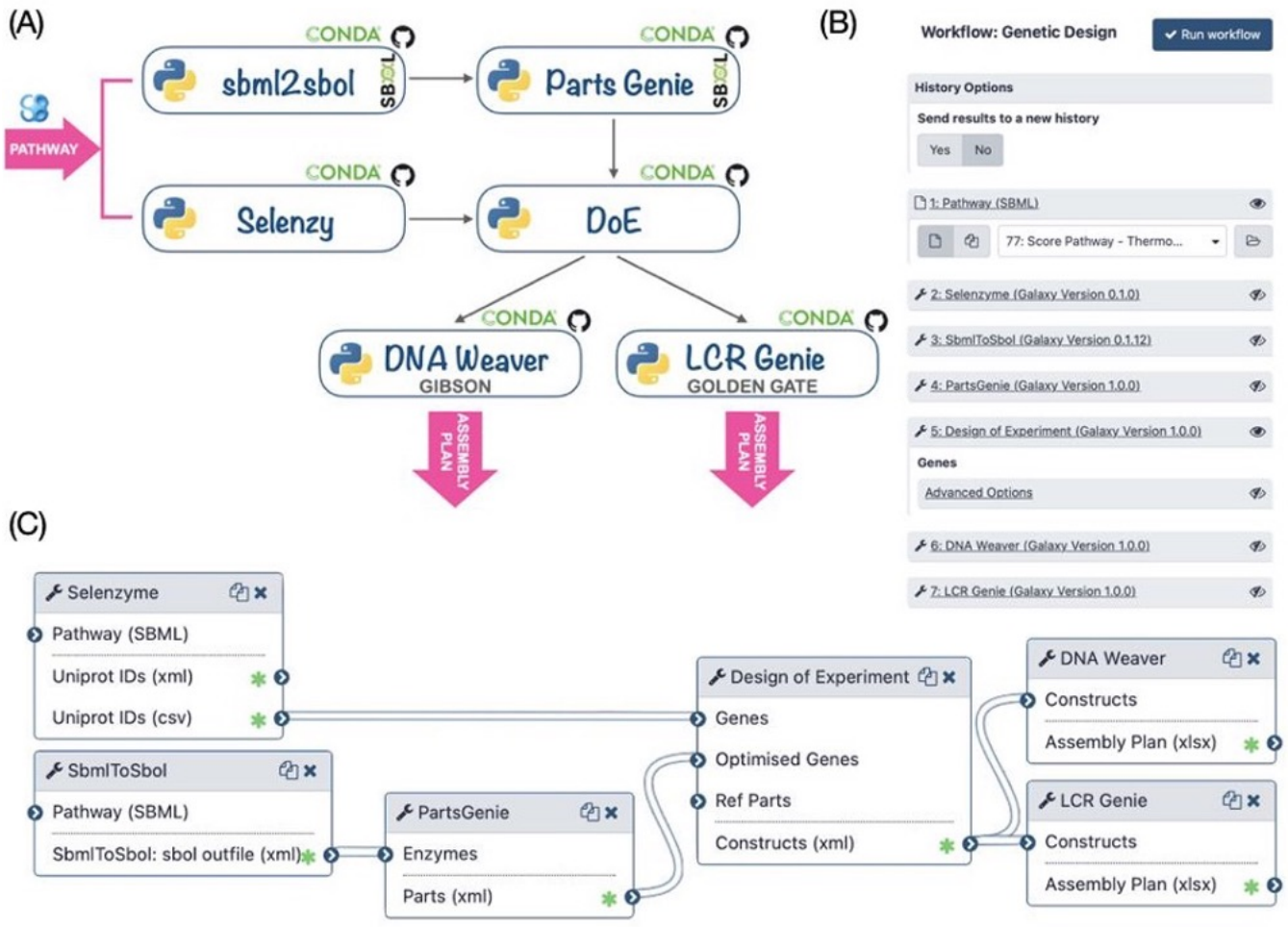


Figure 3: Genetic Design (Golden Gate, Gibson and LCR). (A) Tools can be chained manually by running each tool one after the other in a command-line terminal. Outputs of each tool can be directly given as inputs of the others without any other processing. (B) The workflow menu upon executing it through the Galaxy interface. The user specifies the pathway (SBML) that he wishes to build. The workflow generates assembly plans by using LCR (LCR Genie) or Golden Gate or Gibson (DNA Weaver). (C) The workflow as displayed in the Galaxy workflow manager.

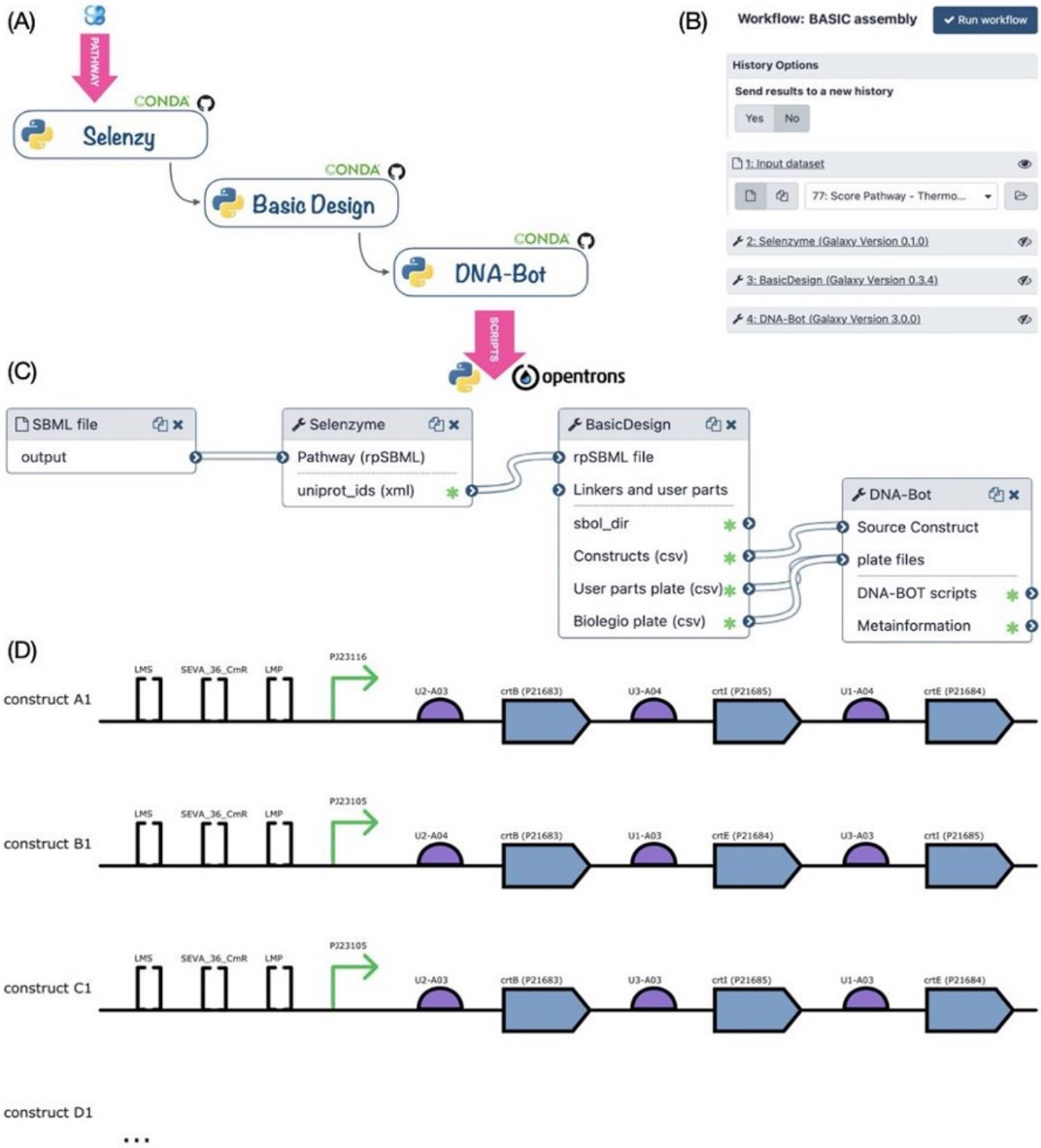


Figure 4: BASIC assembly workflow. (A) The three genetic design tools can be chained manually by running each tool one after the other in a command-line terminal. Outputs of each tool can be directly given as inputs of the others without any other processing. (B) The workflow menu upon executing it through the Galaxy interface. The user specifies the pathway (SBML) that he wishes to build. (C) The workflow as displayed in the Galaxy workflow manager. (D) Architecture of the three first constructs generated by BasicDesign for the lycopene pathway in SBOL format. The view representation is generated using the VisBOL web service. Squared brackets represent "miscellaneous" parts corresponding to methylated prefix and suffix linkers (LMS and LMP) and the plasmid backbone (BASIC_S...). Other parts (promoter, RBS, CDS) are shown using the usual SBOL symbols, the RBS sequences are coded on standardized UTR-RBS linkers and so form the linkers between the promoter and CDS parts.

Implementing Cross-Platform Protocol Execution with the Protocol Activity Modeling Language

Bryan Bartley[1,*], Jacob Beal[1], Alexis Casas[2], Jeremy Cahill[3], Timothy Fallon[4], Daniel Bryce[5], Robert P. Goldman[5], Luiza Hesketh[6], Tim Dobbs[7], Alejandro Vignoni[8]

[1]Raytheon BBN (Cambridge, MA, USA), [2]Imperial College London, [3]Metamer Labs (Boston, MA, USA), [4]Scripps Institution of Oceanography / University of California San Diego, [5]SIFT, LLC (Minneapolis, MN, USA), [6]Campinas State University, [7]Learning Planet Institute (Paris, FR) [8]Universitat Politécnica de València

bryan.a.bartley@raytheon.com

1 INTRODUCTION

Laboratory protocols are used for a wide range of purposes in research and development, at many different stages, including experiment design, execution, data analysis, interpretation, and communication and sharing with other groups (Fig 1). However, protocols are often difficult to communicate or reproduce, given the differences in context, skills, instruments, and other resources between different projects, investigators, and organizations. To this end, the Bioprotocols Working Group (https://github.com/Bioprotocols has developed a draft specification [1] for a unified protocol modeling language, called the Protocol Activity Modeling Language (PAML). The PAML data model has been designed to support the following needs:

- **Execution by either humans or machines**
- **Maintaining execution records and associated metadata markup**
- **Mapping protocols from one laboratory environment to another**
- **Recording modifications of protocols and the relationship between different versions**
- **Verification and validation of protocol completeness and coherence**
- **Planning, scheduling, and allocation of laboratory resources**

Here we describe recent progress implementing PAML and demonstrating that it can be translated to and executed across different laboratory platforms in order to address use cases presented by the stakeholder community.

2 AUTHORING AND EXECUTING PAML PROTOCOLS

PAML protocols may be authored using either the pypaml programming API or the web-based PAML Editor (https://pamled.sift.net). The pypaml library implements the PAML standard in Python and provides modeling, execution, and exporting functionality (https://github.com/Bioprotocols/paml/). The PAML Editor is a React and Django application built upon pypaml. It provides a browser-based visual scripting interface, using activity blocks whose input/output ports are connected by flows. The PAML Editor is intended for a broad user-base and does not require significant programming expertise.

We have also implemented an execution engine as part of the pypaml library that can directly execute a protocol through an instrument's API or via translation into an executable format, such as Autoprotocol. The PAML execution engine uses a token based execution semantics that implements the UML activity model based upon Petri-nets [2]. Execution involves identifying when each activity should be executed and processing the flow of input and output tokens between activities.

3 IMPLEMENTING SPECIALIZED PROTOCOL EXECUTION FOR MULTIPLE PLATFORMS

The pypaml execution engine uses specialized listeners to translate protocols into different protocol formats. Current listener prototypes focus on laboratory automation (Autoprotocol) and human-readable "paper protocols" (Markdown). For example, the Autoprotocol listener translates PAML protocols into a list of instructions in JSON format operating on reagents and containers. The Markdown listener renders written protocols as Markdown documents with hyperlinked definitions of reagents and containers. Currently, members of our working group are also implementing specialized listeners for Opentrons and Echo lab robots.

Much recent PAML development has been driven by the International Genetically Engineered Machines (iGEM) community. This year, as part of the iGEM interlaboratory study, teams will exchange DNA constructs and run measurement assays to assess the reproducibility of the protocol. For this purpose, we have encoded several PAML protocols for calibrated fluorescence measurement in 96-well microplate cultures. This year students will be executing "paper protocols" generated as Markdown from PAML source. This "crowdsourcing" will serve as practical validation of PAML's ability to capture and describe relevant details for experimental execution and reproducibility.

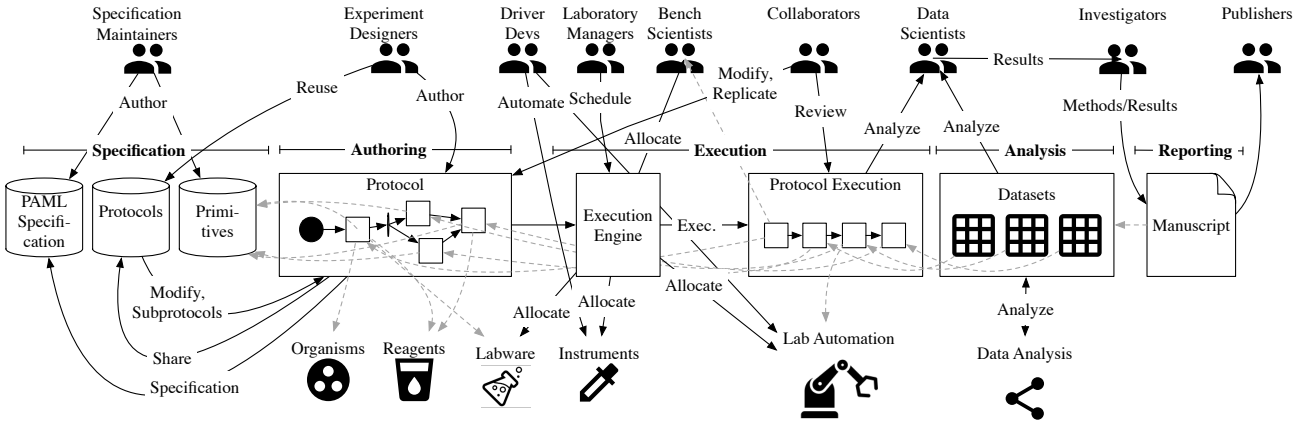


Figure 1: The PAML common protocol language was developed to support needs of diverse users, including experiment designers, bench scientists, driver developers, laboratory managers, data scientists, investigators, and publishers (top). The data model represents the flow of activities and participating elements, including organisms, reagents, labware, scientific instruments, laboratory automation, and datasets (bottom). PAML enables users to communicate about the operations involving these elements and partially or fully automate their design, execution, analysis, and interpretation (middle).

Looking forward to iGEM in 2023, our goal is to validate cross-platform execution of PAML by running these same protocols on Opentrons machines. Toward this end, the "Friendzymes" iGEM team, which represents an international, distributed, open science collaboration developed an Opentrons execution engine. Protocols for PCR and calibrated fluorescence measurements have since been encoded and successfully executed on Opentrons robots in the lab.

Another iGEM initiative related to protocol-sharing has grown out of a prior collaboration between the Imperial College London, Paris-Bettencourt, and Costa Rica teams. These teams have developed an automated Golden Gate assembly protocol with Beckman Coulter Echo™ 525 / 550 acoustic liquid handlers. These teams are currently encoding their protocol in PAML and implementing a new cherry-picking listener to the execution engine.

Eventually, these common molecular biology protocols will be disseminated as PAML on the iGEM Technology Resources web page for use among future iGEM teams and future members of the remote automation collaborative network.

Our working group has also been collaborating with the consortium for Standardization in Lab Automation (SiLA). To demonstrate PAML capabilities for industrial control, we have developed a pH calibration protocol. This protocol illustrates the capability to adapt a base protocol with variable inputs (such as the volume of solution to produce) to multiple scales. It relies upon computational and physical primitives that calculate quantities, select appropriate labware and instruments, mix materials, and alter execution based upon instrument telemetry.

These ongoing efforts to develop executable PAML protocols for a variety of platforms provide an initial demonstration that this emerging standard can address many diverse use cases and challenges faced by existing interlaboratory collaborations.

4 FUTURE DIRECTIONS

We hope these initial demonstrations will convince others in the broader community of the value and utility of PAML and encourage others to contribute. To this end, our Bioprotocols Working Group is open to any organizations and individuals with an interest in standardized representation of biological protocols. We are currently drafting a formal organization, governance, and fundraising strategy for this community.

Acknowledgements

We would like to thank the Interlaboratory Working Group of the iGEM Engineering Committee for their help refining PAML protocols for use by the iGEM community. This work was supported by Air Force Research Laboratory (AFRL) and DARPA contract FA8750-17-C-0184. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

REFERENCES

[1] Bartley, Bryan et al. Building an open representation for biological protocols. *bioRxiv* (2022).

[2] Petri, C. A. *Communication with automata.* PhD thesis, Universitat Hamburg, 1966.

Lightning Talks

The following 20 abstracts feature as 90-second lightning talks and posters in this year's IWBDA program:

(1) *Steps Towards Functional Synthetic Biology.* Ibrahim Aldulijan, Jacob Beal, Sonja Billerbeck, Jeff Bouffard, Gaël Chambonnier, Nikolaos Delkis, Isaac Guerreiro, Martin Holub Martin Holub, Daisuke Kiga, Jacky Loo, Paul Ross, Vinoo Selvarajah, Noah Sprent, Gonzalo Vidal and Alejandro Vignoni

(2) *Adapting Malware Detection to DNA Screening.* Dan Wyschogrod, Jeff Manthey, Tom Mitchell, Steven Murphy, Adam Clore and Jacob Beal

(3) *Artificial Metabolic Networks: enabling neural computations with metabolic networks.* Léon Faure and Jean-Loup Faulon

(4) *Developing a scoring system to optimise the design of CRISPR Cas12 diagnostics.* Akashaditya Das and Ana Pascual-Garrigos

(5) *DBTL bioengineering cycle: developing a population oscillator.* Andrés Arboleda-García, Iván Alarcon-Ruiz, Yadira Boada, Jesús Picó and Eloisa Jantus-Lewintre

(6) *Computer-aided enhancement of genetic design data.* Matthew Crowther and Angel Goñi-Moreno

(7) *Exploring Advantages and Limitations of Discrete Modeling of Biological Network Motifs.* Difei Tang, Gaoxiang Zhou and Natasa Miskov-Zivanov

(8) *SynPath – An Automated Biosynthetic Pathway Design and Analysis Tool.* Carol Gao, Helena van Tol and Xi Wang

(9) *The Context Matrix: A Framework for Context-Aware Synthetic Biology.* Camillo Moschner, Charlie Wedd and Somenath Bakshi

(10) *An Interactive Microfluidic Design and Control Workflow.* Yangruirui Zhou and Douglas Densmore

(11) *Dynamic Behavior Alters Influences and Sensitivities in Biological Networks.* Gaoxiang Zhou and Natasa Miskov-Zivanov

(12) *Experimental Data Converter.* Sai Samineni, Gonzalo Vidal, Jeanet Mante, Guillermo Yañez-Feliú, Carlos Vidal-Céspedes, Chris Myers and Timothy J. Rudge

(13) *Expanding the metaheuristic framework: evolving cells with the bat algorithm.* Víctor Reyes, Nicolás Hidalgo and Martín Gutiérrez

(14) *PLATERO: A Plate Reader Calibration Protocol to work with different instrument gains and asses measurement uncertainty.* Yadira Boada, Alba González-Cebrián, Joan Borràs-Ferrís, Jesús Picó, Alberto Ferrer and Alejandro Vignoni

(15) *Spatially Solving the Graph Coloring Problem Using Intercell Communication.* Daniela Moreno, Diego Araya and Martín Gutiérrez

(16) *A comparison between D-optimal and model-based design of experiments for efficient biomanufacturing.* Iván Blázquez Arenas, Pablo Carbonell and Irene Otero-Muras

(17) *Probabilistic programming for synthetic gene networks.* Lewis Grozinger and Angel Goñi-Moreno

(18) *In-silico design for fold-change detection (FCD) synthetic circuits.* Rongying Huang and Ramez Daniel

(19) *Rule-based generation of synthetic genetic circuits.* Daisuke Kiga, Kazuteru Miyazaki, Shoya Yasuda, Ritsuki Hamada, Sota Okuda, Ryoji Sekine, Naoki Kodama and Masayuki Yamamura

(20) *Standardizing the Representation of Parts and Devices for Build Planning.* Jacob Beal, Vinoo Selvarajah, Gael Chambonnier, Traci Haddock-Angelli, Alejandro Vignoni, Gonzalo Vidal and Nicholas Roehner

Artificial Metabolic Networks: enabling neural computations with metabolic networks

Léon FAURE
Université Paris-Saclay
INRAe
Jouy-en-Josas, France
leon.faure@inrae.fr

Bastien MOLLET
AgroParisTech
Saclay, France

Wolfram LIEBERMEISTER
INRAe
Jouy-en-Josas, France

Jean-Loup FAULON
INRAe
Jouy-en-Josas, France
jean-loup.faulon@inrae.fr

Metabolic networks have largely been exploited as mechanistic tools to predict the behavior of a strain in different environments. However, the performance of this constraint-based modeling approach relies on labor-intensive experiments to determine media intake fluxes. In this paper, we show how neural methods can surrogate constraint-based modeling, make a metabolic network suitable for backpropagation, and consequently be used as an architecture for machine learning. We showcase the performance of our hybrid - mechanistic and neural - model, fitted with an experimental dataset of Escherichia coli growth rates in different media compositions, reaching a regression coefficient of 0.76 on cross-validation aggregated test sets. We expect Artificial Metabolic Networks to provide easier discovery of metabolic insights and prompt new biotechnological applications.

1 ABSTRACT

The increasing amounts of data available for biological research bring the challenge of data integration with machine learning (ML) methods, and Synthetic Biology and Metabolic Engineering make no exception to this trend [1–4]. Metabolic Engineering relies on models that predict the phenotype of a strain from its genotype and environment. In the past three decades, constraint-based Metabolic Flux Analysis (MFA) has been the main approach to study the relationship between the uptake of nutrients and the metabolic phenotype (i.e., the steady-state fluxes distribution) of a given organism [5]. In some cases, data integration of several -omics methods is possible, constituting a state-of-the-art multi-omics data integration for MFA [6–8]. Naturally, ML approaches were developed in order to efficiently integrate the data and enhance the predictive power of constraint-based models. However, as described by Sahu et al., [9] the interplay between MFA and ML is still showing a gap: some approaches use ML as input for MFA, others use MFA as input for ML, but none of them can do both.

An emerging field, Scientific Machine Learning (SciML), aims to develop hybrid models that bridge the gap between ML and Mechanistic Modelling (MM) [10]. The main advantage of hybrid modeling is to offer models that comply well with experimental results via ML, but it also takes mechanistic insights from MM (i.e. the stiff computations from defined equations). Several recent pieces of work developed this kind of hybrid model for biological data [11–13].

The hybrid model shown here fits in the emerging SciML field. Artificial Metabolic Networks (AMNs) bridge the gap between ML and MFA by solving linear programming (LP) problems for metabolic flux models with a recurrent neural network (RNN) that has the same topology as the metabolic network itself. By doing so, our model is a mechanistic model, determined by the stoichiometry and other constraints of classical MFA, but also an ML model, as it can be used as a learning platform, with any MFA suitable data.

The use of RNNs for solving optimization problems is a long-standing field of research [14] inspired by the pioneering work of Hopfield and Tank [15]. In the present work, we use one of the most recent and advanced pieces of work to solve LP with RNNs [16]. We first show the basic design and functioning of AMNs and their ability to surrogate MFA, by using simulation data generated

with Flux Balance Analysis (FBA) on CobraPy [17] as a reference, with different models of different sizes, with different inputs (i.e., sets of upper bounds on uptake fluxes). Figure 1 demonstrates that in all tested cases AMNs produce fluxes close to those computed by FBA.

We next show that surrogating FBA with neural architectures can be exploited for gaining predictive power on experimental data. Indeed, after demonstrating that AMNs were giving the same results as FBA for simulated medium, we next used AMNs for characterizing and predicting the growth of E. coli on many different real media compositions. To that end, we grew E. coli DH5-alpha in 70 different media compositions, with M9 as a basis and 10 different carbon sources that can be added, between 1 and 4 simultaneously. Since AMNs rely on neural methods, they are compatible with gradient descent methods. By back-propagating (through the metabolic network) the errors between experimental measures and the model's predicted values, we can fit a set of experimental data to a single model. To show the utility of such gradient descent on AMNs, the model learned the relationship between each medium element concentration and the AMN's input (upper bounds on uptake fluxes) with the objective of minimizing the mean squared error between AMN-predicted biomass and the actual measured growth rate. A classical ANN is the model used to pass medium concentrations to uptake fluxes. Figure 2 shows that good predictive power is obtained for the growth rate, with a Q^2 of 0.78 on cross-validation sets, these performances far outperform those obtained by the CobraPy FBA solver (Fig. 2C).

In classical MFA, plausible flux distributions can be found by measuring fluxes and imposing constraints on them. With AMNs, we avoided costly experimental flux determinations and found plausible flux distributions by learning the consensual metabolic behavior of an organism in response to a set of environments. We also learned the relationship between medium nutrient concentrations and the actual nutrient uptake by the bacteria, eventually revealing complex regulations from E. coli's environment to its ideal metabolic phenotype. Added to the possible unveiling of biological mechanisms linking an organism's environment to its metabolic phenotype, AMNs can also be exploited for industrial applications. Indeed, since one can design any objective function to optimize with AMNs, they can be used to search optimum media for the bioproduction of compounds of interest as well as new decision-making devices for the multiplexed detection of metabolic biomarkers or environmental pollutants.

More details about this abstract can be found on bioRxiv.

REFERENCES

[1] D. M. Camacho et al., Cell 173, 1581 (2018). 10.1016/j.cell.2018.05.015

[2] P. Carbonell, T. Radivojevic, and H. García Martín, ACS Synth. Biol. 8, 1474 (2019). 10.1021/acssynbio.8b00540

[3] J.-L. Faulon and L. Faure, Curr. Opin. Chem. Biol. 65, 85 (2021). 10.1016/j.cbpa.2021.06.002

[4] S. G. Wu et al., ChemBioEng Rev. 3, 45 (2016). 10.1371/journal.pcbi.1004838

[5] J. L. Reed and B. Ø. Palsson, J. Bacteriol. 185, 2692 (2003). 10.1128/JB.185.9.2692-2699.2003

[6] M. Kim et al., Nat. Commun. 7, (2016). 10.1038/ncomms13090

[7] J. E. Lewis and M. L. Kemp, Nat. Commun. 12, 2700 (2021). 10.1038/s41467-021-22989-1

[8] G. Zampieri et al., PLoS Comput. Biol. 15, (2019). 10.1371/journal.pcbi.1007084

[9] A. Sahu et al., Comput. Struct. Biotechnol. J. 19, 4626 (2021). 10.1016/j.csbj.2021.08.004

[10] N. Baker et al., USDOE Office of Science (SC) (2019). 10.2172/1478744

[11] A. Nilsson et al., bioRxiv, (2021). 10.1101/2021.09.24.461703

[12] R. Anantharaman et al., bioRxiv, (2021). 10.1101/2021.10.10.463808

[13] J. H. Lagergren et al., PLOS Comput. Biol. 16, (2020). 10.1371/journal.pcbi.1008462

[14] L. Jin et al., Appl. Soft Comput. 76, 533 (2019). 10.1007/BF00339943

[15] J. J. Hopfield and D. W. Tank, Biol. Cybern. 52, 141 (1985).

[16] Y. Yang et al., Math. Comput. Simul. 101, 103 (2014). 10.1016/j.matcom.2014.02.006

[17] Ebrahim, A. et al., BMC Syst Biol 7, 74 (2013). 10.1186/1752-0509-7-74

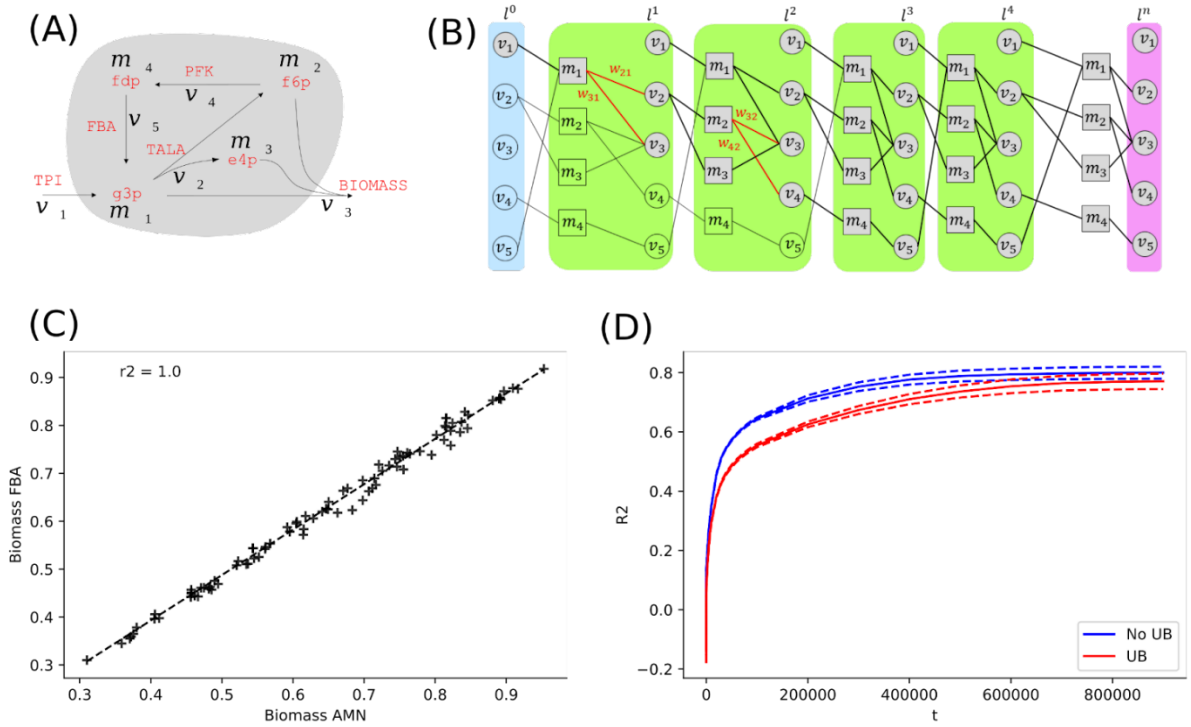


Figure 1. AMNs accurately surrogate FBA by iteratively propagating fluxes through the metabolic network. (A) A toy network is composed of 4 metabolites and 5 fluxes. (B) Schematic iterative unrolling of an AMN at the start (l0) only v1 (TPI) intake fluxes is populated (C) AMNs predicted biomass vs. FBA calculated biomass on the same inputs. (D) Validation set regression coefficient R^2 improves logarithmically with the number of iterations (t), with upper bounds (UB) or exact values on uptake fluxes as inputs (No UB).

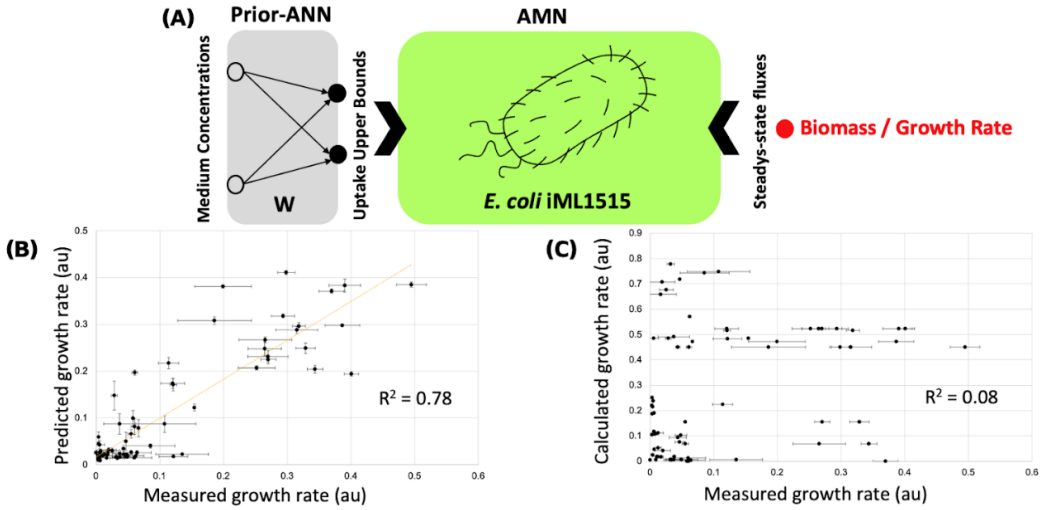


Figure 2. Experimental growth rate fitted with AMNs. (A) An AMN is coupled with a prior-ANN (with weights W) for predicting the steady-state fluxes distributions (including the growth rate) from medium elements concentrations. (B) Regression coefficient on cross-validation test sets for the architecture depicted in (A) trained on 73 experimental nutrient concentrations and corresponding growth rates. (C) Regression coefficient obtained with a FBA approach for same experimental nutrient concentrations (nutrients uptake fluxes' upper bounds are set to an arbitrary large value.

Computer-aided enhancement of genetic design data

Matthew Crowther
School of Computing, Newcastle University, Newcastle Upon Tyne, United Kingdom
m.crowther1@ncl.ac.uk

Ángel Goñi-Moreno
Centro de Biotecnología y Genómica de Plantas, UPM-INIA/CSIC Madrid, Spain
angel.goni@upm.es

1 INTRODUCTION

Data formats for representing DNA sequences differ in the complexity of the information they can encode. While FASTA and GenBank are ubiquitous within fields ranging from genetic engineering to molecular biology, these can only represent limited information and the static nature is counter productive to the iterative nature of synthetic biology. For instance, these formats cannot capture non-DNA elements, connections (e.g., repression between a regulator and a promoter sequence) and hierarchical information. However, the Synthetic Biology Open Language[5] (SBOL) is a data format specifically developed to represent all of the types above (and more), which are highly relevant to synthetic biology efforts. SBOL is not commonly used, partly due to the lack of methods to bridge existing representations (e.g., GenBank) to more information-rich formats (e.g, SBOL). This issue underpins our work, where we present a user-friendly method to bridge that gap.

Several methods and tools exist for the specification of new designs, such as SBOL Visual[1], ShortBOL[2] and many programming language packages[6]. However, processes to retroactively enhance existing data have been explored much less. Currently, the most common approach for bringing existing non-standard designs into a standardised workflow is converting the data into a basic standard representation, for example, translating Genbank into SBOL. However, these processes require considerable pre/post-processing because certain data elements will be partially encoded or missing as they cannot be specified within older formats. For example, a hierarchical design structure fundamental to synthetic biology cannot be encoded within Genbank because of the inherent flat layout. A recent tool, SYNBICT [8], aims to enrich genetic designs by matching and swapping elements with existing and known templates. Our approach infers data by expanding user-provided input and uses network methods to automatically modify designs accordingly.

Networks are at the core of our approach because, unlike the free-text-based formats, networks are dynamic structures, resulting in much higher levels of interrogation, manipulation and integration of multivariate data. In order to build networks out of circuit designs, data is represented in the form of nodes (individual points of data) and edges (relationships between the data) [4]. For example, when building networks from circuit designs, a repression relationship edge links two nodes representing a regulator protein (e.g. aTc) and its cognate promoter (pTet)[3].

When design data is represented, the network is multivariate, i.e. multiple types of information are captured in a single network. Therefore, the data must be structured according to a known model because the relevance of connections between data cannot be quantified computationally. Knowledge graphs are networks that model a real-world environment. Practically where metadata comprises semantic labels and rules are established around how nodes can connect. The ability to capture noisy real-world information in a structured domain allows more abstract and unanticipated questions to be asked, resulting in the ability to perform a more comprehensive analysis.

Here, we explore the specification of genetic designs via a network-centric approach. The focus is on integrating additions into the original dataset, including the method to infer new data automatically when using abstract projections.

2 RESULTS

Network Representation

Before any inference or integration of new data can be applied to the design, the data is represented as a network. No inference of information that is not explicitly defined in the original data is made; instead, the approach simply restates the current data into a knowledge graph. To display the processes involved, we will take an existing GenBank encoded design (0x87 [7]) as an example to illustrate graph-centric data integration. Figure1A translates GenBank annotations into a set of unconnected nodes, including metadata. Currently, the resultant graph contains the same information as the Genbank but can be edited more efficiently.

Integrating new data from multiple projections

Networks are dynamic and can be manipulated to present particular aspects of a dataset, allowing the individual editing of specific aspects of the data. Two examples follow.

Adding structural information. Figure 1A is the result of translating the design from Genbank into a network and does not contain information on how entities are physically or conceptually structured. Therefore, the first additions will specify the hierarchical tree of conceptual entities (devices and circuits) and physical entities (biological parts). Figure

1B displays the output of adding edges denoting possession; for example, Betl-NOR contains the part Ptet. The addition of structure will be automatically applied to the underlying SBOL structure, and therefore the enhancements are integrated without requiring manual input.

Adding functional information. Another type of data that Genbank designs cannot capture is functional data, i.e. how entities interact. Figure 1C is the output where protein, input regulation and non-genetic entity nodes are added and combined with interaction data such as regulatory and transcriptional processes. Like the structural additions, the data is automatically integrated into the underlying SBOL structure.

Inferring data. The key advantage of developing genetic designs when represented as a knowledge graph is that network analysis can be performed to infer data. Knowledge graphs follow structural rules on what data (nodes) can connect and what data must be present. However, not all the information is relevant to all representations, but the information must be added to ensure consistency of the underlying model. Therefore, required elements of the design can be added automatically despite this information not explicitly being defined. Graph traversal is a fundamental process in network science and finds the path between two or more nodes or assesses the availability and quality of paths. Here graph traversal is used to find the most likely object an addition is targetting. If a protein interaction graph is projected, the interactions between proteins are not direct and are implicit, i.e. a protein affects previous regulatory mechanisms. Therefore, pathfinding is required to find the most likely element the subject protein affects. Figure 2 illustrates a simple example of this. The protein interaction graph is projected.

(1) A repression edge between BM3R1 and SprR is stated.
(2) Project interaction graph in reverse.
(3) Perform a breadth-first search from SprR to find closest Activators (pBM3R1).
(4) Create a repression relationship between BM3R1 and pBM3R1.
(5) Project interaction graph in reverse.

During this exercise, 68 manual additions were provided, but the overall dataset grew from 39 nodes and 0 edges to 514 nodes and 1314 edges via inferences made.Therefore the final amount of automatically inferred data is considerably higher than the small pool of manual inputs.

3 METHODS

A Genbank encoded design file was used as input for this example. The GenBank is then automatically translated into a graph representation where a user would then pick a graph representation based on the what aspect of the data is being developed. A user can manually add new edges based on the projection, which will then be automatically integrated into the underlying data ensuring the structure is maintained, and inferences are applied where relevant. Once all additions are made, the graph can be automatically exported into the SBOL format.

4 DISCUSSION & FUTURE WORK

Here, we present a process to specify genetic designs via a network approach focused on inferring data. The ability to present the design in several ways depending on what aspect is being developed provides a specification method that abstracts unnecessary complexity. These specifications are an initial effort to explore a more automated approach to design enhancement. A network and knowledge graph approach can be leveraged to enhance data quality and biological completeness.

Future

Referential standardisation refers to each biological entity having a virtual analogue. These external analogues must be manually embedded within designs, which is seldom done. Network analysis can identify these virtual analogues by comparing graph elements within the design to virtual elements. Knowledge graphs do not inherently ensure information is biologically accurate i,e, a design may be structurally correct but semantically false. However, design agnostic knowledge graphs can be used to capture and evaluate correct information, which can then be used to enhance designs.

REFERENCES

[1] Beal, J., Nguyen, T., Gorochowski, T. E., Goni-Moreno, A., Scott-Brown, J., McLaughlin, J. A., Madsen, C., Aleritsch, B., Bartley, B., Bhakta, S., et al. Communicating structure and function in synthetic biology diagrams. *ACS synthetic biology 8*, 8 (2019), 1818–1825.

[2] Crowther, M., Grozinger, L., Pocock, M., Taylor, C. P., McLaughlin, J. A., Misirli, G., Bartley, B. A., Beal, J., Goñi-Moreno, A., and Wipat, A. Shortbol: a language for scripting designs for engineered biological systems using synthetic biology open language (sbol). *ACS synthetic biology 9*, 4 (2020), 962–966.

[3] Crowther, M., Wipat, A., and Goñi-Moreno, Á. A network approach to genetic circuit designs. *ACS Synthetic Biology* (2022).

[4] Krempel, L. Network visualization. *The SAGE handbook of social network analysis* (2011), 558–577.

[5] Madsen, C., Moreno, A. G., Umesh, P., Palchick, Z., Roehner, N., Atallah, C., Bartley, B., Choi, K., Cox, R. S., Gorochowski, T., et al. Synthetic biology open language (sbol) version 2.3. *Journal of integrative bioinformatics 16*, 2 (2019).

[6] Mitchell, T., Beal, J., and Bartley, B. pysbol3: Sbol3 for python programmers. *ACS Synthetic Biology* (2022).

[7] Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., and Voigt, C. A. Genetic circuit design automation. *Science 352*, 6281 (2016), aac7341.

[8] Roehner, N., Mante, J., Myers, C. J., and Beal, J. Synthetic biology curation tools (synbict). *ACS Synthetic Biology 10*, 11 (2021), 3200–3204.

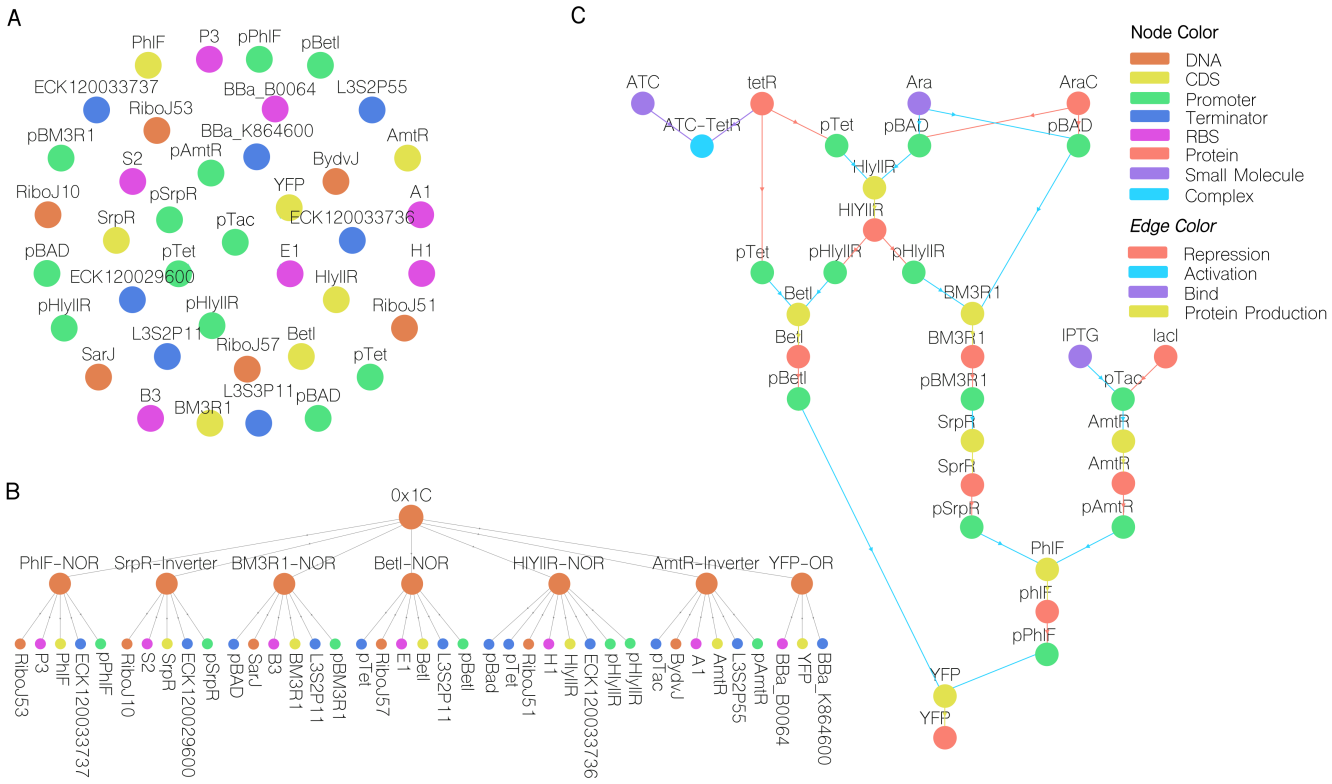


Figure 1: A) Genetic circuit with name 0x1C from Nielsen et al. directly translated from a GenBank file into a network structure. No connecting information is derived; flat layout by default. B) Adding hierarchical structure by creating groups and connecting the related physical parts into modules. C) Specifying functional information by adding Protein, non-genetic elements and mechanisms to regulate input and interactions between physical entities, i.e., transcription and regulation.

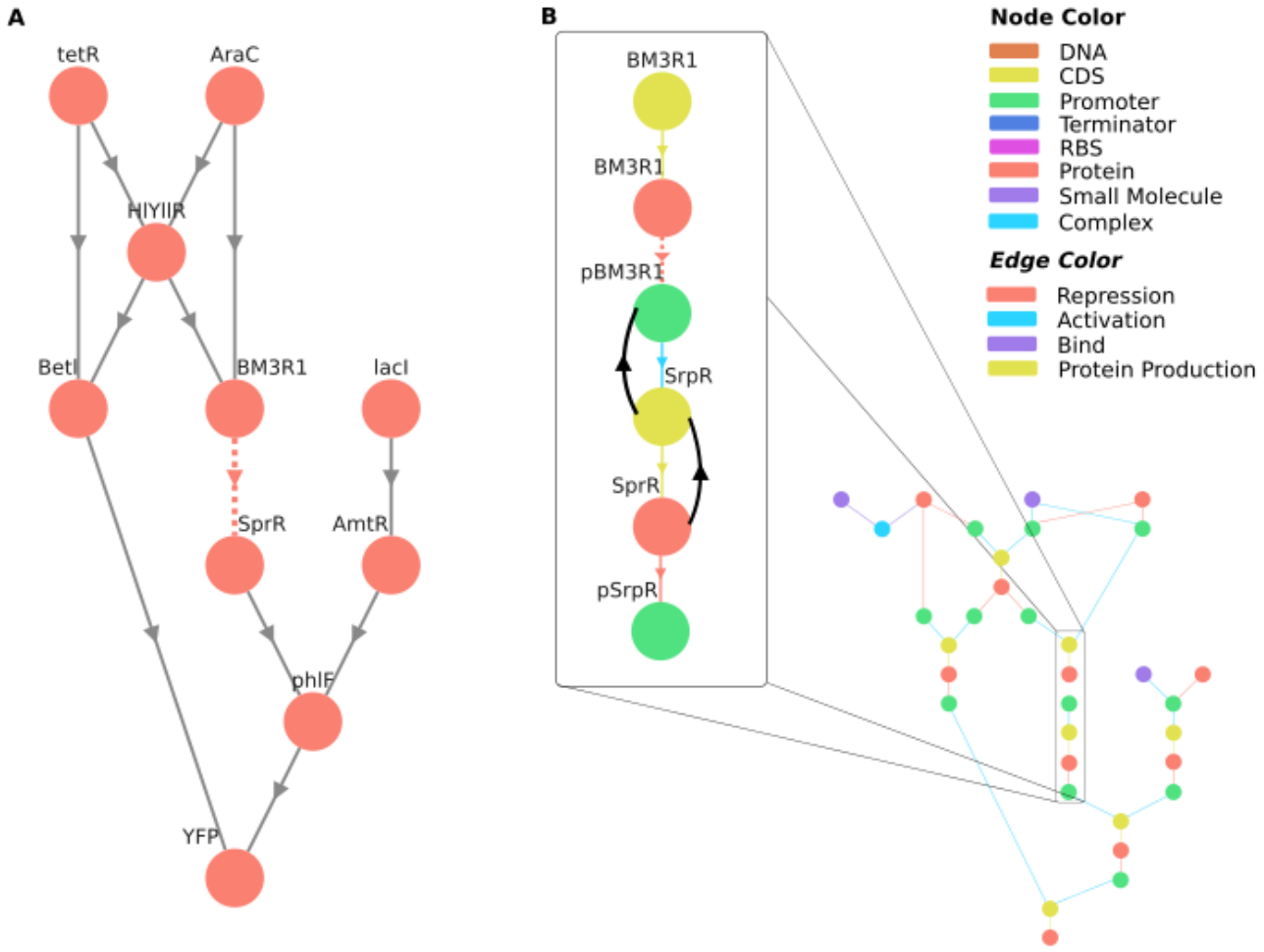


Figure 2: A) Starting with a protein interaction map, the user adds a connection: BM3R1 represses SprR. B) The previous modification (adding a repression interaction) modifies the underlying data structure automatically. Network analysis (pathfinding) is performed on a full interaction graph to find the activators of SprR. The following elements are targets for BM3R1. All this information is modified and saved into the new design.

SynPath – An Automated Biosynthetic Pathway Design and Analysis Tool

Yuzhi Gao
University of Pennsylvania
Philadelphia, USA
carolgyz@seas.upenn.edu

Helena van Tol
Amyris, Inc
Emeryville, USA
vantol@amyris.com

Xi Wang
Lawerence Berkeley National Laboratory
Berkeley, USA
xiwang@lbl.gov

1 INTRODUCTION

Biosynthesis is a multi-step, enzyme-catalyzed process where substrates are converted into more complex products in living organisms. Biosynthetic pathways refer to the series of biochemical reactions that connect two different metabolites. Introducing heterologous pathways into an organism allows it to produce non-native metabolites. Thus, the design and construction of biosynthetic pathways have gained attention given their potential to produce valuable chemicals at scale. However, due to the huge combinatorial possibilities of enzymes and reactions that convert one chemical into another, selecting the optimal combination of enzymes becomes an arduous task. Computational tools could aid in the experimental process at the design stage of engineering to screen for the best-performing pathways and downsize the pathway candidates pool for wet-lab experiments.

The enumeration of possible biosynthetic pathways has been enabled by the expansive reaction and enzyme databases of experimentally elucidated biochemical reactions, such as Metacyc [1]. Using such databases, we can iteratively apply reversed chemical transformation starting from a target product to reach precursors native to the host and to discover all possible combinations of reactions leading to the target product – a concept called retrosynthesis.

Genome-scale models (GEM) are mathematical representations of an organisms' gene-reaction-metabolites networks using matrices. Constraint-based modeling (CBM) enabled us the quantitatively analysis of the reconstructed metabolic networks with physiological relevance [2]. By introducing a novel pathway into existing GEMs, we can evaluate its impact on the biochemical network, as well as the metabolic flux channeled towards the production of the target metabolite.

Here, we introduce an automated computational design tool integrating retrosynthesis and constraint-based genome-scale modeling to identify promising biosynthetic pathways.

2 CONTENTS OF THE ABSTRACT

The design and construction of de novo biosynthetic pathways require careful consideration of the metabolic context of the host organism. We built an automated design tool to guide the construction and engineering of de novo biosynthetic pathways to improve the production of valuable chemicals through metabolic engineering. Here, we've developed a user-friendly software and interface, SynPath, that integrates the retrosynthesis pathway discovery algorithm with the analysis of these pathways using genome scale modeling and constraint-based flux analysis (Figure 1). The pathways are evaluated from theoretical yields, as well as from the metabolic burden that the heterologous pathways could incur upon the host organism. High metabolic burdens lead to undesirable physiological changes, placing hidden constraints on host growth and productivity [3]. SynPath suggests pathway designs that have been experimentally validated pathways, such as those for 1,4-butanediol and 3-hydroxypropionate. Furthermore, the program evaluates multiple pathways simultaneously, allowing for the ease of comparing nuances between pathways. In the example of 1,3-propanediol synthesis in Escherichia coli, the software identified the difference using NADP+ and NAD+, a subtle difference that was experimentally confirmed[4]. By providing meaningful analysis of co-factor balance, the software provides additional dimensions to guide the design de novo biosynthetic pathways and strain optimization. Finally, our software is compatible with the Synthetic Biology Open Language, a standardized format for the electronic exchange of information of biological designs.

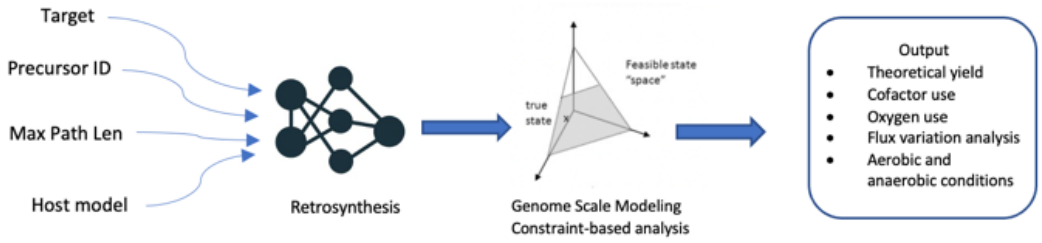


Figure1: An overview of the SynPath workflow.

idx	theoretical_yield	eng_atp	eng_nad	eng_nadp	fva_dif	o2_use	yield_anaerobic	anaerobic_atp_use
5	2.2331627546731747	-43.18951080607732	-20.617115429315184	-10.015116239607057	79908.36750401586	-8.785802101862647	0.42066353201057566	-27.06607090595299
18	2.2331627546731747	-43.18951080607732	-20.617115429314048	-10.015116239607721	79908.36750402101	-8.785802101862759	0.4206635320105839	-27.0660709059522
8	2.2331627546731747	-43.18951080607732	-20.617115429314048	-10.015116239607721	79908.3675040224	-8.785802101862759	0.4206635320105839	-27.0660709059522
22	2.2331627546732067	-43.189510806078424	-20.617115429313788	-10.015116239607808	79908.36750402313	-8.785802101862743	0.4206635320105825	-27.066070905952742
23	2.2331627546732067	-43.189510806078424	-20.61711542931182	-10.015116239608867	79908.36750402325	-8.785802101862883	0.4206635320105654 4	-27.066070905952 38
4	2.2331627546732022	-43.18951080607819	-20.617115429313102	-10.015116239608172	79908.3675040243	-8.785802101862766	0.420663532010578	-27.06607090595264
15	2.2331627546731747	-43.18951080607732	-20.617115429315184	-10.015116239607057	79908.36750402479	-8.785802101862647	0.42066353201057566	-27.06607090595299
21	2.2331627546732093	-43.18951080607857	-20.61711542931674	-10.015116239606169	79908.3675040257	-8.785802101862474	0.4206635320105664 4	-27.066070905952063
20	2.2331627546732093	-43.18951080607857	-20.61711542931333	-10.015116239608037	79908.3675040258	-8.785802101862755	0.4206635320106032	-27.066070905951687
11	2.2331627546732093	-43.18951080607857	-20.61711542931674	-10.015116239606169	79908.3675040258	-8.785802101862474	0.4206635320105664 4	-27.066070905952063
25	2.2331627546732067	-43.189510806078424	-20.61711542931182	-10.015116239608867	79908.3675040277	-8.785802101862883	0.4206635320105654 4	-27.066070905952 38
24	2.2331627546732067	-43.189510806078424	-20.617115429313788	-10.015116239607808	79908.3675040277	-8.785802101862743	0.4206635320105825	-27.066070905952742
19	2.2331627546732067	-43.189510806078424	-20.61711542931182	-10.015116239608867	79908.36750402777	-8.785802101862883	0.4206635320105654 4	-27.066070905952 38
12	2.2331627546732067	-43.189510806078424	-20.617115429313788	-10.015116239607808	79908.36750402777	-8.785802101862743	0.4206635320105825	-27.066070905952742
9	2.2331627546732067	-43.189510806078424	-20.61711542931182	-10.015116239608867	79908.36752632043	-8.785802101862883	0.4206635320105654 4	-27.066070905952 38
2	2.2331627546732067	-43.189510806078424	-20.617115429313788	-10.015116239607808	79908.36752632043	-8.785802101862743	0.4206635320105825	-27.066070905952742
17	2.2331627546732022	-43.18951080607819	-20.617115429312054	-10.015116239608789	79908.3675263225	-8.7858021018629	0.4206635320105507	-27.06607090595259
7	2.2331627546732022	-43.18951080607819	-20.617115429312054	-10.015116239608789	79908.36752632385	-8.7858021018629	0.4206635320105507	-27.06607090595259
10	2.2331627546732093	-43.18951080607857	-20.61711542931333	-10.015116239608037	79908.36752632601	-8.785802101862755	0.4206635320106032	-27.06607090595169
14	2.2331627546732022	-43.18951080607819	-20.617115429313102	-10.015116239608172	79908.36752632624	-8.785802101862766	0.420663532010578	-27.06607090595264
1	2.2331627546732067	-43.189510806078424	-20.61711542931014	-10.015116239609883	79908.36757092796	-8.785802101863124	0.42066353201059115	-27.066070905952675
0	2.2331627546732027	-43.18951080607819	-20.617115429320336	-10.015116239604222	79908.36759323295	-8.785802101862215	0.4206635320106527	-27.06607090595308
13	2.2794081681605545	-41.11970509767855	-19.922839121490085	-11.235771524730005	79930.70426627678	-7.81464841862867	0.4206635320105026	-27.066070905952593
3	2.2794081681605505	-41.11970509767838	-19.92283912149028	-11.235771524729998	79930.7042885738	-7.81464841862838 3	0.42066353201059825	-27.06607090595271
6	2.316538718055551	-39.94801875833535	-29.62635674999023	-3.772530995824788	79959.48894029157	-7.034906870833592	0.44170069027781345	-26.30873320833234
16	2.3165387180555563	-39.94801875833433	-29.62635674999023	-3.772530995824788	79959.48896258925	-7.034906870833592	0.44170069027781345	-26.308733208332338

Download results as sbol

Figure 2: Output page example for the synthesis of farnesene with a maximum of 8 steps. Pathways are listed and indexed, followed by a table summarizing calculation results, in which users can sort according to their desired parameter. There is also an option to download pathway information as SBOL documents.

REFERENCES

[1] Caspi, R., Billington, R. and Keseler, I. et al. The MetaCyc database of metabolic pathways and enzymes - a 2019 update. 2022

[2] Rau, M. and Zeidan, A. Constraint-based modeling in microbial food biotechnology. Biochemical Society Transactions 46, 2 (2018), 249-260.

[3] Wu, G., Yan, Q., Jones, J., Tang, Y., Fong, S. and Koffas, M. Metabolic Burden: Cornerstones in Synthetic Biology and Metabolic Engineering Applications. Trends in Biotechnology 34, 8 (2016), 652-664.

[4] Frazão, C., Trichez, D. and Serrano-Bataille, H. et al. Construction of a synthetic pathway for the production of 1,3-propanediol from glucose. Scientific Reports 9, 1 (2019).

Developing a scoring system to optimise the design of CRISPR Cas12 diagnostics

Akashaditya Das[1], **Ana Pascual Garrigos**[2], **Dr. Jennifer C Molloy**[2], **Dr. James W Ajioka**[1]
[1]Department of Pathology, University of Cambridge [2]Department of Chemical Engineering and Biotechnology, University of Cambridge
das.akashaditya@gmail.com

Introduction

Diagnostic technologies using knowledge of Clustered Regularly Interspaced Palindromic repeats (CRISPR) and CRISPR-associated (Cas) proteins are growing in popularity [2]. In particular, CRISPR Cas12 and CRISPR Cas13 technologies have generated interest because they are able to act as RNA-guided sensors for specific nucleic acid sequences. This has resulted in detection assays for a variety of pathogens [1] [5]. CRISPR technologies have been the subject of extensive design and optimisation research for applications in genome editing [3]. However, limited work has been undertaken to develop design tools for CRISPR Cas12 diagnostics. In this abstract, we showcase our process for identifying optimal CRISPR Cas12 diagnostics through the development of a scoring system for guide RNAs (gRNAs). We use Hepatitis B Virus as a test case and report our initial results using this workflow.

Assay design

We use Cas12's collateral cleavage activity to generate a signal when target DNA sequences are present. Figure 1 shows a pictorial representation of Cas12 collateral cleavage. To activate collateral cleavage, Cas12 forms a complex with a gRNA designed to be complementary to the target DNA. Upon complex formation, this gRNA binds to a complementary DNA sequence and brings Cas12 into close contact with the DNA strand. A structural change occurs in the complex and collateral cleavage is activated due to the exposure of a nuclease domain. The activity of this nuclease domain is known as collateral cleavage activity. It is where the Cas12 cleaves nearby DNA, regardless of their sequence. To generate a signal from collateral cleavage, reporter molecules made from fluorophore and quencher molecules are joined by a short ssDNA linker. When there is no collateral cleavage activity, the proximity of the fluorophore to the quencher means that minimal fluorescence is observed. However, once collateral cleavage is activated, the reporter linker is cut and a fluorescent signal can be measured using the correct excitation/emission spectra. In this way the fluorescent signal observed can be used as a proxy for the presence of a specific DNA sequence.

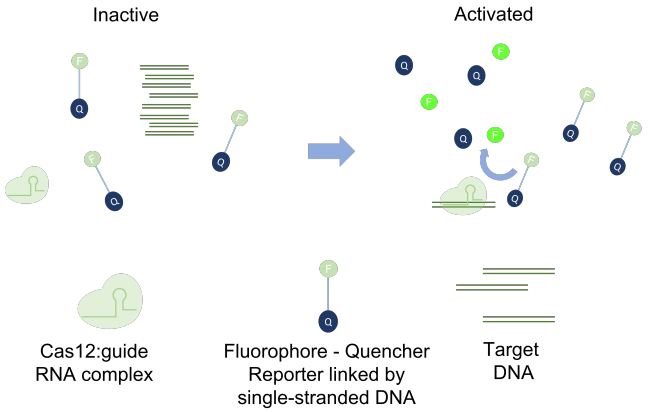


Figure 1: Schematic of Cas 12 collateral cleavage.

Our workflow

To design an optimal diagnostic using CRISPR Cas12, selecting the gRNA used is extremely important. While many papers detail the use of the CRISPR Cas12 system, we found that details of how gRNAs were chosen were often lacking. We propose a workflow (Figure 2) looking at three features that we believe are essential in creating an optimal diagnostic: conservation of the target DNA, selectivity of the target DNA and the collateral cleavage efficacy of a gRNA target DNA pairing. Ideally, we want to combine these three metrics so that proposed gRNAs are assigned a score so they can be ranked against each other in silico.

Our methodology

Conservation scoring. Conservation is the degree to which the target DNA sequence of a gRNA remains the same across variants of the target genome. A target with a high conservation score would be able to detect multiple genomic variants of the same target. To quantify the degree of conservation for a particular sequence, we use BLAST to align the target DNA sequence against variants of the target genome. After performing this alignment we score conservation by the following calculation:

$$Conservation = 1 - \frac{\sum_{i=1}^{N} M_i}{LN_t}$$

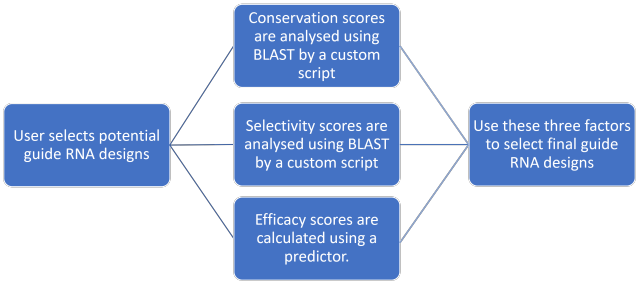


Figure 2: Pictorial representing of the workflow for our scoring system

where M_i is the number of mismatches in alignment i, L is the length of the target DNA and N_t is the total number of target organism genomes that the target DNA is aligned against.

Selectivity Scoring. Selectivity is the degree to which the gRNA targets the target DNA sequence in the intended target. A high selectivity score ensures that any collateral cleavage activity is a true positive result. DNA sequences from non-target genomes in the sample may trigger collateral cleavage activity due to sequence similarity with the target DNA and result in false positive results. We want to minimise this by choosing gRNAs with high selectivity scores. To quantify the degree of selectivity of a particular guide, we align the target DNA for a gRNA against genomes that are likely to be in the sample. We then calculate:

$$Selectivity = \frac{\sum_{i=1}{}^{N} M_i}{LN_{nt}}$$

where M_i is the number of mismatches in alignment i, L is the length of the target DNA and N_nt is the total number of genomes of a non-target organism that was aligned that the target DNA is aligned against.

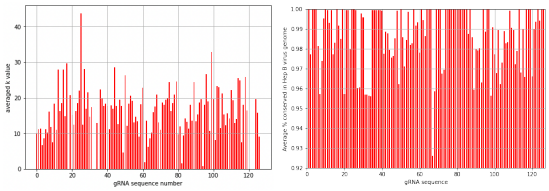
Efficacy scoring. Efficacy tells us how effective a gRNA is in generating collateral cleavage. In the absence of existing theoretic models, we decided to use the fluorescence curves from collateral cleavage assays as an indicator of the efficacy of a given gRNA and therefore required experimental data. From the theoretical model proposed by Metsky et al [4], the cleavage of the reporter follows first-order kinetics in the form of:

$$\frac{d[R]}{dt} = \frac{k_{cat}}{K_M}[E][R]$$

where [R] is the concentration of the not-yet-cleaved reporter, [E] is the concentration of the Cas12 gRNA -target DNA complex, $\frac{k_{cat}}{K_m}$ is the catalytic efficiency of the particular guide-target complex, and t is time. We combine $\frac{k_{cat}}{K_m}$ into one term, k, as a proxy for the efficacy of a gRNA:target DNA pairing. We used automation to perform CRISPR Cas12 collateral cleavage assays at high throughput using gRNAs designed for the World Health Organisation reference genome for Hepatitis B Virus, which generated a data set of 127 gRNAs for the predictor to be trained on.

Preliminary Results: Hepatitis B Virus case study

Here we show the results for conservation and efficacy of Hepatitis B Virus. We are currently working on identifying an optimal selectivity screen for Hepatitis B Virus. As expected, we see a range of efficacy and conservation scores. This highlights the importance of gRNA selection when designing new diagnostics.



(a) Efficacy scoring **(b) Conservation scoring**

Figure 3: Results for gRNA screens for Hepatitis B Virus

Current work

We are currently undertaking a new gRNA screen using the *Salmonella* serotype Typhi genome to validate the method on an independent dataset. This is a good test of the ability of our efficacy predictor to generalise across target genomes.

REFERENCES

[1] Chen, J. S., Ma, E., Harrington, L. B., Da Costa, M., Tian, X., Palefsky, J. M., and Doudna, J. A. CRISPR-Cas12a target binding unleashes indiscriminate single-stranded DNase activity. *Science 360*, 6387 (4 2018), 436–439.

[2] Lander, E. S. The Heroes of CRISPR, 1 2016.

[3] Liu, G., Zhang, Y., and Zhang, T. Computational approaches for effective CRISPR guide RNA design and evaluation. *Computational and Structural Biotechnology Journal 18* (1 2020), 35–44.

[4] Metsky, H. C., Welch, N. L., Pillai, P. P., Haradhvala, N. J., Rumker, L., Mantena, S., Zhang, Y. B., Yang, D. K., Ackerman, C. M., Weller, J., Blainey, P. C., Myhrvold, C., Mitzenmacher, M., and Sabeti, P. C. Designing viral diagnostics with model-based optimization. *bioRxiv* (9 2021), 2020.11.28.401877.

[5] Myhrvold, C., Freije, C. A., Gootenberg, J. S., Abudayyeh, O. O., Metsky, H. C., Durbin, A. F., Kellner, M. J., Tan, A. L., Paul, L. M., Parham, L. A., Garcia, K. F., Barnes, K. G., Chak, B., Mondini, A., Nogueira, M. L., Isern, S., Michael, S. F., Lorenzana, I., Yozwiak, N. L., MacInnis, B. L., Bosch, I., Gehrke, L., Zhang, F., and Sabeti, P. C. Field-deployable viral diagnostics using CRISPR-Cas13. *Science 360*, 6387 (4 2018), 444–448.

Active Learning-based Optimal Experimental Design for efficient biomanufacturing

Iván Blázquez Arenas
Polytechnic University of Madrid
Madrid, Spain
ivan.blazquez.arenas@alumnos.upm.es

Pablo Carbonell
ai2, Univ. Politècnica de València
I2SysBio, UV-CSIC
València, Spain
pablo.carbonell@upv.es

Irene Otero-Muras
Institute for Integrative Systems Biology I2SysBio, UV-CSIC
València, Spain
irene.otero.muras@csic.es

1 INTRODUCTION

The complexity of biotechnological processes has prevented so far the systematic adoption of the biomanufacturing approach to bio-based production for many targets of industrial interest [8, 9]. In order to tackle such complexity, some progress has been achieved through the application of industrial manufacturing optimization techniques such as *D*-optimal design of experiments (ODoE). Such approaches have been incorporated into DBTL pipelines [5] to improve the efficiency of the optimization process. ODoE techniques are agnostic about the mechanism, and therefore convenient to address design problems in contexts with null or very scarce *a priori* mechanistic knowledge. On the contrary, Model-based Optimal Design of Experiments (MBODoE) relies on the mechanistic model of the process, and ensures optimality of the design (for a given model and set of parameters) [1]. MBODoE, in its original formulation [3], is still too costly and unsuited for the optimization of biomanufacturing processes, due to their high levels of uncertainty and complexity. We work on a combined approach for experimental design that exploits the dynamic mechanistic knowledge available on the process with statistical methods, being our ultimate goal finding the global optimal design. As a first step in this direction, we develop here a comparative method between a *D*-optimal DoE strategy [4] and a MBODoE, and illustrate it through case studies of relevance to synthetic biology and biomanufacturing.

2 METHODS

Optimal design of experiments refers to a set of methodologies that aim to find the experimental conditions, admissible in their design space, that maximize the information necessary for the estimation of the parameters of predictive models, while keeping the number of experiments relatively small, which is often the case in biomanufacturing. To find these optimal experiments, the Fisher matrix is widely used in the literature, being a measure of the information contained in an experimental design. Given a response model with the form:

$$y = y(\xi, p) \tag{1}$$

where $\xi \in \mathbb{R}^k$ (k = number of factors considered) represents the support point (experimental conditions) and $p \in \mathbb{R}^{n_p}$ (n_p = number of parameters in the model) the parameters to estimate, the Fisher matrix is defined as:

$$F(\Xi, p) = \frac{1}{\sigma_\epsilon^2} \sum_{i=1}^{n} s_y(\xi_i, p) s_y^T(\xi_i, p) \tag{2}$$

where σ_ϵ^2 is the variance of the error in the estimation, typically assumed as a Gaussian density independent of the experimental conditions; Ξ is the matrix of support points, in which the i-row is the support point ξ_i of each of the n experiments to do; and $s_y(\xi, p)$ is the sensitivity vector:

$$s_y(\xi, p) = \left[\frac{\partial y(\xi,p)}{\partial p_1}, \frac{\partial y(\xi,p)}{\partial p_2}, \cdots, \frac{\partial y(\xi,p)}{\partial p_{n_p}} \right] \tag{3}$$

being n_p the number of parameters to estimate. Among the possible metrics, the *D-Optimality criterion* will be used in this study, given its good results and ease of calculation. According to this criterion, the optimal design of experiments is the result of the optimization problem:

$$\Xi_{opt} = arg\{ \max_{\Xi \in \Xi_{ad}} \{ Det[F(\Xi, p)] \} \} \tag{4}$$

where Ξ_{ad} corresponds to the experimental design space. It can be seen that, without losing any generality, the Fisher matrix depends on the value of the parameters to be estimated. Depending of the structure of the response model selected, the methodology used is either ODoE or MBODoE.

Optimal Design of Experiments (ODoE). In this case, the response model is a linear combination of functions that only depends on the experimental conditions, being the coefficients the parameters of the model to estimate (*LP-structure*). In general, this models can be written:

$$y(\xi, p) = p^T f(\xi) = \left[p_1, p_2, \cdots, p_{n_p} \right] \begin{bmatrix} f_1(\xi) \\ f_2(\xi) \\ \vdots \\ f_{n_p(\xi)} \end{bmatrix} \tag{5}$$

The Fisher matrix that results of the use of this kind of model only depends on the experimental conditions and can be computed analytically, with sensitivity vector:

$$s_y(\xi, p) = s_y(\xi) = \left[f_1(\xi), f_2(\xi), \cdots, f_{n_p}(\xi) \right] \tag{6}$$

Model Based Optimal Design of Experiments (MBODoE). In contrast with the above, this methodology accepts all kind of models, independent of its structure with respect to the parameters, allowing its implementation with complex models that take into account the mechanistic knowledge of the problem. In the case of a *Non-LP structure* model, the Fisher matrix depends on the value of the parameters and maybe can not be possible to obtain an analytical expression of it (as before), being necessary to calculate it numerically.

Implementation. To solve the optimization problem for ODoE, a quadratic model with bifactor interactions has been chosen, and the *Coordinate Exchange* algorithm [6] has been implemented so that the design of experiments matrix is traversed element by element, increasing the value of the *D-optimality criterion* until some termination criteria are met. On the other hand, in the case of MBODoE, a mechanistic approach that approximates the case study by means of a system of ordinary differential equations is used. The *DETMAX* algorithm [7] has been employed to find the local optimal design of experiments, taking several initial values of the parameters in order to achieve (or at least get close) to the global optimum. Codes implemented in *MATLAB* are available at https://github.com/pablocarb/mbodoe and we plan to make also available a *Python* version.

3 RESULTS AND DISCUSSION

In order to perform the comparison between ODoE and MBODoE in practical scenarios, we have considered two synthetic biology and metabolic engineering applications. As a first case study, we considered a protein expression system where the dynamics of activated transcription and translation by an inducer $I(t)$ were modeled by the following ODEs [2]:

$$\dot{m}(t) = K_I \frac{I(t)}{K_d + I(t)} - d_m m(t) \tag{7}$$

$$\dot{P}(t) = K_P m(t) - d_P P(t) \tag{8}$$

where $I(t)$, $m(t)$, and $P(t)$ are the concentrations of the activator, the RNA messenger, and the protein, respectively.

By setting a fixed number of experiments from 5 up to 30, 100 experimental runs were performed for each case by adding ±10% of Gaussian noise to the output signal. As shown in Figure 1, mean relative error of the estimated model in the ODoE decreased with the number of experiments towards the combined effect of noise filtering and the approximation error of the assumed quadratic model. Mean error results for the MBODoE case, in turn, were mainly influenced by the noise signal.

As a second case study, we considered a multi-step metabolic pathway where in addition to gene regulation, enzyme kinetics parameters were introduced into the model. The goal was to obtain an optimal design, starting from a collection of enzyme variants with different Michaelis Constant (K_M) and turnover rates (k_{cat}). The mechanism for the MBODoE is represented by a system of Ordinary Differential Equations based on [4].

In both case studies evaluated, the results show how the use of a mechanistic model, which attempts to approximate the phenomenology present in the experiment, greatly improves the quality of the model obtained and therefore can dramatically reduce the number of required experiments. MBODoE allows us to extract models from which we can draw much more accurate conclusions than what would be obtained with ODoE, for a similar number of experiments to be performed. As a drawback, MBODoE is more computationally expensive (and might result prohibitive for increasing complexities with the current implementation). The parallelization and time efficiency of MBODoE are the subject of ongoing work. As an initial step, here we aimed at a systematic comparative to assess the performance of each method, and our future goal is to advance towards a combined approach for optimal experimental design.

4 ACKNOWLEDGMENTS

PC was supported by the Next Generation EU (NGEU) fund from the Spanish Ministry of Universities (UNI/551/2021) and MCIN/AEI "BioDynamics" (PID2020-117271RB-C21). This research received financial support from Spanish Ministry of Science and Innovation through the Project "CompSynBio" (PID2021-127888NA-I0) and Generalitat Valenciana through "SmartBioFab" (CIAICO/2021/159).

REFERENCES

[1] Akkermans, S., Nimmegeers, P., and Van Impe, J. Comparing design of experiments and optimal experimental design techniques for modelling the micro ial growth rate under static environmental conditions. *Food Microbiology 76* (2018), 504–512.

[2] Alon, U. *An Introduction to Systems Biology: Design Principles of Biological Circuits (2nd ed.).* Chapman and Hall/CRC, 2019.

[3] Balsa-Canto, E., Bandiera, L., and Menolascina, F. Optimal experimental design for systems and synthetic biology using amigo2. *Methods Mol Biol* (2018), 221–239.

[4] Carbonell, P., Faulon, J., and Breitling, R. Efficient learning in metabolic pathway designs through optimal assembling. *IFAC-PapersOnLine 52* (2019), 7–12.

[5] Carbonell, P., J, J. A., and Robinson, C. J. e. a. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Communications Biology 1* (2018), 66.

[6] Goos, P., and Jones, B. *Optimal design of experiments: a case study approach.* Wiley, 2011.

[7] Lin, C. D., Anderson-Cook, C. M., Hamada, M. S., Moore, L. M., and Sitter, R. R. Using genetic algorithms to design experiments: a review. *Quality and Reliability Engineering International 31*, 2 (2015), 155–167.

[8] Otero-Muras, I., and Carbonell, P. Automated engineering of synthetic metabolic pathways for efficient biomanufacturing. *Metabolic Engineering 63* (2021), 61–80.

[9] Tellechea-Luzardo, J., Otero-Muras, I., Goñi-Moreno, A., and Carbonell, P. Fast biofoundries: coping with the challenges of biomanufacturing. *Trends in Biotechnology 40* (2022), 831–842.

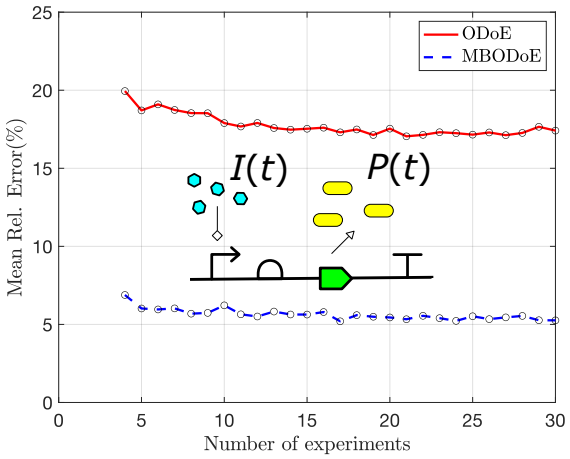


Figure 1: Mean relative error per number of experiments for the ODoE and MBODoE approach for a protein expression system with inducer activation (central inset).

Rule-based generation of synthetic genetic circuits

Daisuke Kiga[1], Kazuteru Miyazaki[2], Shoya Yasuda[3], Ritsuki Hamada[1], Sota Okuda[1], Ryoji Sekine[3], Naoki Kodama[4], Masayuki Yamamura[3]

[1]Waseda University, Tokyo, Japan, [2]National Institution for Academic Degrees and Quality Enhancement of Higher Education, Tokyo, Japan, [3]Tokyo Institute of Technology Kanagawa, Japan, [4]Meiji University Kanagawa, Japan

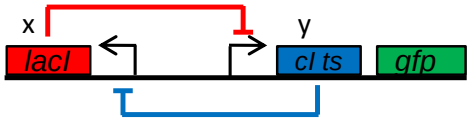
kiga@waseda.jp, teru@niad.ac.jp, my@c.titech.ac.jp

Introduction

As well as expandability of natural biological systems, that of synthetic biological system is derived from huge combinatorial search space of biological components such as protein coding sequences and regulatory sequence. Due to this huge space, in turn, adequate design strategy is required in implementation of synthetic genetic circuit in cells. One strategy is combination of sub-circuit designed in collaborations between biology-background and control engineering-background researchers. Although we had actually achieved designed behavior of cells by combination of synthetic circuits [1, 2], most area of huge combinatorial space was not examined, due to limited search performance either of human being or brute force search by electrical computer.

Results and Discussion

Here, we tried to combine inference machine and deep learning to generate and select candidates of synthetic genetic circuits. From synthetic biologist view point, a logic programming language such as Prolog allows to break down a designed cellular behavior into combinations of biological components when adequate rule base was prepared. Furthermore, when multiple rules share the same head, variations of circuits for the same designed behavior can be generated. Minor variation can be a gene overexpression either by addition of an inducer for a corresponding transcription activator or addition of inhibitor for a corresponding transcription repressor. Variation can be at a higher layer of circuit design (Figure 1). In our previous work, inner state of cells having a toggle switch was set on a separatrix of the potential landscape of the toggle switch by gene over expression of both of the repressors of the toggle switch [2]. We also found that cells can be set on the separatrix by inhibition of expression of the both of the repressors. The strong point of such rule base is that it can be expanded by accumulating cases of synthetic



$$\frac{dx}{dt} = \frac{\alpha_x}{1+(y/K_y)^{n_y}} - \gamma_x x + I_1$$

$$\frac{dy}{dt} = \frac{\alpha_y}{1+(x/K_x)^{n_x}} - \gamma_y y + I_2$$

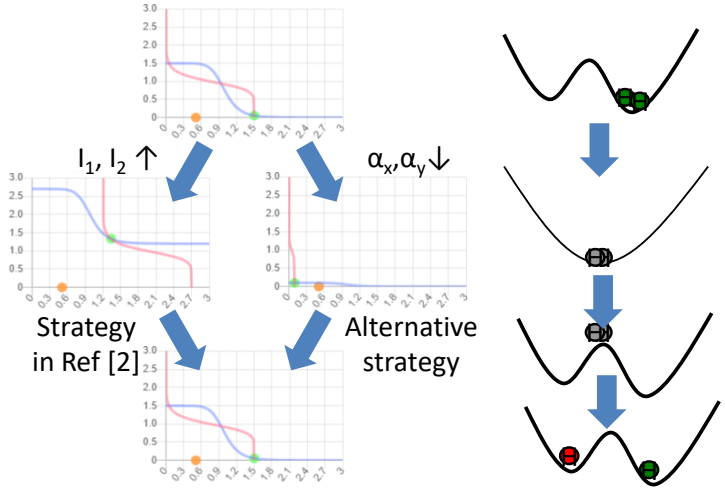


Figure 1 Inference Machine can generate our previous and alternative strategy for reprogramming

The research was funded by JST, CREST Grant Number JPMJCR21N4, Japan

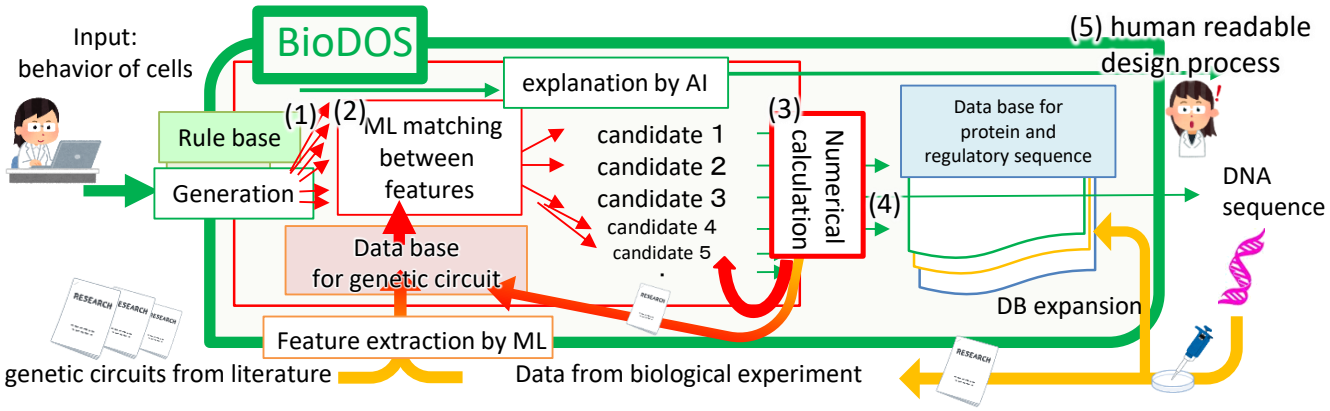


Figure 2: generation and screening process of candidate circuits for input behavior of cells

biology. We actually started semi-automatic collection of synthetic genetic circuit in synthetic biology literatures as described in the last part of this abstract. Those literatures are going to be used not only for expansion of the rule base, but for the screening of the generated candidate.

Although combinatorial explosion had known to be a weak point for generation by inference machine, we think that recent development of machine learning will allow appropriate screening of genetic circuit generated as candidates. Figure 2 shows our design process. Step 1: generation of candidates. Step 2: Screening by comparison of features between a designed circuit and each of combat proven circuits in data base. Step 3: numerical calculation for various sets of parameters for a topology of circuit. This step requires highest computational cost in the whole process. Thus pruning or ranking of candidates at step2 is important. Step 4: Corresponding to parameters of the screened candidate, appropriate biological components are selected from database. Step 5: Not only DNA sequence of the designed circuit, but design rules used by the Inference machine can be read by researchers.

Towards the construction of the database for genetic circuits, we started collection of network topology figures in synthetic biology articles. For semi-automated collection of articles, we used machine learning of network topology figures of related studies. As positive examples, we picked up 116 topology figures from 70 articles in ACS synthetic biology. As negative examples, we used figures of 85 ACS synthetic biology articles not related to genetic circuits. In order to avoid bias, we used similar number of negative articles to the positive ones. After training, 46 genetic circuit articles from other journals, as well as 185 negative example papers from ACS synthetic biology, were evaluated (Table 1). Further developments will allow more accurate classification.

Future Directions

We also started implementation of Prolog code to generate candidate circuits and parameters in Figure 1. Accumulation of such codes will allow biology-background researcher to write new rules for the rule base and to implement new circuits showing what life could be.

Table 1: Classification of genetic circuit papers

	percentage correctly classified
Positive example papers	61.8
negative example papers	63.3

REFERENCES

[1] Ryoji Sekine, Masayuki Yamamura, Shotaro Ayukawa, Kana Ishimatsu, Satoru Akama, Masahiro Takinoue, Masami Hagiya, and Daisuke Kiga. 2011. Tunable synthetic phenotypic diversification on Waddington's landscape through autonomous signaling. PNAS. 108 (44) (October 2011) 17969-17973. https://doi.org/10.1073/pnas.1105901108

[2] Kana Ishimatsu, Takashi Hata, Atsushi Mochizuki, Ryoji Sekine, Masayuki Yamamura, and Daisuke Kiga. 2013. General Applicability of Synthetic Gene-Overexpression for Cell-Type Ratio Control via Reprogramming. ACS Synth. Biol., 3, 9, (December 2013) 638–644 https://doi.org/10.1021/sb400102w

Experimental Data Converter (EDC)

Sai P. Samineni[*]
Gonzalo Vidal[*]
sasa6749@colorado.edu
g.a.vidal-pena2@newcastle.ac.uk
University of Colorado Boulder
Newcastle University

Jeanet Mante
University of Colorado Boulder
jet@mante.net

Carlos Vidal-Céspedes
Newcastle University
carlos.vidal.c@ug.uchile.cl

Guillermo Yañez-Feliú
Newcastle University
g.yanez-feliu2@newcastle.ac.uk

Timothy J. Rudge[†]
Newcastle University
tim.rudge@newcastle.ac.uk

Chris Myers[‡]
University of Colorado Boulder
chris.myers@colorado.edu

1 INTRODUCTION

Synthetic biology (SynBio) aims to engineer biological systems in a predictable and reproducible way. To this end, software tools are needed to aid researchers to iterate the *design–build–test–learn* (DBTL) cycle. The *Synthetic Biology Open Standard* (SBOL) is a free and open-source standard for the representation and electronic exchange of information on the structural and functional aspects of biological designs [1]. The SBOL community has developed an ecosystem of tools for automating and connecting different stages of the DBTL cycle, such as SBOLCanvas [5], iBioSim [7], LOICA [6], pySBOL3 [4], SynBioHub [3] and Flapjack [8]. Two of these tools, SynBioHub and Flapjack, are connected by the software application, the *Experimental Data Converter* (*EDC*), described here.

One critical aspect of the development of synthetic biology applications is *experimental workflow automation* (*EWA*). EWA is a set of software tools to capture the underlying information needed to build and test genetic circuits in a reproducible manner. The use of software is essential to manage the development of complex SynBio applications. Experimental data repositories are used to increase the reproducibility of DBTL iterations. There are several types of experimental repositories including ones focused on experimental metadata and ones focused on experimental measurement data. They overlap in some ways however they are generally not connected.

Experimental data repositories that support the build and test iterations of experimental workflows include SynBioHub [3] and Flapjack [8]. SynBioHub is a web application that provides a repository for DNA sequences, biological parts and devices, strains, experimental setup information, and other metadata (https://synbiohub.org). Flapjack is a data management system that enables researchers to store, visualize, analyze, and share genetic circuit experimental data, including measurement data and corresponding metadata (http://flapjack.rudge-lab.org/). Both SynBioHub and Flapjack are accessible by API frameworks to facilitate development and integration with the wider computational synthetic biology environment.

The current Synbiohub-Flapjack workflow is a manual, time-consuming process that requires new data imports for each new experimental context. This is partially due to the fact that Flapjack's front-end interface has not yet implemented a bulk upload functionality. Thus, having more than one assay implies uploading each Excel file individually.

Here we present a new software tool, *EDC*, for experimental data capture using Excel, Flapjack, and SynBioHub. The *EDC* provides researchers with an Excel template to capture both experimental results and contextual metadata. These data are then converted to a uniform data representation for the SBOL standard and Flapjack's data model. Through SBOL, users of the *EDC* can store experimental results with sequence, part, and other metadata information. Additionally, they can retrieve the data and share with others, improving reproducibility and collaboration.

2 RESULTS

EDC, illustrated in Figure 1, has two parts, the *template spreadsheets* and the *converter software*. The *template spreadsheets* are a set of spreadsheets that are designed to fit a wide range of experimental SynBio data. The spreadsheets are agnostic to the equipment that generated the data allowing the use of plate readers from different brands, flow cytometers, or any other equipment. The spreadsheets map all their fields to standardized metadata represented using SBOL [1] and the Flapjack data model [8]. The mapping can be modified by changing the column definitions on the first sheet.

[*]Both authors contributed equally to this research.

[†]GV, CVC, GYF and TR are supported by the Newcastle University School of Computing, ANID PIA Anillo ACT192015 and ANID Fondecyt Regular 1211598.

[‡]SS, JM, and CM are supported by the National Science Foundation Grant No. 1939892. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

The *converter software* was developed to read the spreadsheets, which upload both the metadata and measurement data into Flapjack. The metadata in the spreadsheets can also be converted using the existing Excel-to-SBOl converter [2] (https://github.com/SynBioDex/Excel-to-SBOL) to a standard SBOL format. If both pieces of software are used then the metadata stored in SynBioHub can be linked to study designs and measurement data stored in Flapjack in future. The converter software creates Flapjack objects with the SBOL data representation, pyFlapjack (https://github.com/RudgeLab/pyFlapjack) and excel2flapjack module (https://github.com/SynBioDex/Experimental-Data-Converter).

An improved workflow for capturing and representing experimental data and metadata will be established using the *EDC*. An example of this workflow could be as follows: a researcher designs an experiment with different promoters to measure gene expression. The researcher builds the necessary plasmids, transforms into a chassis, and tests them measuring fluorescence on a plate reader. Then, the researcher enters the experimental measurement data and metadata in the *template spreadsheets*, manually or in a semi-automated way. Researchers can then relate experimental measurement data to the conditions of the experimental study including, how the DNA was built, the parts used to create the DNA, the original developer of the parts, and other additional metadata. Next, the researcher uses the *converter software* to both build standardized SBOL objects and upload them to Flapjack. The same spreadsheet maybe used as the input for Excel-to-SBOL which supports metadata outputs as standardized SBOL objects. The SBOL is uploaded to SynBioHub. Representing the metadata in SynBioHub alongside the measurement data in Flapjack through the converter enables the analysis of the measurement data in Flapjack to be more reproducible with the capture of the study setup, sample designs and other design data. In this way, the user is able to both identify the implementations of DNA parts employed in the study, and access the underlying experimental data results through links which may be developed for measurement and analysis data in Flapjack.

3 DISCUSSION

EDC is a new tool that utilizes an Excel template for creating and uploading standardized experimental measurement data and metadata. It enables the use of digital data storage through multiple repositories, such as Flapjack and the associated standards by a wide range of researchers, particularly those with limited coding skills. Finally, the tool supports the use of the SBOL standard to capture build and test experimental data without prior knowledge of SBOL.

EDC supports the creation of new versions of the spreadsheets that can convert the data into SBOL3 and upload data to future versions of Flapjack, or other repositories. The initial spreadsheet templates (for metadata and measurement data) are provided to researchers in the *EDC* repository (https://github.com/SynBioDex/Experimental-Data-Converter). Further information on the development of the spreadsheets is described in prior work [2].

EDC provides a semi-automated workflow to assist researchers with the transition from the test to the learn stage. However, more tools and workflows are needed to fully connect the DBTL cycle. In the future, we plan to expand linkage between SynBioHub and Flapjack repositories by incorporating an SBOL URI field to all the Flapjack objects and vice versa to enable more robust storage and sharing of data. We envision a more connected and automated DBTL cycle in research laboratories and industry applications. Automation tools like the liquid handling robots and lab management systems will automatically capture relevant metadata. This workflow will coexist with spreadsheet metadata and measurement data capture to improve the reproducibility of experiments.

REFERENCES

[1] Galdzicki, M., et al. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology 32*, 6 (2014), 545–550.

[2] Mante, J., Abam, J., Samineni, S. P., Pötzsch, I. M., Singh, P., Beal, J., and Myers, C. J. Excel-sbol converter: Creating sbol from excel templates and vice versa. *bioRxiv* (2022).

[3] McLaughlin, J. A., Myers, C. J., Zundel, Z., Misirli, G., Zhang, M., Ofiteru, I. D., Goñi-Moreno, A., and Wipat, A. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synthetic Biology 7*, 2 (Feb. 2018), 682–688.

[4] Mitchell, T., Beal, J., and Bartley, B. pySBOL3: SBOL3 for Python Programmers. *ACS Synthetic Biology 11*, 7 (July 2022), 2523–2526.

[5] Terry, L., Earl, J., Thayer, S., Bridge, S., and Myers, C. J. SBOLCanvas: A Visual Editor for Genetic Designs. *ACS Synthetic Biology 10*, 7 (July 2021), 1792–1796.

[6] Vidal, G., Vidal-Céspedes, C., and Rudge, T. J. LOICA: Integrating Models with Data for Genetic Network Design Automation. *ACS Synthetic Biology 11*, 5 (May 2022), 1984–1990.

[7] Watanabe, L., Nguyen, T., Zhang, M., Zundel, Z., Zhang, Z., Madsen, C., Roehner, N., and Myers, C. iBioSim 3: A Tool for Model-Based Genetic Circuit Design. *ACS Synthetic Biology* (2019), 4.

[8] Yáñez Feliú, G., Earle Gómez, B., Codoceo Berrocal, V., Muñoz Silva, M., Nuñez, I. N., Matute, T. F., Arce Medina, A., Vidal, G., Vidal Céspedes, C., Dahlin, J., Federici, F., and Rudge, T. J. Flapjack: Data management and analysis for genetic circuit characterization. *ACS Synthetic Biology 10*, 1 (2021), 183–191. Publisher: American Chemical Society.

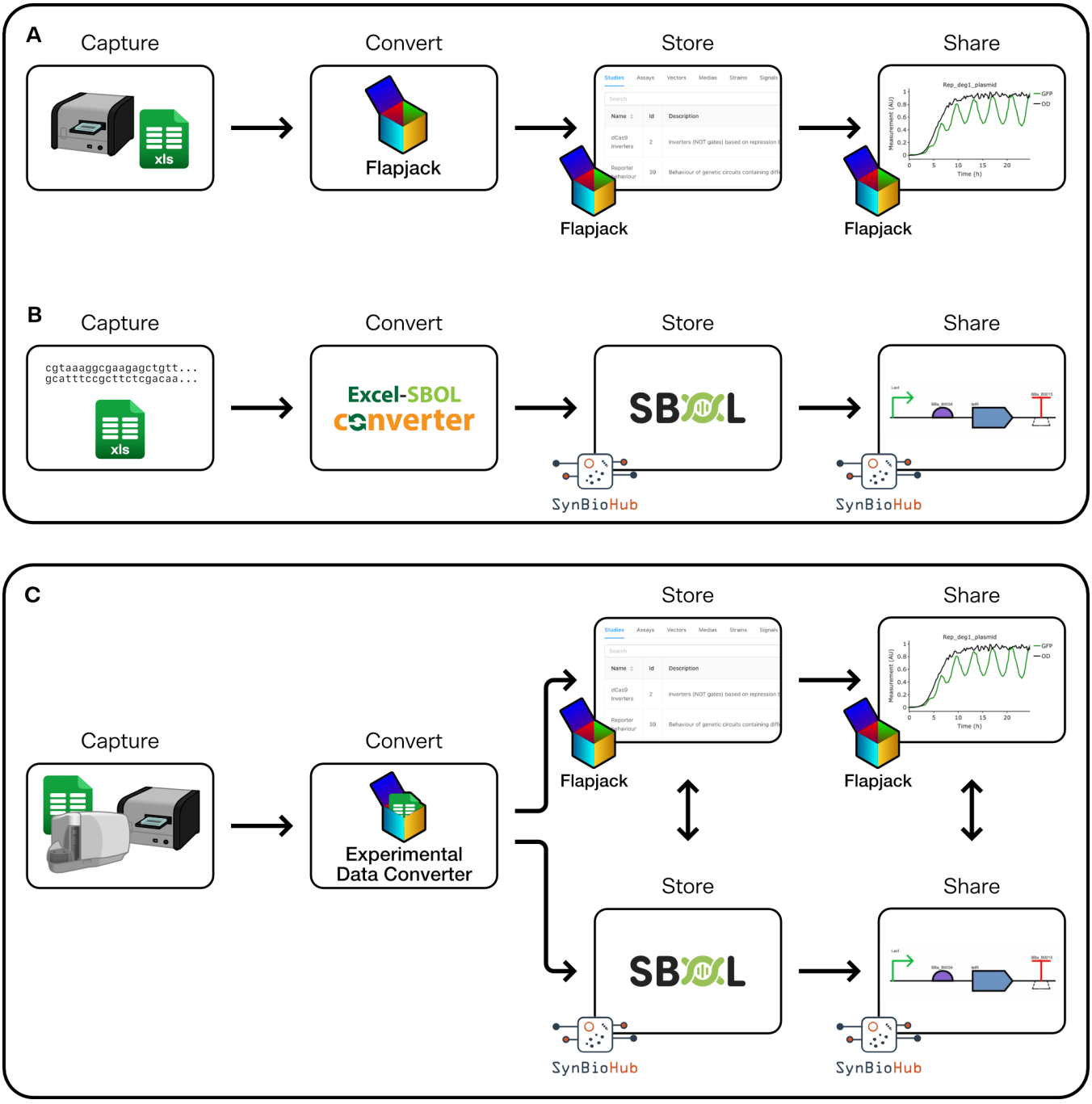


Figure 1: Diagram of the existing workflow (A and B) and the new proposed workflow (C) for capturing and connecting experimental data. A. Workflow for storing experimental data in Flapjack. The data is captured from a plate reader which delivers the readings in an Excel (xls) file, which is modified to incorporate the missing metadata necessary to be uploaded to Flapjack, this is then converted to the Flapjack data model and uploaded to the platform through the front-end and stored on the server. Once the data is stored, it can be viewed or shared. B. Workflow for storing data in SynBioHub. Genetic information is captured either through sequences obtained from text, GenBank or by using an Excel file. This file is then converted to an SBOL file by using the Excel-to-SBOL Converter, which allows it to be stored in SynBioHub and subsequently shared. C. Workflow using the *Experimental Data Converter*. Data can be obtained from various sources including a plate reader and fluorescence cytometer, which are then captured in an Excel file. This file is converted using the *Experimental Data Converter*, which in turn uploads and stores the experimental measurement data into Flapjack and at the same time generates an SBOL file with the experimental measurement data and metadata with genetic information that can be stored in SynBioHub.

Probabilistic programming for synthetic gene networks

Lewis Grozinger
l.grozinger2@ncl.ac.uk
Newcastle University
Newcastle-Upon-Tyne, UK

Ángel Goñi-Moreno
angel.goni@upm.es
Centro de Biotecnología y Genómica de Plantas,
UPM-INIA/CSIC
Madrid, España

1 ABSTRACT

Automated design of synthetic gene networks relies mathematical modeling frameworks used to characterise gene expression and transcriptional control. Simple deterministic models such as Hill equations are useful but ignore information about population heterogeneity and gene expression noise. Finding models which faithfully capture such information and have mechanistic interpretations that link them to the underlying genetic processes is still a challenge for synthetic biology. Here we fit probabilistic models to flow cytometry data to capture information encoded in the noisy fluorescence signals that cannot be captured deterministically. The models have rough physical interpretations and we apply these to study context-circuit dependencies with promising results.

2 INTRODUCTION

Synthetic gene networks have been used to construct genetic logic circuits from smaller networks known as genetic logic gates. Combinatorial design using these synthetic networks can be automated, so long as mathematical models of the individual gates are available and fitted to experimental data[5]. The automated design process therefore relies on significant effort in collecting data on the performance of individual gates in the library, and on the mathematical model chosen to characterise them.

The robustness of models and characterisations is fundamental to the design of genetic logic circuits and the libraries from which they are built. However they are not robust to the context to which they are deployed[6]. This is a bottleneck in the construction of libraries since experimental effort spent in the optimisation and measurement of the genetic parts must be repeated for each context in which the library is used. Model choice also affects robustness, even when using the library in the context in which it was first characterised. Past approaches to characterisation have used deterministic models based on point estimates of data, for example by fitting Hill equations to the medians of samples[5]. This approach ignores information about population heterogeneity and noisy gene expression[2], even though these are critical factors in failures of synthetic gene networks to perform as expected.

Models accounting for gene expression noise have previously been developed. Models based on Gamma distributions can be derived from the Master equation and enjoy physical interpretations of their parameters [3]. However the applicability of these models to synthetic gene networks is challenging because of the broad range of expression levels they exhibit. In the case of strong transcription the parameters of these Gamma models do not reflect our understanding of the mechanisms of gene expression [1].

Here probabilistic programs [4] characterise synthetic gene networks from experimental data obtained in three different contexts. The program adapts previous Gamma models[3] to include population heterogeneity. These programs capture information about gene expression noise and can provide insight into differences in gene expression parameters between identical synthetic gene networks in three different plasmid vectors. We present how the inclusion of population heterogeneity addresses some of the problems associated with modeling synthetic gene expression using Gamma distributions while retaining the physical relevance of the parameters of the model. We aim to investigate how to use these insights to predict performance differences between contexts without experimental data, and to use this workflow for combinatorial design of genetic circuits, using confidence intervals as part of the optimisation.

3 RESULTS

Constitutive expression network

A synthetic gene network was previously designed to constitutively express yellow fluorescent protein (YFP), placed into three different SEVA plasmids (pSeva221, pSeva231, pSeva251 in Figure 1A) and transformed into *P. putida* KT2440[6]. Twelve replicate flow cytometry experiments were performed to measure YFP expression from these three different synthetic genetic constructs.

The probabilistic program of Equations 1-3 adapts the Gamma distributed model as a mixture model which include population heterogeneity in the parameters from Figure 1A. Markov Chain Monte Carlo methods were used to estimate

parameters from the flow cytometry data. Fitted models predict probability distributions of expression and are readily embedded into other probabilistic programs for combinatorial design of more complex networks.

$$k_i \sim \text{LogNormal}(\mu_1, \sigma_1) \quad (1)$$

$$\theta_i \sim \text{LogNormal}(\mu_2, \sigma_2) \quad (2)$$

$$Y_i \sim \text{Gamma}(k_i, \theta_i) \quad (3)$$

Figure 1B shows the estimates for the means of the Gamma distribution parameters for all 36 different flow cytometry experiments. These estimates cluster according to SEVA plasmid, and since the parameters have physical interpretations as the number of expression bursts per cell cycle and the size of the bursts, these clusters potentially represent information on the contextual effects of SEVA plasmids on the gene network [3].

Induced expression network

Previous networks were modified to express YFP in response to induction by IPTG[6]. Flow cytometry was again used to measure expression the three SEVA plasmids and in the presence of twelve different IPTG concentrations, corresponding to levels of induction of the network. Figure 2A shows the results of fitting of the model to these experiments. It can be seen that $\frac{k_1}{\gamma_2}$ increases with IPTG as expected, since transcription rate k_1 should increase upon induction. A potential issue highlighted previously is that $\frac{k_2}{\gamma_1}$ is also seen to vary with induction under Gamma distribution models [1], where intuitively this should not be the case. Here the introduction of population heterogeneity seems to mitigate this problem, while retaining the physical interpretation for the parameters of Figure 1A.

Figure 2A was used to propose the following probabilistic program for induced gene expression, in which $\frac{k_2}{\gamma_1}$ is identically distributed for all IPTG concentrations, and the mean of $\frac{k_1}{\gamma_2}$ increases with IPTG concentration according to a Hill function.

$$\alpha_i \sim \text{LogNormal}(\mu_1, \sigma_1) \quad (4)$$

$$K_i \sim \text{LogNormal}(\mu_2, \sigma_2) \quad (5)$$

$$\epsilon_i \sim \text{LogNormal}(\mu_3, \sigma_3) \quad (6)$$

$$\beta_i \sim \text{LogNormal}(\mu_4, \sigma_4) \quad (7)$$

$$f(x) = \alpha x/(K + x) \quad (8)$$

$$Y_i \sim \text{Gamma}(f(x_i) + \epsilon_i, \beta_i) \quad (9)$$

Where x is a vector of log transformed IPTG concentrations, α_i and K_i are the maximum transcription rates and the half maximal induction concentration for the i^{th} sample. This model is similar to the constitutive model in that it is a Gamma distribution, but the shape parameter of the distribution increases with IPTG concentration.

Fitting produces a probabilistic program predicting the distribution of gene expression for any IPTG concentration. An example is shown in Figure 2B for the case of the network in pSEVA251. We suggest this model is more useful for predicting the effects of noise in combinatorially designed gene networks.

4 DISCUSSION

Probabilistic programming provides a framework for modeling the distributions of synthetic gene networks that makes full use of available flow cytometry data. The approach also lends itself well to building complexity by composition of different probabilistic programs to model more complex networks. Previously statistical models based on Gamma distributions have been found unsuitable for modeling synthetic gene networks with transcriptional repression. The networks have a broad range of expression levels, and it is suggested that stochasticity of transcription may not be the dominant source of noise [1] in these cases. Here we modify a Gamma distribution model to capture heterogeneity in gene expression parameters at the population level, and fit the model to inducible (rather than repressible) synthetic gene networks that also exhibit a broad range of expression levels. The results are more promising in this case, and have reasonable physical interpretations using the analytical derivation of the Gamma distribution model from the schematic of Figure 1A[3]. This physical interpretation is important to make sense of the differences in genetic processes between different contexts, which manifest as the context dependent clusters of parameters identified using the models presented here.

REFERENCES

[1] Beal, J. Biochemical complexity drives log-normal variation in genetic expression. *Engineering Biology 1*, 1 (2017), 55–60. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/enb.2017.0004.

[2] Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. Stochastic Gene Expression in a Single Cell. *Science 297*, 5584 (Aug. 2002), 1183–1186.

[3] Friedman, N., Cai, L., and Xie, X. S. Linking stochastic dynamics to population distribution: an analytical framework of gene expression. *Physical Review Letters 97*, 16 (Oct. 2006), 168302.

[4] Ge, H., Xu, K., and Ghahramani, Z. Turing: A Language for Flexible Probabilistic Inference. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (Mar. 2018), PMLR, pp. 1682–1690. ISSN: 2640-3498.

[5] Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., and Voigt, C. A. Genetic Circuit Design Automation. *Science 352*, 6281 (2016), aac7341–aac7341.

[6] Tas, H., Grozinger, L., Stoof, R., de Lorenzo, V., and Goñi-Moreno, Contextual dependencies expand the re-usability of genetic inverters. *Nature Communications 12*, 1 (Jan. 2021), 355. Number: 1 Publisher: Nature Publishing Group.

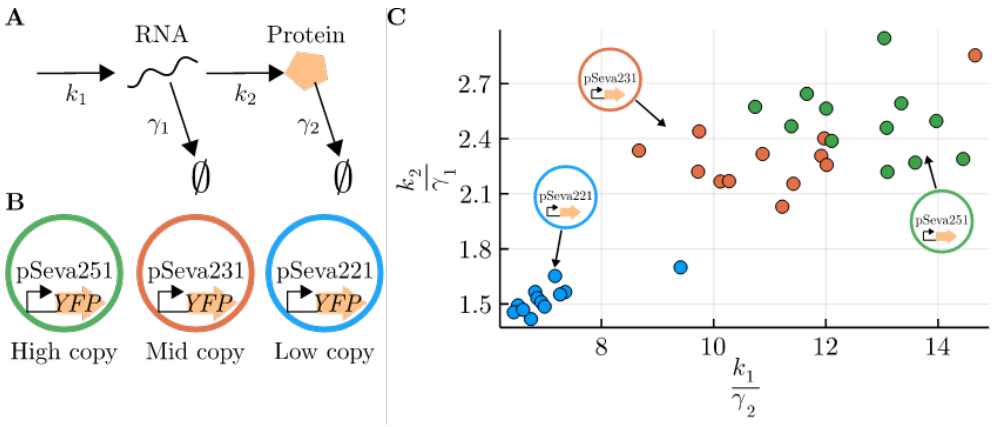


Figure 1: A simple synthetic genetic network which constitutively expresses YFP is modeled under a simple scheme shown in A. It is from this model that the Gamma distribution model of constitutive gene expression is derived. The gene network is placed in three different SEVA plasmids with high, medium and low copy numbers shown in B. The constitutively expressed YFP from the network is measured in twelve different flow cytometry experiments in each of these three contexts. In C the model of Equations 1-3 is fitted to samples from each flow cytometry experiment. Plotted are the parameters μ_1 and μ_2, which correspond to the population means of $\frac{k_1}{\gamma_2}$ and $\frac{k_2}{\gamma_1}$ from A. These are coloured according to SEVA plasmid, revealing clustering of parameter values that can be attributed to the contextual effects of the plasmids.

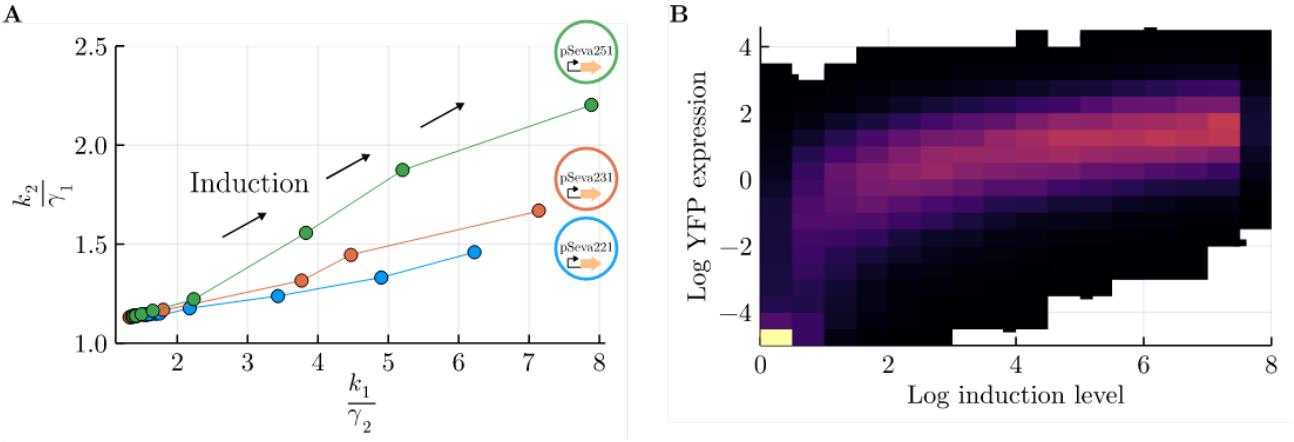


Figure 2: A synthetic gene network which expresses YFP and is induced by IPTG. This network is placed in three different SEVA plasmids and expression measured under 12 levels of IPTG induction. In A the model of Equations 1-3 is fitted to samples from each flow cytometry experiment, and again μ_1 and μ_2 is plotted for each. Three distinct response curves emerge for each of the SEVA plasmids, where $\frac{k_1}{\gamma_2}$ (number of transcription bursts) increases with the level of induction. B uses the model in Equations 4-9 fitted to the network in pSEVA251 to predict the distribution of YFP expression for any induction level and interpolate between the particular IPTG level observed in experiment.

A Conceptual Interactive Microfluidic Design and Control Workflow

Yangruiuri Zhou
Boston University
Boston, Massachusetts
yrrzhou@bu.edu

Douglas Densmore*
Boston University
Boston, Massachusetts
dougd@bu.edu

1 INTRODUCTION

Microfluidic devices represent a powerful tool for miniaturizing processes in the life sciences, including biosensor deployment and therapeutics screening. These devices are proposed to reduce the cost, time, and difficulty of automating experiments. To increase the benefit from small-scale experimentation, researchers try to parallelize multiple tasks in a single microfluidic device. However, this process increases the difficulty of correctly designing, in-silico validating and controlling the devices. Compared with the traditional iterative procedure, this proposal provides a new conceptual interactive workflow for designing and controlling microfluidic devices. This workflow has lower costs and fewer iterations using an explicit "simulation" module. The new workflow can provide feedback to help refine the biochip design in the interactive design stage. For the interactive control stage, the workflow provides manual control instructions, automatic software for controlling hardware pumps, and real-time control responses so that users rely less on microfluidic expertise and experience. In general, this workflow and future supporting software will benefit researchers with reproducible experiment control instructions, confidence, and less costs.

2 MOTIVATION

The quality of a microfluidic device design depends on many conditions, like the geometric parameters of the components inside, the layout of microfluidic components, etc. The traditional iterative procedure includes 7 steps:

(1) **Start** the experiment with a problem
(2) **Specify** the ideal microfluidic device architecture
(3) **Design** the microfluidic device for experiments
(4) **Build** the device
(5) **Test**the physical version of it
(6) **Execute** experiments on it
(7) **Archive** the final design for future use

Researchers published many different approaches to help solve some specific problems in the 7-step procedure, like auto-design algorithms [2] for the *Design* stage, fabrication-aid design tools for the *Build* stage [3], and control tools [1] for the *Execute* stage, etc. However, the lack of an in-silico validation stage makes it hard to find the potential logic error inside a complex design before testing the fabricated device. After physical validation, the researchers have to redesign the devices if they find an unacceptable error inside the device, which makes a costly **DBT loop** among the 3 stages: *Design*, *Build*, and *Test* (**Figure 1**). To reduce the cost and increase the confidence with the microfluidic experiments, we propose a *Simulation* module for the traditional procedures to provide a visual check prior to the *Build* stage and a real-time control approach to the devices.

3 METHOD

In this conceptual workflow, the *Simulation* module only accepts 3 types of input files: *Design*, *User requirement (UR)* and *Constraint*. A *UR* describes the goal of an experiment on the biochip as a starting and ending location. *Constraint* specifies the state of a single *Control-layer Component* or logical relationship between multiple *Control-layer Components*. From the figure 1, there are two bi-arrow connections between the *Simulation* module and the traditional workflow, we call them pre-fabricate (interactive design) and post-fabricate (interactive control).

We make interactive design possible by providing verification feedback according to the *UR* and *Constraints* when users want to check their current design. Users will get an error feedback if the *Simulation* module finds conflict among *Constraints*, *Design*, and *UR* (*CDR Conflict*). If no *CDR Conflict* appears, to make the verification result more convincing, it will provide a visual simulation result on the GUI for the user to check. After the users accept the result(s), the *Simulation* module can generate an automated hardware script and manual control instructions to the *Execute* stage. Thus, users can reduce the cost by figuring out the operation details before fabricating and validating the physical device.

For the post-fabricate interaction, the *Simulation* module can provide a real-time actuation response on the *Control-layer Component* when the users modify the device on the GUI after connecting hardware, software and microfluidic devices together. All the changes are transformed as modified *UR* and *Constraints*, then the *Simulation* module will perform the verification and simulation again. If a *CDR Conflict* appears, users can identify and correct it on the GUI. If not, users can visualize the simulation result on the GUI and the same physical changes on the device.

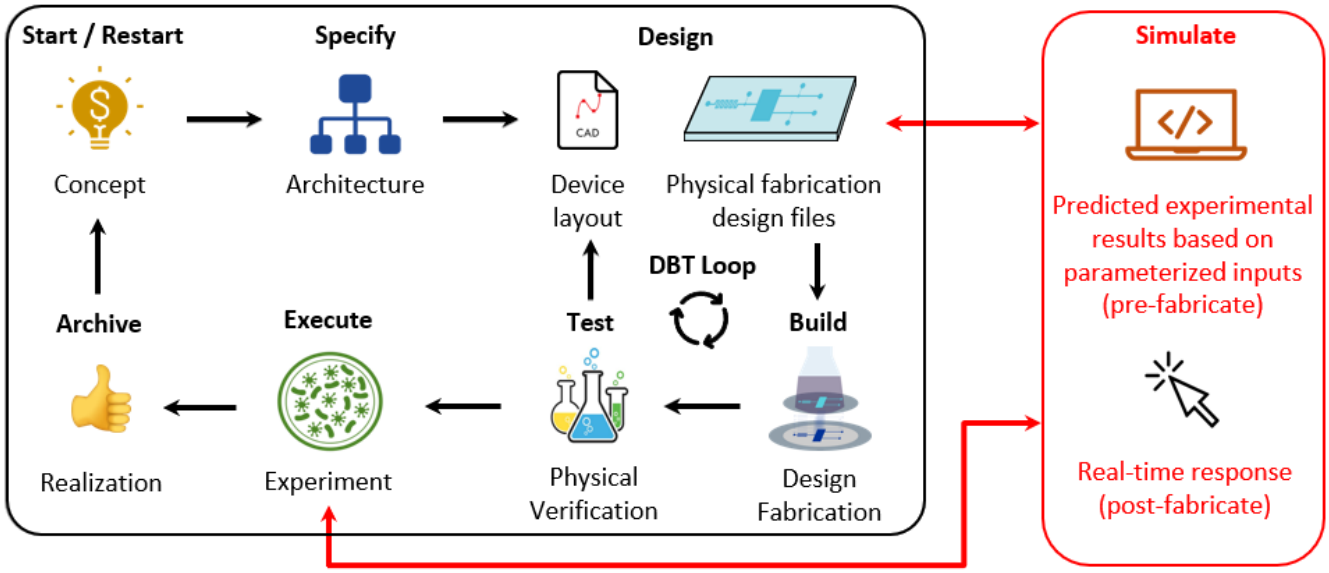


Figure 1: The black box shows the traditional procedure of doing an experiment on a microfluidic device. Some stages in the black box can go back and forth in practice, but the sequence in logic is shown as above. The red box shows additional analysis coming after we import a new module called "Simulate". To reduce the cost in the DBT loop (shown as a cycle), we add the *Simulation* module into the traditional procedure. The interaction between the traditional stages and the *Simulation* module is through file transfer. The detailed I / O information between the two boxes can be found in Figure 2.

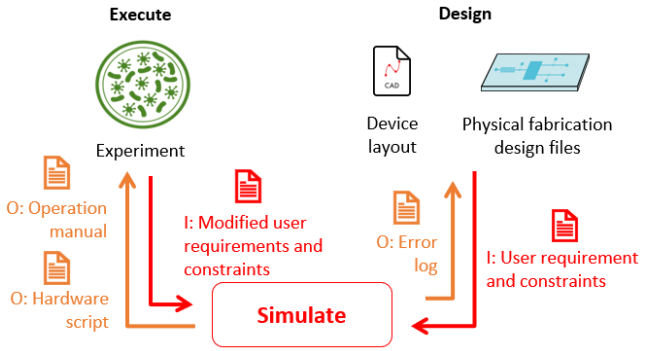


Figure 2: I/O Details in the workflow. I: input for the *Simulation* module, O: output of the *Simulation* module. Only two stages have interactions with the *Simulation* module.

4 CONCLUSION AND FUTURE WORK

This proposal provides a new conceptual interactive biochip design workflow to reduce the cost, increase user confidence, and improve experiment robustness, even if users are not experts in microfluidics. Similar to the pre-silicon test in electronic chip design, this workflow provides possible experiment results or errors given the *Design*, *User requirement*, and *Constraints*. Users can check their designs in this workflow and make changes according to the feedback. Additionally, it provides an automated hardware script and manual control instructions to help the users complete their experiment, which also makes the whole workflow reproducible and robust. After connecting software, hardware, and devices together, users can get real-time responses on the device when they make changes on the GUI. Finally, we will build a system to implement this interactive design and control workflow. We believe this system can benefit both inexperienced and experienced researchers. Our newbie-friendly system allows inexperienced users to make mistakes and practice at a lower cost. Experienced microfluidic researchers can use it to learn more about their experiments before fabricating the devices.

5 ACKNOWLEDGEMENTS

This work was supported by NSF [Grant #2211040]. The authors thanks all CIDAR lab members for their advice throughout this proposal.

REFERENCES

[1] Brower, K., Puccinelli, R. R., Markin, C. J., Shimko, T. C., Longwell, S. A., Cruz, B., Gomez-Sjoberg, R., and Fordyce, P. M. An open-source, programmable pneumatic setup for operation and automated control of single-and multi-layer microfluidic devices. *HardwareX 3* (2018), 117–134.

[2] Lashkaripour, A., Rodriguez, C., Mehdipour, N., Mardian, R., McIntyre, D., Ortiz, L., Campbell, J., and Densmore, D. Machine learning enables design automation of microfluidic flow-focusing droplet generation. *Nature communications 12*, 1 (2021), 1–14.

[3] Sanka, R., Lippai, J., Samarasekera, D., Nemsick, S., and Densmore, D. 3dμf-interactive design environment for continuous flow microfluidic devices. *Scientific reports 9*, 1 (2019), 1–10.

Standardizing the Representation of Parts and Devices for Build Planning

Jacob Beal[1], Vinoo Selvarajah[2], Gael Chambonnier[3], Traci Haddock-Angelli[2], Alejandro Vignoni[4], Gonzalo Vidal[5], Nicholas Roehner[1]
[1]Raytheon BBN, [2]iGEM Foundation, [3]Massachusetts Institute of Technology, [4]Universitat Politecnica de Valencia, [5]Newcastle University
jakebeal@ieee.org,vinoo@igem.org,gchambon@mit.edu,traci@igem.org,
vignoni@isa.upv.es,g.a.vidal-pena2@newcastle.ac.uk,nicholas.roehner@raytheon.com

1 MOTIVATION

One of the most common tasks in synthetic biology is building genetic constructs by assembling smaller parts. Despite this commonality, however, there is often much confusion when practitioners communicate about parts, sequences, and build plans. Parts often go through many stages during a build process, each with a different sequence. For example, a fragment of DNA may be synthesized as an insert into a vector backbone, then digested out of that backbone and assembled together with other fragments to produce a final construct. At present, without a shared standard for describing build plans, it is often difficult to tell which stage a given sequence is describing, leading to frequent confusion, errors, difficulty sharing information, and waste.

We address this problem with a standard vocabulary for describing build plans, which we have further mapped into a concrete representation using the SBOL 3 standard [4]. Specifically, we target representation of assembly based on digestion and ligation, supporting at least BioBricks Assembly [6] and Type IIS assemblies like GoldenGate [1], MoClo [7], and GoldenBraid [5]. The resulting vocabulary should be useful to practitioners no matter what tools or representations they may be using, while representation in SBOL 3 provides full details for use by software tool builders.

2 STANDARDIZING TERMINOLOGY

Our first target of standardization is the terminology used for describing DNA at different stages of build planning. Developing this vocabulary was motivated by challenges in developing the iGEM 2022 distribution, where we found many miscommunications between collaborators about how sequences related to our build plans (e.g., did a sequence already include flanking sequences, was this what should be synthesized or what it would look like after insertion into a backbone, etc.). To this end, we have proposed the following definitions, cleaving as closely as possible to pre-existing patterns in descriptions, and aligning with typical digestion/ligation build planning as shown in Figure 1:

- **Part:** Design for a single contiguous linear DNA construct with a completely specified sequence.
- **Unitary Part:** Any part that is not designed with reference to an assembly, often but not always having a well-defined role such as a CDS or promoter.
- **Composite Part:** A part designed as the composition of two or more other parts through an assembly plan.
- **Assembled Part:** A part, plus any 5' or 3' flanking scars, in the post-assembly context of a composite part.
- **Scar:** A sequence that is produced by the combination of flanking sequences in an assembly.
- **Backbone:** A DNA construct into which parts are intended to be inserted at one or more designated insertion sites (often, but not always, a circular plasmid).
- **Drop-Out Sequence:** A portion of a backbone at an insertion site that is removed when a part is inserted at that site. Some backbones include drop-out parts while others do not.
- **Part Insert**: A part, plus any 5' and 3' flanking sequences, that is intended to be placed into a designated insertion site of a backbone.
- **Part in Backbone:** A backbone with at least one insertion site where a part insert has been incorporated.
- **Part Extract:** A part, plus any 5' or 3' flanking sequences, that has been extracted from a part in backbone as part of an assembly process.
- **Assembly:** A plan for combining a set of parts in order to build one or more composite parts.

In the iGEM Engineering Committee, we found that agreeing on this common terminology greatly reduced the amount of confusion, and use of these terms has become commonplace in our multi-institution collaboration.

3 REPRESENTING ASSEMBLY PLANS IN SBOL

To facilitate better tool support for planning and communicating build information, we mapped the vocabulary and build plans shown in Figure 1 onto the SBOL 3 standard [4], which we found to provide all of the concepts necessary for a succinct representation. Here we present a summary of key points; full details are available as SBOL Best Practice Proposal (BPP) 001 in the SBOL Examples collection at https://github.com/SynBioDex/SBOL-examples/pull/4.

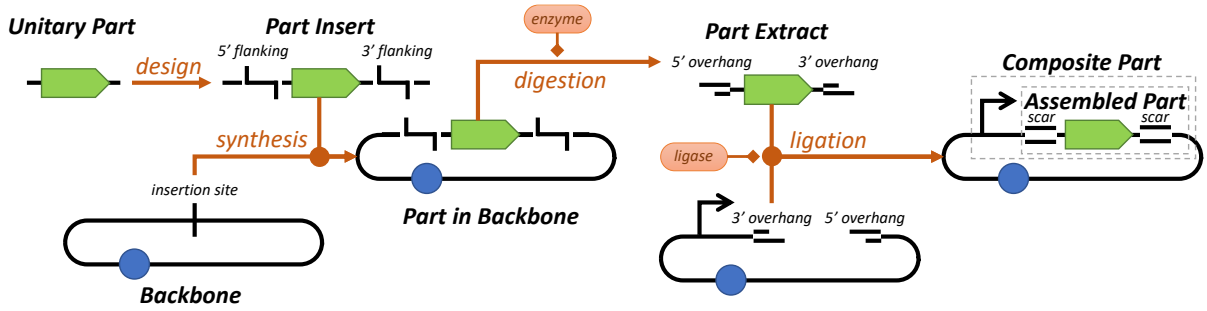


Figure 1: Proposed build terminology, illustrated on a typical digestion/ligation build workflow: a *unitary part* is extended with flanking sequences needed for assembly to create a *part insert* that can be synthesized or assembled into an insertion point on a *backbone* to produce a *part in backbone* ready for assembly. Digestion produces a *part extract* that can be ligated together with other part extracts to produce a *composite part* in backbone, including the original part as an *assembled part* in its final context.

In this representation, each part is an SBOL `Component`, and the distinction between a unitary part and a composite part can be made by using the `prov:wasGeneratedBy` property to link any composite part to a `prov:Activity` representing an assembly plan, as described below. An assembled part within the composite part is represented by an appropriate `Feature` (typically a `SubComponent`), and similarly a scar is a `SequenceFeature` with its role set to the Sequence Ontology (SO) term for assembly scar.

A backbone is also represented by an SBOL `Component`, but has a role indicating its use as a backbone, such as `SO:plasmid_vector`. An insertion site or drop-out sequence is indicated using a `Feature` with the corresponding role, respectively `SO:insertion_site` and `SO:deletion`. Part in backbone and part insert are much the same, represented with a `Component` that includes a `SequenceFeature` for each restriction site (with SO restriction site terms), while a part extract will typically have features for overhangs.

Finally, an assembly plan is represented by a `prov:Activity` with appropriate typing and a link to an SBOL `Component` describing the network of digestion and ligation reactions for the assembly. Each reaction can be described by an `Interaction`, with each reactant, enzyme, and product a `Participant`. Digestion uses type `SBO:cleavage`, with the part in backbone and enzyme having role `SBO:reactant` and the part extract having role `SBO:product`. Ligation uses type `SBO:conversion`, with the part extracts and ligase being the reactants and the composite part in backbone being the product. Many composite parts will be described with just one digestion/ligation stage, but a more complex assembly may have multiple digestion and ligation stages and may have multiple products.

4 FUTURE DIRECTIONS

The proposal has met with general consensus during community review and is currently in process of being adopted as a best practice officially endorsed by the SBOL community. A full supporting Python API is currently being implemented for the SBOL Utilities library (https://github.com/SynBioDex/SBOL-utilities). This implementation is intended to form the basis for integration of these representations with laboratory automation. Finally, while the current proposal has been worked out specifically with regards to Type IIS and BioBricks assembly methods, we believe it is likely to extend well to other assembly methods as well, such as Gibson Assembly [2] or Ligase Cycling Reaction Assembly [3], though certain details will likely need to be adjusted.

5 ACKNOWLEDGEMENTS

This work was supported in part by AFRL and DARPA contract FA8750-17-C-0184. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

REFERENCES

[1] Engler, C., Kandzia, R., and Marillonnet, S. A one pot, one step, precision cloning method with high throughput capability. *PloS one 3*, 11 (jan 2008), e3647.

[2] Gibson, D. G., Young, L., Chuang, R.-Y., Venter, J. C., Hutchison, C. A., and Smith, H. O. Enzymatic assembly of dna molecules up to several hundred kilobases. *Nature methods 6*, 5 (2009), 343–345.

[3] Kok, S. d., Stanton, L. H., Slaby, T., Durot, M., Holmes, V. F., Patel, K. G., Platt, D., Shapland, E. B., Serber, Z., Dean, J., et al. Rapid and reliable dna assembly via ligase cycling reaction. *ACS synthetic biology 3*, 2 (2014), 97–106.

[4] McLaughlin, J. A., Beal, J., Misirli, G., Grünberg, R., Bartley, B. A., Scott-Brown, J., Vaidyanathan, P., Fontanarrosa, P., Oberortner, E., Wipat, A., et al. The synthetic biology open language (sbol) version 3: simplified data exchange for bioengineering. *Frontiers in Bioengineering and Biotechnology 8* (2020), 1009.

[5] Sarrion-Perdigones, A., Falconi, E. E., Zandalinas, S. I., Juárez, P., Fernández-del Carmen, A., Granell, A., and Orzaez, D. Golden-braid: an iterative cloning system for standardized assembly of reusable genetic modules. *PloS one 6*, 7 (2011), e21622.

[6] Shetty, R., Lizarazo, M., Rettberg, R., and Knight, T. F. Assembly of biobrick standard biological parts using three antibiotic assembly. In *Methods in enzymology*, vol. 498. Elsevier, 2011, pp. 311–326.

[7] Weber, E., Engler, C., Gruetzner, R., Werner, S., and Marillonnet, S. A modular cloning system for standardized assembly of multigene constructs. *PloS one 6*, 2 (2011), e16765.

Adapting Malware Detection to DNA Screening

Dan Wyschogrod[1], **Jeff Manthey**[2], **Tom Mitchell**[1], **Steven Murphy**[1], **Adam Clore**[2], **Jacob Beal**[1]
[1]Raytheon BBN, Cambridge, MA USA, [2]Integrated DNA Technologies, Coralville, IA, USA
{dan.wyschogrod,tom.mitchell,steven.t.murphy}@raytheon.com,{aclore,jmanthey}@idtdna.com,jakebeal@ieee.org

1 MOTIVATION

As DNA synthesis becomes cheaper and more accessible, there is a corresponding increase in opportunities for synthesis of dangerous pathogenic sequences by either malicious or careless actors [2–4, 6, 8]. To mitigate this threat, major DNA synthesis providers screen sequence orders for pathogenic content, following guidance from the US Department of Health and Human Services [9] and the International Genome Synthesis Consortium (IGSC) [7].

Current methods for screening, however, have been unable to scale sufficiently to keep up. The current dominant method for screening is to evaluate sequence homology, using BLAST (or similar) to test if the sequence's best alignment is with a controlled pathogenic organism [2, 5, 8]. This approach produces a high rate of false positives, estimated at more than 4% from a survey of IGSC member companies [2], worsened by the fact that these methods generally search for all genes in an organism, including harmless "housekeeping" genes and others that have no functional relationship to pathogenesis. Moreover, the rate of false positives increases markedly as sequence length shortens [6]. Due to the cost of resolving false positives, synthesis providers thus typically only screen dsDNA sequences that are at least 200 bp long and do not screen oligonucleotides at all [2, 5].

We hypothesized that these challenges could be addressed by adapting methods for detection of malware in network traffic, which faces even greater challenges of scale. To this end, we adapted the Framework for Autogenerated Signature Technology (FAST) signature extraction method [10] for use with nucleic acid sequences, producing the FAST for Nucleic Acids (FAST-NA) method for DNA screening. Our resulting implementation of FAST-NA is able to detect DNA sequences far faster than BLAST-based methods, and with equivalent sensitivity and significantly improved specificity, even while reducing the minimum scanning window from 200bp to 50bp.

2 DEVELOPMENT OF FAST-NA

FAST begins by breaking collections of target and contrast material into small "signature" fragments. FAST stores the contrast signatures in a Bloom filter [1], a highly efficient data structure for testing set membership. The Bloom filter is then used to remove all target signatures that match any contrast signature, leaving only signatures that are diagnostic of threats. This proves highly effective for malware detection: even though polymorphic malware constantly mutates itself to try to evade detection, there are generally still some conserved sequences required for its function, which FAST is able to identify. Matching software using these signatures can then identify malware extremely rapidly and with high sensitivity and specificity.

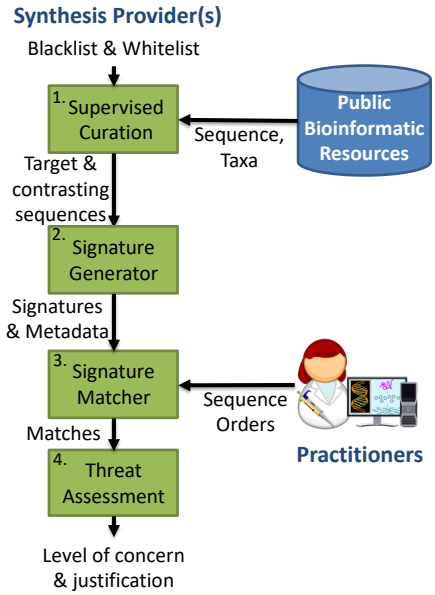
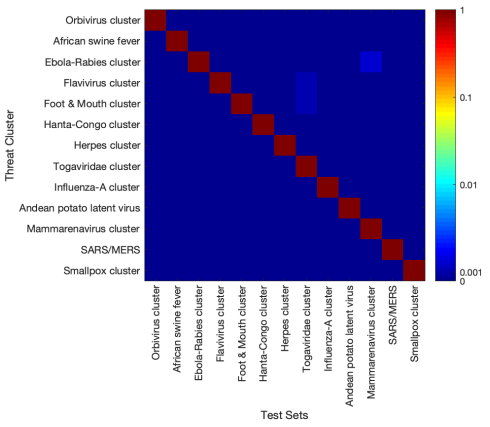


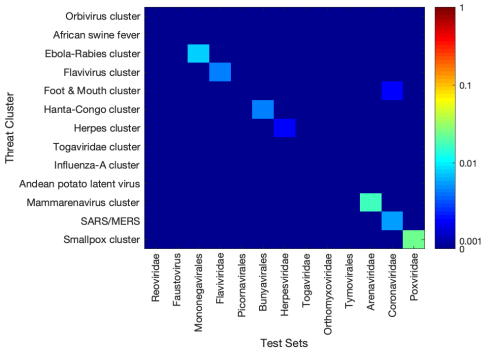
Figure 1: FAST-NA architecture: diagnostic signatures are identified by comparing target sequences to contrasting material, then applying these signatures in a matcher that scans sequence orders to assess their threat content.

Adapting this method for FAST-NA (Figure 1), we use nucleic acid and protein sequences from public databases such as NCBI as the source material, taking the target material from clusters of threat taxa to be detected and the contrasting material from other taxa that are closely related but not controlled. For example, SARS and MERS are in the coronavirus threat cluster, while the more benign human coronaviruses 229E and NL63 are in the coronavirus contrast collection. For signatures, we use k-mers, ranging from 26-42 base pairs for nucleic acids and 14-20 residues for amino acids.

Just as with malware, this process identifies signatures for conserved sequences defining the nature of a biological threat. These signatures, along with metadata on their origins, can then be given to a matcher that scans sequence orders to assess their threat content. With appropriate tuning and curation, this produces a signature collection that is both highly sensitive and highly specific.



(a) Threat Identification



(b) False Positives

Figure 2: Sensitivity and specificity of FAST-NA signatures for controlled viral pathogens: (a) probability of correct identification of threat sequences, (b) probability of false positives for closely related non-controlled sequences.

Figure 2 shows an example of FAST-NA performance, in this case for the set of all viral threats in the IGSC Regulated Pathogen Database. When comparing all 50+ bp viral threat sequences from NCBI and from close contrasting taxa, we find the signatures are highly sensitive, producing no false negatives. They are also highly specific: mean per-taxa likelihood that a threat is multiply identified is 0.039%, while the mean per-taxa likelihood of a false positive is 0.55%. Other kingdoms are not as clean as viruses—particularly the bacteria, which are highly prone to horizontal transfer—but the average all-threat rate for multiple identification and for false positives are both less than 2%, far lower than the typical 4% rate for BLAST-based screening despite the much-reduced screening window. Moreover, because it focuses only on diagnostic signatures, FAST-NA is able to scan >10 kilobases/second (orders of magnitude faster than BLAST) and with far less required computing resources.

The distribution of sequences in commercial synthesis orders is, of course, quite different than that found in sequences in NCBI. We have found, however, that the performance is maintained when applied to synthesis orders. A commercialized version of the system, named FAST-NA Scanner, is now deployed at IDT, and is seeing similar or better results when used against live customer data.

3 APPLICATIONS AND FUTURE DIRECTIONS

At present, the primary application of FAST-NA remains DNA synthesis order screening, with FAST-NA Scanner available from BBN as a commercial software product. In addition to the improvements in false positive rate, the high speed and low computational cost of FAST-NA can also enable other workflows that are impractical with BLAST-based scanning, such as online pre-order screening, secure on-site screening (e.g., in a benchtop synthesizer), and combinatorial screening of oligo assemblies. Finally, beyond synthesis order screening, we aim to further develop FAST-NA for other types of biosecurity applications, such as interpretation of sequencing data, incorporation of biosafety and biosecurity considerations into design tools, and threat scanning in information systems and laboratory management processes.

ACKNOWLEDGEMENTS

This work was partially supported by IARPA contract 2018-17110300002 and ARO grant W911NF-17-2-0092. Views and conclusions are of the authors and should not be interpreted as representing official policies, either expressed or implied, of ARO or the U.S. Government. Contains no technology or technical data controlled under U.S. ITAR or EAR.

REFERENCES

[1] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM 13*, 7 (1970), 422–426.

[2] Carter, S. R., and Friedman, R. M. Dna synthesis and biosecurity: lessons learned and options for the future. *JCVI* (2015).

[3] Carter, S. R., and Warner, C. M. Trends in synthetic biology applications, tools, industry, and oversight and their security implications. *Health security 16*, 5 (2018), 320–333.

[4] DiEuliis, D., Berger, K., and Gronvall, G. Biosecurity implications for the synthesis of horsepox, an orthopoxvirus. *Health security 15*, 6 (2017), 629–637.

[5] Diggans, J., and Leproust, E. Next steps for access to safe, secure dna synthesis. *Frontiers in bioeng. and biotech. 7* (2019), 86.

[6] Garfinkel, M. S., Endy, D., Epstein, G. L., and Friedman, R. M. Synthetic genomics: options for governance. *Industrial Biotechnology 3*, 4 (2007), 333–365.

[7] IGSC (International Gene Synthesis Consortium). Harmonized screening protocol v2.0. Available at https://genesynthesisconsortium.org/wp-content/uploads/IGSCHarmonizedProtocol11-21-17.pdf, November 2017. Accessed October 20, 2020.

[8] NASEM. *Biodefense in the age of synthetic biology*. National Academies Press, 2018.

[9] US Department of Health and Human Services. Screening framework guidance for providers of synthetic double-stranded dna. *Fed Regist 75*, 197 (2010), 62820–62832.

[10] Wyschogrod, D., and Dezso, J. False alarm reduction in automatic signature generation for zero-day attacks. In *2nd Cyberspace Research Workshop* (2009), p. 73.

Expanding the metaheuristic framework: evolving cells with the bat algorithm

Víctor Reyes, Nicolás Hidalgo, Martín Gutiérrez
Universidad Diego Portales
Santiago de Chile, Chile
{victor.reyes1,nicolas.hidalgoc,martin.gutierrez}@mail.udp.cl

1 INTRODUCTION

Cell colonies provide a suitable environment for the execution of costly algorithms, as they offer a large degree of parallelism. Nowadays, this is an important feature, since Artificial Intelligence (AI) algorithms require a high amount of computational processing power. In fact, cell-based AI algorithms have been reported recently [1, 2, 4, 6, 10, 11], and establish a starting point for developing in-cell automation of synthetic cell circuit organization and evolution. Metaheuristics are one type of AI techniques in which inspiration is drawn from physical and/or biological phenomena and translated into algorithm mechanics. This type of algorithms is well suited to be implemented in cell colonies, since many of the general tasks involved in these algorithms are already undertaken by cells and ease their interpretation. In-cell automated design for metaheuristic algorithms was reported in the form of a framework [10]. An implementation based on this work and oriented towards improving the fitness of the cell population itself was also recently published [9]. In this work, we adapt the bat algorithm, a swarm-based metaheuristic to integrate it into an existing framework. We also map the elements of the algorithm and test the implementation on simple instances of the set covering problem. An example of how this could be applied is in the identification of conditions for releasing programmable antibiotics (given conditions evaluated by this algorithm) as an alternative to other methods [7].

2 THE BAT ALGORITHM

The bat algorithm (BA) [12] is a bio-inspired metaheuristic algorithm oriented toward solution search and continuous optimization, inspired by the echolocation of bats. In this algorithm, each solution is encoded as a single bat, and this solution changes its position according to: 1) the direction found in which the fitness of the solution improves and 2) the best solution attained so far among all bats. This is modelled through frequency and velocity values associated respectively to the bat movement and the echolocation signal. The main idea driving this algorithm is to attempt an approximation towards a local maximum/minimum using these cues. It has been proved that this algorithm is able to outperform well-known algorithms, such as genetic algorithms and particle swarm optimization [12]. However, the original form of this algorithm is not well suited for binary problems. For this reason, in this work we use a discrete version of the BA, known as binary BA (BBA) [8]. The main difference from the original proposal is that the position of a bat is mapped to a discrete space by using transfer functions, such as sigmoids or v-shape functions. This version of the algorithm is implemented to faithfully map to the existing metaheuristic framework.

3 MAPPING WITHIN THE METAHEURISTICS FRAMEWORK

Following the paradigm established in the metaheuristics framework [10], the problem solutions are encoded as bacteria holding plasmids expressing proteins associated to a given bit value in the solution. The plasmids can be conjugated, for solution evolution, and its rate defines how fast the solution is updated. Echolocation is mapped to a Quorum Sensing (QS) signal. Bacteria are more sensitive to the signal when the solution is "far" from being a good one, to promote evolution. However, upon complying with more constraints, sensitivity decreases. This sensitivity is tied also to the growth of bacteria: if the bacterium contains a "bad" solution, it grows slowly, but upon approaching a better solution, it grows faster. This aids in identifying and replicating better solutions, but also indirectly speeds up bacterial conjugation since division time is faster. In the `gro` implementation built for this proof-of-concept, the division times chosen for different stages of evolution of solutions were: 0 plasmids contained - 1200 hours division time, 1 plasmid contained - 12 hours division time, 2 plasmids contained - 1 hour division time, 3 plasmids contained - 20 minutes division time. Also, mutations can be implemented at each circuit in the `gro` simulations to improve realism.

4 EXPERIMENTS AND RESULTS

The proof-of-concept mapping was implemented in `gro` [3, 5] simulations, but also in its original version using C++ to assess both implementations side by side. The problem solved by both implementations is a unicost binary set covering

problem. In the C++ implementation, each solution is represented as a binary array and evolution is carried out by mutating the array one element at a time, and assessing the direction and velocity through the fitness function $\sum_{i=1}^{n} x_i - \phi$, where $x_i \in \{0, 1\}$ and ϕ is a penalty function with respect to the number of unsatisfied constraints. In the `gro` implementation solutions and their evolution are encoded as described in the previous section. We worked on a small instance of the problem covering 3 elements of a set and in which the goal is to reach a set in which all elements are present. A comparison of such tests was made based on the number of generations it takes to reach the optimal solution, as was done in the metaheuristic framework [10] is shown in Tables 1 and 2, and visual execution of the `gro` version is shown in Figure 1. More simulation results can be found at data repository link.

Table 1: C++ Simulation results for measuring convergence generations for different values of echolocation signal frequency.

Frequency	Optimal value generations
25	2
50	2
100	2

Table 2: `gro` Simulation results for measuring convergence generations for different values of QS detection threshold (parameter mapping for echolocation frequency).

QS lower threshold	Optimal value generations
0.00005	0.29395897904
0.0001	0.27929299074
0.005	0.19029879164

5 CLOSING REMARKS AND OBSERVATIONS

Both solutions implement the bat algorithm, and the `gro` version makes use of the massive parallelism of cells in the simulation, along with native cell processes such as cell growth and intercell communication. In spite of further experiments needed to fully characterize the bat algorithm implementation, our first observation is that a cell-based version of the algorithm is able to operate in a similar manner to the original abstraction of the algorithm, shown by our proof-of-concept. Also, the bacterial version of the algorithm can be tuned by altering parameter values such as conjugation rate, growth rate and QS thresholds, making the algorithm general. Values in the tables are faster in the `gro` implementation due to two reasons: 1) initially there are 1000 cells evolving the solution, exhibiting a large amount of parallelism and, 2) heterogeneous division times and large amounts of cells account for faster convergence to optimal solutions.

6 ACKNOWLEDGEMENTS

The authors thank Matías Bravo, Ricardo Caballero and Nicolás Frías for running the experiments pertaining to this work.

REFERENCES

[1] Becerra, A. G., Gutiérrez, M., and Lahoz-Beltra, R. Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network. *Biosystems 213* (2022), 104608.

[2] Gargantilla Becerra, Á., Gutiérrez, M., and Lahoz-Beltra, R. A synthetic biology approach for the design of genetic algorithms with bacterial agents. *International Journal of Parallel, Emergent and Distributed Systems 36*, 3 (2021), 275–292.

[3] Gutiérrez, M., Gregorio-Godoy, P., Perez del Pulgar, G., Muñoz, L. E., Sáez, S., and Rodríguez-Patón, A. A new improved and extended version of the multicell bacterial simulator gro. *ACS synthetic biology 6*, 8 (2017), 1496–1508.

[4] Huang, S. Towards multicellular biological deep neural nets based on transcriptional regulation. *arXiv preprint arXiv:1912.11423* (2019).

[5] Jang, S. S., Oishi, K. T., Egbert, R. G., and Klavins, E. Specification and simulation of synthetic multicelled behaviors. *ACS synthetic biology 1*, 8 (2012), 365–374.

[6] Li, X., Rizik, L., Kravchik, V., Khoury, M., Korin, N., and Daniel, R. Synthetic neural-like computing in microbial consortia for pattern recognition. *Nature communications 12*, 1 (2021), 1–12.

[7] López-Igual, R., Bernal-Bayard, J., Rodríguez-Patón, A., Ghigo, J.-M., and Mazel, D. Engineered toxin–intein antimicrobials can selectively target and kill antibiotic-resistant bacteria in mixed populations. *Nature Biotechnology 37*, 7 (2019), 755–760.

[8] Mirjalili, S., Mirjalili, S. M., and Yang, X.-S. Binary bat algorithm. *Neural Computing and Applications 25*, 3 (2014), 663–681.

[9] Moškon, M., and Mraz, M. Programmable evolution of computing circuits in cellular populations. *Neural Computing and Applications* (2022), 1–13.

[10] Ortiz, Y., Carrión, J., Lahoz-Beltrá, R., and Gutiérrez, M. A framework for implementing metaheuristic algorithms using intercellular communication. *Frontiers in Bioengineering and Biotechnology 9* (2021).

[11] Sarkar, K., Bonnerjee, D., and Bagh, S. A single layer artificial neural network with engineered bacteria. *arXiv preprint arXiv:2001.00792* (2020).

[12] Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

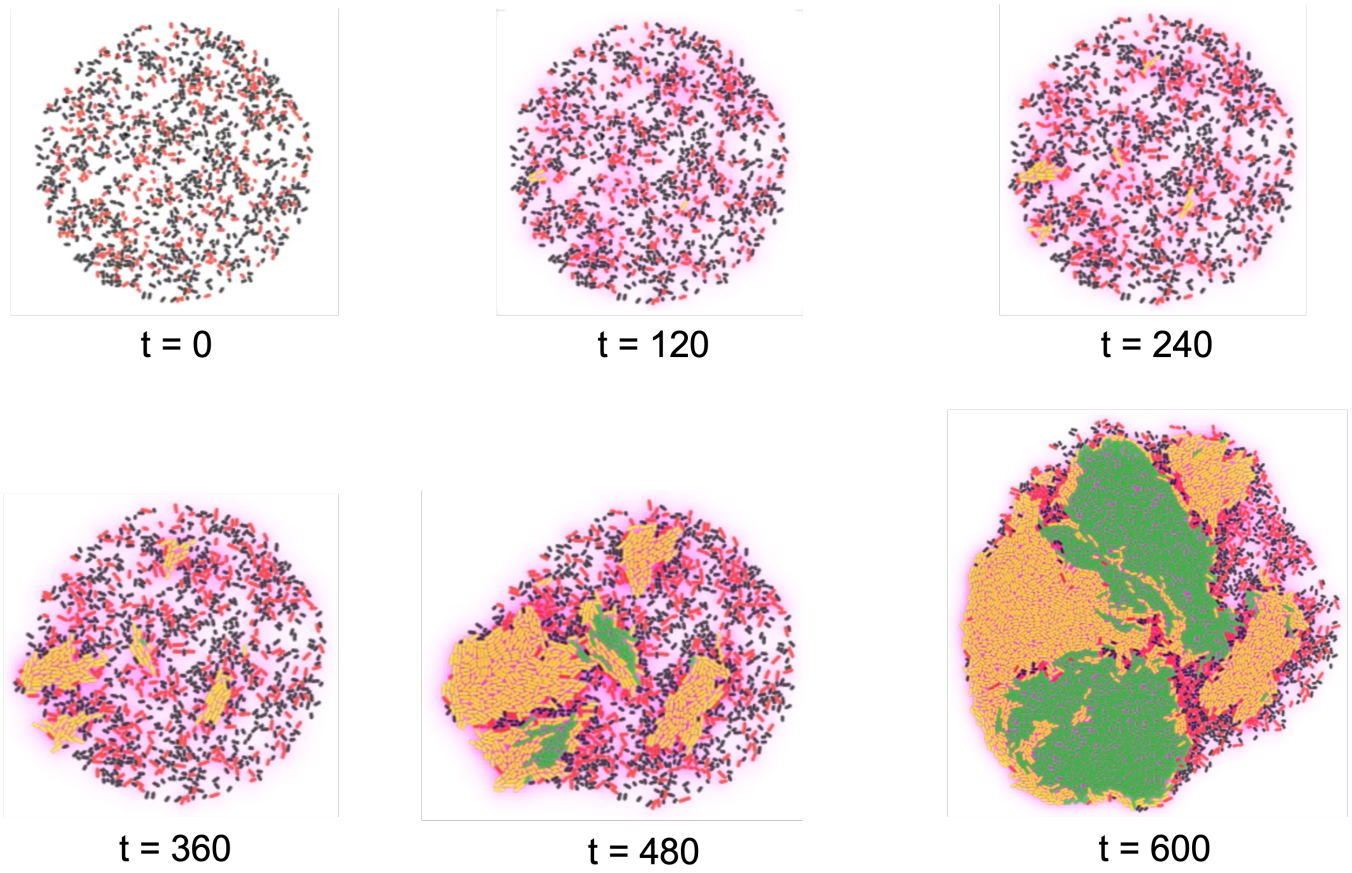


Figure 1: Visual representation of the simulation of an instance of the unicost binary set covering problem in gro. Black cells carry no plasmids, cells expressing RFP carry a single plasmid, cells with YFP have two plasmids and cells expressing GFP contain the optimal solution: all three plasmids (this can be related to the original fitness function through $\sum_{i=1}^{n} P_i$, where $P_i \in \{0, 1\}$: 0 represents absence of plasmid P_i, and 1 represents its presence. Also, for this case $i \in \{1, 3\}$, as we evaluate presence of all plasmids). The purple signal is QS, which represents the echolocation in the algorithm mapping. Parameters for this run were the following: conjugation rate 0.2 conjugations per life cycle (this accounts for the bat change in location related to the original algorithm) and a range of $[0.0001, 0.16]$ QS signal units per second (this relates to the echolocation frequency value of the original version of the algorithm). The first optimal solution is obtained at around t = 360. It should also be noted that as solutions get better in fitness, its division time decreases, leading to better exploitation and marking a spatial zone of good solutions.

DBTL bioengineering cycle: developing a population oscillator

Andrés Arboleda-García[1,*], Iván Alarcon-Ruiz[2], Yadira Boada[1], Jesús Picó[1], Eloisa Jantus-Lewintre[3]
[1]Synthetic Biology and Biosystems Control Lab, Instituto de Automática e Informática Industrial, Universitat Politécnica de València (Valencia, Spain). [2]CNIC - Spanish National Center for Cardiovascular Research (Madrid, Spain). [3]Molecular Oncology Laboratory, Fundación Investigación Hospital General Universitario de Valencia (Valencia, Spain).
maarbgar@upv.es

1 BACKGROUND

Synthetic biology aims at the targeted design or redesign and construction of new biological and bio-based parts, devices, and systems to perform desired functions [5, 6]. Not only is going up in this hierarchy (DNA, part, device, and system) the final objective of synthetic biology but also its main challenge [1]. To successfully accomplish it, engineering principles and methodologies are to be used. The engineering design-build-test-learn (DBTL) cycle is the common paradigm used in any engineering discipline where the design is made from the bottom by combining basic biological parts into devices and these into systems [2]. Essential for the success of this inherently modular approach of bottom-up synthetic biology is the need of starting from well-characterized parts [3]. A nested approach, consisting of a small DBTL cycle inside a larger one is an interesting option to tackle the lack of characterization some of the involved biological parts have. Here, we perform a nested DBTL cycle for the development of a bacterial population oscillator.

2 DESIGN

The design proposed to realize the population oscillator is a lysis circuit based genetic circuit based on [4]. The lysis circuit model was designed employing quorum sensing properties; the LuxR-AHL transcription factor activates when sufficient amounts of cells in the population produce AHL, which activates the expression of GFP and the lysis protein PhiX174E as it can be seen in Figure 1.

Previous to the construction of the device in the laboratory, a mathematical model was made. To create the model, it is necessary to determine the components of the system and how they are correlated. First, it is necessary to represent the system's different components, products, and interactions in the system. Afterward, the reactions of the system and corresponding stoichiometry were written, and applying the law of mass action; the ODEs were obtained. The Ordinal Differential Equations (ODE) model confirmed that the design was able to produce the desired oscillation. This model has seven states, N is the number of cells growing in the culture, and GFP is the system output. Ae is the number of molecules in the culture medium, and the remaining species are intracellular ones. The computational simulation with the model confirms that the designed gene circuit shows an oscillatory behavior since cells die when the lysis protein is activated.

$$
\begin{aligned}
\frac{d[N]}{dt} &= \mu N(N_0 - N) - \frac{\mathrm{k}L^n}{\mathrm{L_0}^n + L^n}N \\
\frac{d[A_e]}{dt} &= D\left(-N\mathrm{V_c}[A_e] + N[A]\right) + \frac{\mathrm{k}L^n}{\mathrm{L_0}^n + L^n}N - \mathrm{d_{Ae}}[A_e] \\
\frac{d[A]}{dt} &= D\left(\mathrm{V_c}[A_e] - [A]\right) + \mathrm{k_A}[I] - \mathrm{d_A}[A] - \mu[A] \\
\frac{d[R]}{dt} &= \frac{\mathrm{C_R\,k_R\,p_R}}{\mathrm{dm_R} + \mu} - \mathrm{d_R}[R] - \mu[R] \\
\frac{d[I]}{dt} &= \frac{\mathrm{C_I\,k_I\,p_I}}{\mathrm{dm_I} + \mu} - \mathrm{d_I}[I] - \mu[I] \\
\frac{d[L]}{dt} &= \frac{\mathrm{C_L\,k_L\,p_L}}{\mathrm{dm_L} + \mu}\alpha + \frac{\beta\frac{[R][A]}{A_0}^4}{1 + \frac{[R][A]}{A_0}^4} - \mathrm{d_L}[L] - \mu[L] \\
\frac{d[Ve]}{dt} &= \mathrm{C_{ve}\,k_{ve}} - \mathrm{dm_{ve}}[Ve] - \mu[Ve]
\end{aligned}
\tag{1}
$$

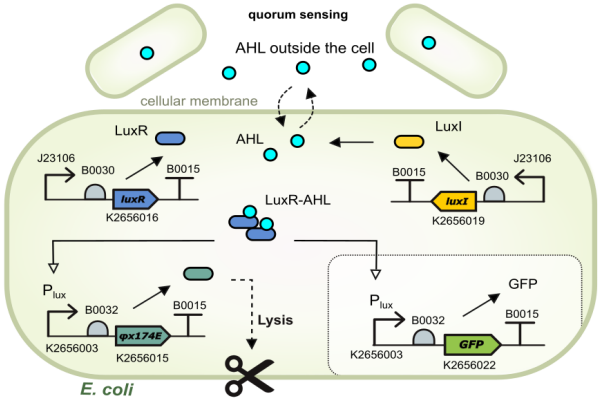


Figure 1: Schematic of the genetic circuit.

Using this model and a preliminary set of parameters, we perform computational experiments, i.e. simulation, to determine the behaviour of our genetically engineered bacteria

carrying the proposed genetic circuit. The results, shown in Figure 2, clearly demonstrate the circuit is able to oscillate, giving us sufficient knowledge to pass to the next step of the DBTL cycle.

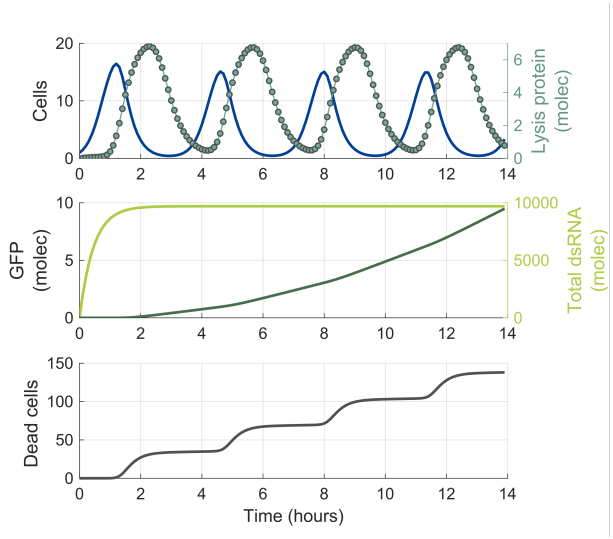


Figure 2: Simulations from the Design stage of the DBTL

3 BUILD

Using the information from the Design stage, we start our Build stage. The engineered Escherichia coli DH5 carrying the plasmid pARKA1-O1 was used in this work. This plasmid, containing the entire genetic circuit to give E. coli the oscillatory population behaviour, was constructed using a Golden Braid assembly method with three levels to build our genetic circuit. First, combining BBa_K2656122, a Level 1 transcriptional unit (TU) expressing GFP, and BBa_K2656114, a Level 1 transcriptional unit expressing luxR gene, we created part BBa_K3893028, a level 2 TU. Afterwards we combined BBa_K3893026, a Level 1 TU expressing luxI gene, with BBa_K3893027, a Level 1 TU expressing PhiX174E, to make BBa_K3893029, a level 2 TU. Finally, combining the two Level 2 TU we constructed BBa_K3893030, our genetic circuit as shown in Figure 3.

4 TEST

We designed a temporal experiment to measure absorbance and fluorescence data from the gene circuit to test the cell lysis. The absorbance and fluorescence data from the genetic circuit collected from laboratory experiments show how the bacteria cell population decrease after the population reaches a threshold that activates pLux promoter to express GFP and the lysis protein.

Figures 4A and 4B show cells when the lysis protein is activated by the lux promoter. The number of cells was calibrated using standardized particle units from Engineering Committee (Measurement Committee) and iGEM Interlab study 2018-2019.

The total GFP fluorescence expressed by the population and a single-cell is shown in Figures 4C and 4D, respectively. After cell lysis, the molecules of GFP are released into the surroundings, increasing the concentration shown in the figure. We used MEFL/Particle (molecules of equivalent fluorescein per particle) as a standardized unit to quantify GFP expression per cell.

5 LEARN

Using the experimental data obtained in the Test stage and in order to characterize the pLux promoter, we first did a small DBTL cycle inside the larger one, a nested DBTL cycle.

Inner DBTL Cycle

First, we (D)esign and (B)uilt a genetic circuit composed of BBa_K26561224 expressing GFP and BBa_K2656114, constitutively expressing luxR gene (Figure 5). This circuit was (T)est with different AHL inductions, and the results allow us to (L)earn and adjust our model (Figure 6).

Using the information acquired from the experimental data in the (T)est stage and with the things we (L)earned from the nested DBTL cycle, we adjusted the model of our system and performed new computational simulations. By comparing the in silico vs in vivo results, our model captures the temporal dynamics of the oscillator. The newly adjusted ODE model can be used to redesign the device by predicting outcomes in silico instead of building a new set of circuits to see which one performs better than the original.

6 FUTURE APLICATIONS

The oscillation pattern of the system and its versatility can have applications like localized drug delivery, the design of synthetic microbial communities to understand biofilm formation, chemical production from biomass, or the prevention of biofilm formation by other bacteria.

7 ACKNOWLEDGMENTS

This research was funded by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 grant number PID2020-117271RB-C21 and GVA grant CIAICO/ 2021/159. Y.B. thanks Grant PAID-10-21 Acceso al Sistema Español de Ciencia e Innovación-Universitat Politècnica de València. Y.B. also thanks to Secretaría de Educación Superior, Ciencia, Tecnología e Innovación of Ecuador (Scholarship Convocatoria Abierta 2011). A.A.G. thanks Grant PAID-01-21 Programa de Ayudas de Investigación y Desarrollo - Universitat Politècnica de València.

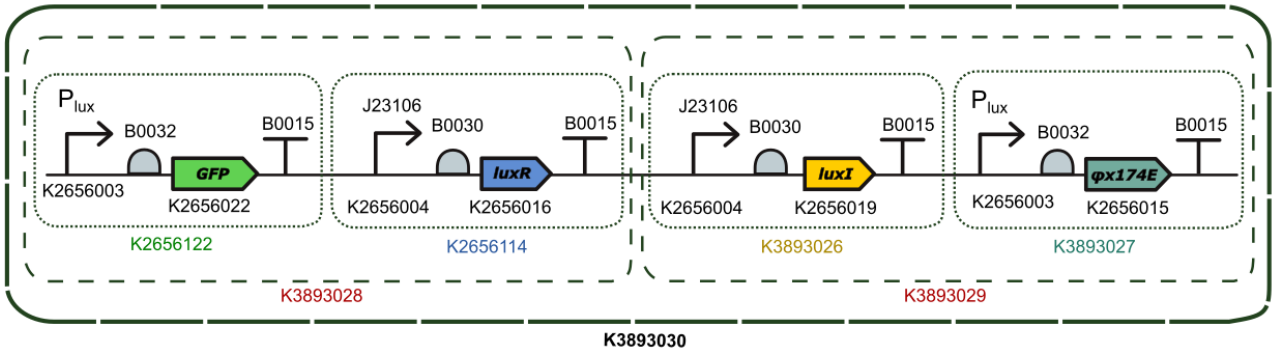


Figure 3: Schematic in visualSBOL of the constructed genetic circuit showing the different assembly levels.

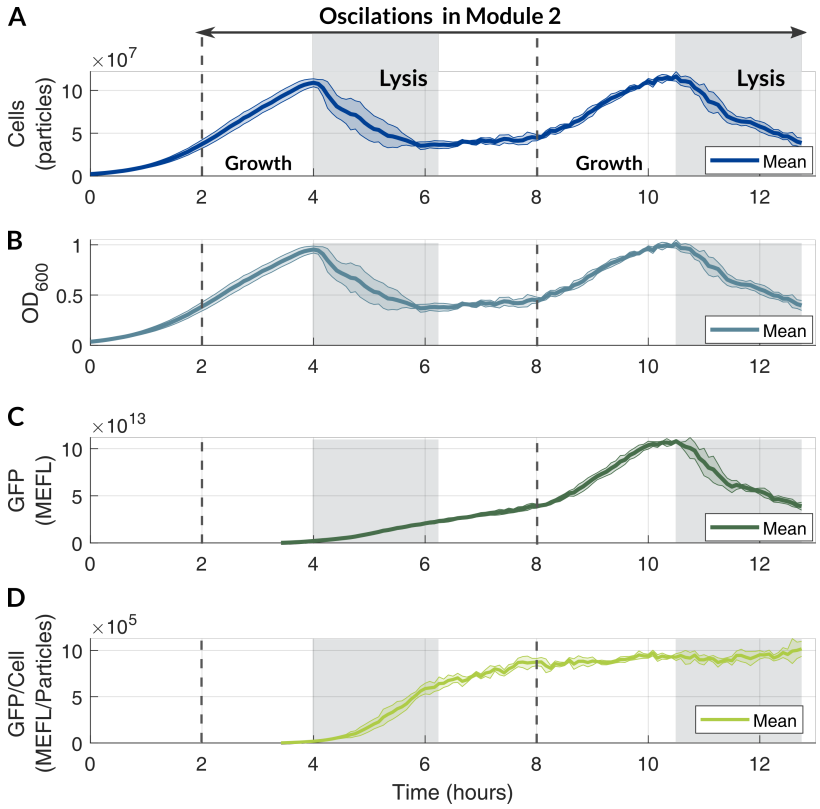


Figure 4: Experimental data from the Test stage of the DBTL

REFERENCES

[1] Beal, J., Haddock-Angelli, T., Farny, N., and Rettberg, R. Time to get serious about measurement in synthetic biology. *Trends in biotechnology 36*, 9 (2018), 869–871.

[2] Canton, B., Labno, A., and Endy, D. Refinement and standardization of synthetic biological parts and devices. *Nature biotechnology 26*, 7 (2008), 787–793.

[3] Church, G. M., Elowitz, M. B., Smolke, C. D., Voigt, C. A., and Weiss, R. Realizing the potential of synthetic biology. *Nature Reviews. Molecular Cell Biology 15*, 4 (2014), 289–294.

[4] Din, M. O., Danino, T., Prindle, A., Skalak, M., Selimkhanov, J., Allen, K., Julio, E., Atolia, E., Tsimring, L. S., Bhatia, S. N., et al. Synchronized cycles of bacterial lysis for in vivo delivery. *Nature 536*, 7614 (2016), 81–85.

[5] ERASynBio. Next steps for european synthetic biology: a strategic vision from erasynbio. Report, ERASynBio, 2014.

[6] Kojima, R., and Fussenegger, M. Synthetic biology: Engineering mammalian cells to control cell-to-cell communication at will. *ChemBioChem 20*, 8 (2019), 994–1002.

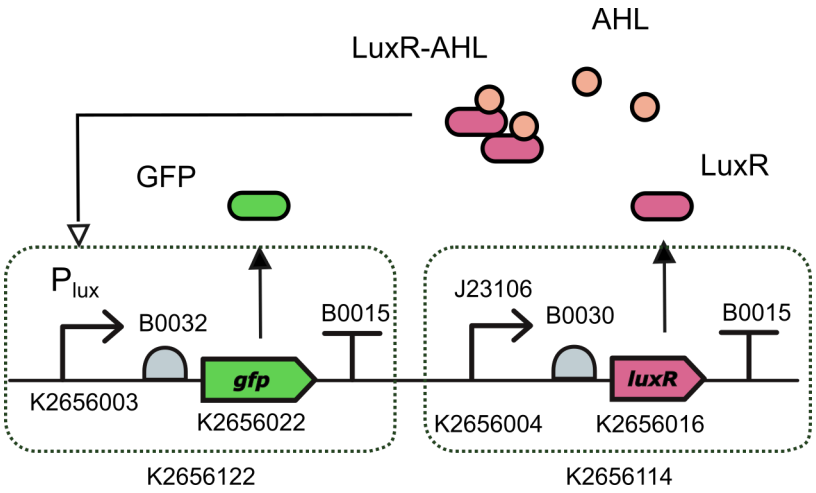


Figure 5: Circuit build for the nested DBTL cycle: AHL induced expression of GFP protein and luxR constitutive expression.

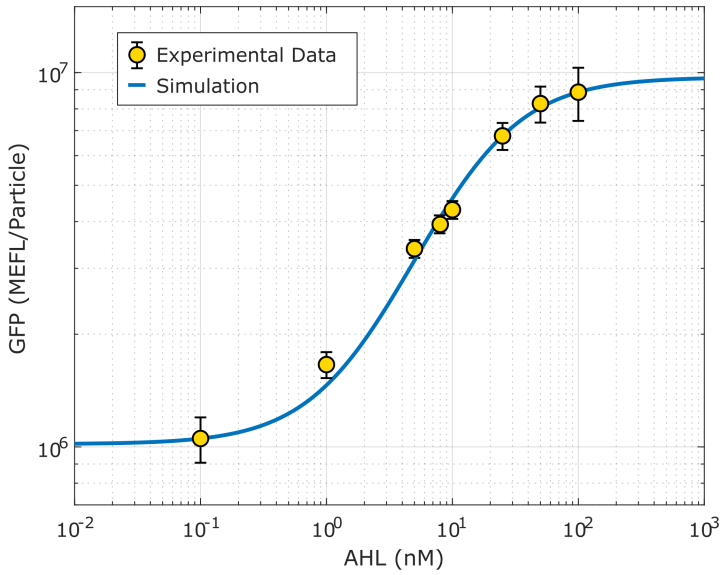


Figure 6: Learn stage within the nested DBTL cycle. Model and experimental data of the circuit construted.

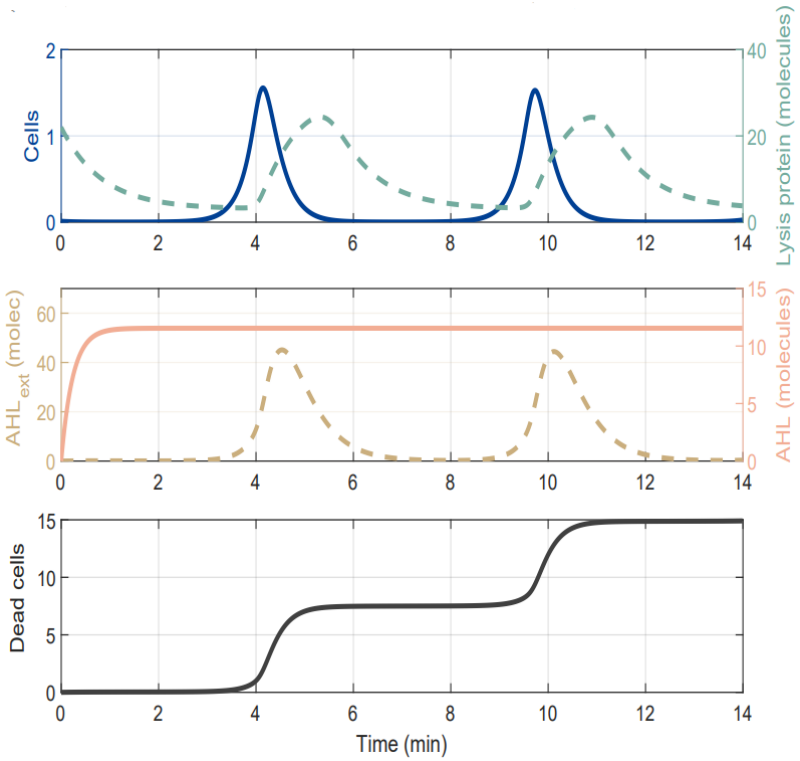


Figure 7: Learn stage of the DBTL. Incorporating informationfrom both principal and nested DBTL cycles we achieve a better representation of the experimental results with our model

Dynamic Behavior Alters Influences and Sensitivities in Biological Networks

Gaoxiang Zhou
University of Pittsburgh
Pittsburgh, United States
gaz11@pitt.edu

Natasa Miskov-Zivanov
University of Pittsburgh
Pittsburgh, United States
nmzivanov@pitt.edu

1 INTRODUCTION

Networks have emerged from many biological studies as researchers try to capture correlations and causation between different components of studied systems. As a graph representation, visually depicting the entities (often as nodes) and their connections (often as edges), network can provide a systematic view for researchers and facilitate the scientific process of query, explanation, hypothesis, and verification. One typical task when dealing with biological networks is to detect among all molecules those that are "influencers", narrowing the analysis scope and allowing for modularity and abstraction. The sensitivity analysis of biological networks aims to fulfill this task and identify and explain the influences of internal or external changes.

Currently the identification of influential nodes in complex networks largely relies on the network topology. Topological metrics of influence study range from simple ones like in-/out-degree, to more elaborate attributes like centralities [1]. The main shortcoming of analyzing influence of biological networks via topological attributes is that they fail to capture influences associated with semantic meaning assigned to nodes, in other words, the rules and dynamics of state change are ignored.

Network physicists have recently begun developing new influence metrics to accommodate networks with enriched knowledge representations, such as percolation centrality [2]. However, these metrics still fail to fully leverage the benefit of the abstracted update rules of state change. The rules which govern the dynamics of biological networks via mathematical equations can provide detailed insights since they are synthesized using knowledge from literature, data mining and expertise.

As of now, there have been few discussions about developing a node influence metric that not only takes network topology and dynamics into account, but also exploits the closed-form rules of state change. In this work, we propose a new metric assigned to edges and develop a unified framework to study node influence. Though rooted in simple partial derivative quantifying the influence of function to change of one variable, our proposed metric is enriched and modified to fit discrete state representations and account for the state biases resulting from network dynamics. When analyzing influences across remote connected elements, our metric also explores the advantages of topology-based techniques, by summing the influence propagation across all possible paths. Given that the assembly of many biological networks relies on prior information and evolves in a self-learning manner, the design of our influence metric accommodates this feature via its dependence on joint state distributions. Our metric is applicable to networks without or with prior information of initial state, and extendable upon network modification as a result of new knowledge.

Although our framework is applicable to discrete models with different function types, we focus here on Boolean functions. Previous research on Boolean difference-based influence analysis explored several directions. The authors in [3] only studied "local" sensitivity of a model element and the influence of its regulators on this element, while authors in [4] focused on the "global" long-run sensitivity of how likely a mutation is to change the converging attractor of a biological network. In addition, prior analysis has been mainly applied on two types of biological networks, i.e., the probabilistic Boolean networks which are usually inferred from gene expression profiles, and random Boolean networks. In this work, we conduct an influence and sensitivity study on Boolean networks that were created to capture known influence mechanisms, with applications in both modeling and design of biological circuits.

Here, we first define a quantity to measure immediate influence between directly connected elements, then we extend it to remote influences between indirect ones. Given the quantity defined under certain distributions, we analyze it under uniform versus scenario-dependent distributions of joint element states. We implemented two

methods for computing influences, namely, the function-based and hybrid-based methods, both of which have been rigorously tested in a human-machine-interface platform designed for open access. Overall, these different types of analysis constitute a universal framework to parse any rule-based model and systematically and quantitatively evaluate causal influences between elements. Using our framework, in a model of naïve T cell differentiation [5] we identified the most influential elements and investigated how the difference between uniform and scenario-dependent analysis can affect our sensitivity results.

2 METHODOLOGY

We use an element rule-based modeling approach, which allows for simulation of state transitions, feedback loops, integration of both prior knowledge and data, as well as analysis of large hybrid networks that include protein-protein interactions, gene regulation and metabolic pathways. All components and interactions are represented as a directed graph $G(V, E)$, with a set of nodes $V = \{x_1, \dots, x_N\}$ where N is the number of elements in the model, and a set of directed edges E denote regulatory interactions between elements. We also define the number of discrete values representing different levels of the element's activity, n_i, such that $x_i \in X_i: \{0,1 \dots, n_i - 1\}$. By assigning values to all elements in the model, we obtain the model state, a vector $\boldsymbol{v} = (x_1, \dots, x_N)$. We assign a state transition function to model elements, defining a state change of the element, given the states of its regulators. We refer to these functions as element update rules and to the model with update rules as an executable model $M(V, F)$, with functions $F = \{f_1, \dots, f_N\}$ for $V = \{x_1, \dots, x_N\}$, respectively. In the case of Boolean variables representing element states, the basic operations are AND, OR and NOT.

The *immediate influence* of element x_i on function $x_j = f_j(x_1, \dots, x_N)$ where $i, j \in \{1, \dots, N\}$, α_i^j, is defined:

$$\frac{\partial f_j}{\partial x_i} = \left(f_j|_{x_i=0}\right) \oplus \left(f_j|_{x_i=1}\right) = f_{ji}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N)$$

$$\alpha_i^j = E\left(\frac{\partial f_j}{\partial x_i}\right)$$

Given that α_i^j is defined as expectation under certain distribution, we refer to this analysis as *uniform* when assuming uniform distributions among all possible states, and as *scenario-dependent* when distributions are estimated from simulation context.

If there is a regulatory directed pathway between a source node x_s and a target node x_t, we assign pathway score to the $path: \{x_s, x_{p_1}, \dots, x_{p_m}, x_t\}$:

$$s(path) = \exp\left[-\left(\sum_{i=0}^{m} w_{p_i p_{i+1}}\right)\right], \qquad w_{ij} = -log(\alpha_i^j)$$

The *remote influence* of element x_s on element x_t, r_s^t, the *remote sensitivity* of element x_t, $sensi_t$ and the *remote impact* of element x_s, $impt_s$ are defined as

$$r_s^t = \sum_p s(p), \quad sensi_t = \sum_s r_s^t, \quad impt_s = \sum_t r_s^t$$

3 RESULTS AND DISCUSSIONS

As our case study, we use the model of the circuitry that controls differentiation of naïve T cells into regulatory (Treg) and helper (Th) cell phenotypes presented in [5] (the studied network circuitry is also shown in [5]). Previous research has shown that these two types of T-cells have different functions and can be distinguished by expressions of several key markers. For example, in the Treg type, the transcription factor forkhead box P3 (Foxp3) is expressed, and Interleukin-2 (IL-2) is inhibited, while in the Th type, Foxp3 is inhibited, and IL-2 is activated. Besides the list of elements, and their update functions, we also defined scenarios for conducting scenario-dependent analysis. We explored three scenarios: (1) high antigen dose (TCR=2), (2) low antigen dose (TCR=1), and (3) toggle (TCR=2 initially and changed to TCR=0 at a defined time step).

Figure 1(A) shows remote sensitivity versus impact gains of all elements under the three scenario-dependent analyses, compared to uniform analysis. The majority point distribution in first quadrant suggests both sensitivity and impact increase for most elements, while the scenario of high dose reveals an even higher growth. Figure 1(B) contrasts the changes across three scenarios and highlights elements with significant variations. For example, elements like AP1 and JAK3 exhibit more sensitivity in high dose scenario, but less sensitivity in low dose. These results emphasize the importance of including dynamic conditions when exploring sensitivities and impacts in both natural and synthetic biological networks, especially in the context of automation when generating explanations or designing biological circuits and interventions.

REFERENCES

[1] Chen, D.B., et al., Identifying influential nodes in complex networks. Physical Statistical Mechanics and Its Applications

[2] Piraveenan, M., M. Prokopenko, and L. Hossain, Percolation centrality: quantifying graph-theoretic impact of nodes during percolation in networks. PLoS One, 2013. 8(1): p. e53095.

[3] Shmulevich, I. and S. Kauffman, Activities, and sensitivities in Boolean network models. Physical Review Letters, 2004. 93(4).

[4] Qian, X., and E. Dougherty, On the long-run sensitivity of probabilistic Boolean networks. Journal of Theoretical Biology, 2009. 257(4): p. 560-577.

[5] Miskov-Zivanov, N., et al., The Duration of T Cell Stimulation Is a Critical Determinant of Cell Fate and Plasticity. Science Signaling, 2013. 6(300).

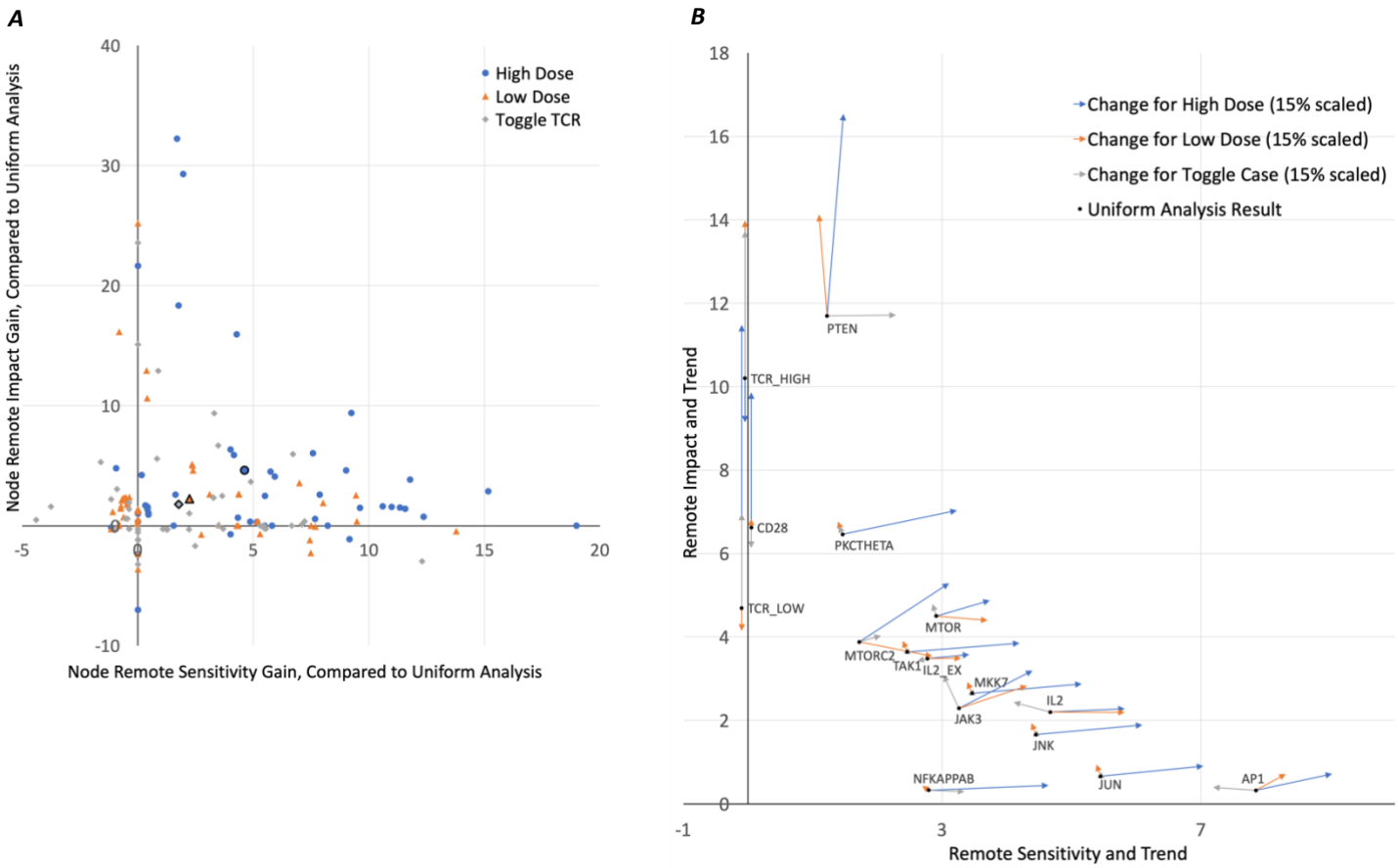


Figure 1: (A): the remote sensitivity gains versus remote impact gains (i.e., scenario-dependent analysis minus uniform analysis) of 52 nodes in model [5] across three studied scenarios; the highlighted scatter points (larger marker size with black border) of each scenario denote the average gain in that scenario. (B): the sensitivity analysis results of 16 elements whose remote sensitivities and impacts are significantly altered across scenarios, their uniform analysis results are shown as stationary points, while arrows of different styles correspond to change trends of different scenarios; Arrows are all 15% scaled to avoid graph overlap but directions are preserved; Element TCR_HIGH, TCR_LOW, CD28 are intentionally moved off the y-axis slightly for clear plots.

Spatially Solving the Graph Coloring Problem Using Intercell Communication

Daniela Moreno, Diego Araya, Martín Gutiérrez
Universidad Diego Portales
Santiago de Chile, Chile
{daniela.moreno1,diego.araya1,martin.gutierrez}@mail.udp.cl

1 INTRODUCTION

NP-Hard problems are a class of computationally complex problems. This class of problems cannot be solved in reasonable time, making it difficult to attempt their solution. As early as 1994, alternative approaches based on DNA computing for tackling such problems were reported [1, 5]. Later, a different approach using protein filament networks was also reported [6]. The key property of such solutions lies in the large parallelism that biology-based solutions have to offer. However, to the best of the authors' knowledge, no solution including a spatial resolution has been proposed. This work shows a spatial-resolution based solution for Graph 3-Coloring problem simulated in `gro` [3, 4].

2 GRAPH 3-COLORING PROBLEM AND ITS MODEL IN GRO

The Graph Coloring problem refers to the assignation of colors to all nodes in any graph so that no neighbor nodes have the same color assignation. This problem is NP-complete starting from 3 colors (Graph 2-Coloring is in P). A graph consists both of nodes to be colored and edges that define the links between nodes, making them neighbors. In a `gro` implementation, a cell colony is a landscape for a graph in which a Quorum Sensing (QS) signal limits a "node zone", which represents the traditional node as a zone inside the colony that is marked and identified based on band detection circuits [2, 3], and assigned a certain color. Within this zone, a leader cell is a single one that selects the color to be assigned to the node zone and is responsible for limiting the zone area. Follower cells are the ones expressing the color selected by a leader and compose the larger part of the node zone. Edges are not visible in the colony, but instead they are denoted by borders between each zone that defines a node. Thus, the expected representation of a correct spatial solution is to have color zones, separated by borders, that are never adjacent to a different zone with the same color.

3 PROBLEM ELEMENTS AND SOLUTION

The solution for the Graph 3-Coloring problem usually works by steps, which is a problem in bacterial colonies, since bacteria grow and evolve in parallel. This setback was tackled by adding delays in activation times, achieved by sending two QS signals - a first one that selects the color for the zone, and a second one that enables a CRISPR-Cas9 system to eliminate part of the circuit and prohibit the selection of the same color. Thus, only the cells that are closer will be able to choose the color indicated by the first signal and a node zone will be formed. Beyond the node zone, this will force the selection of different colors for adjacent regions of the colony (also in a limited zone, applying the same logic as previously). The global circuit design is shown in Figure 2.

Circuit operation is started by two bacteria -initiators- that establish the initial interaction between two different color bands. Once these two regions are adjacent and touch each other, color selection for adjacent nodes to these two zones is executed. As explained previously, difference in activation times will result in some bacteria selecting a color for the zone before others. Once the zone has selected and expressed its color, the second signal is sent to avoid having adjacent node zones selecting the same color. The delays in activation times is crucial here, since they allow each node's color to be established correctly, preventing the activation of multiple coloring circuits creating a mix.

The result is a greedy algorithm in which each node zone converges to a color, and never changes its selection. Upon generation of the new node zone color, it will establish a signal range. When two of these node zone ranges intersect, the selection of a new leader and therefore color selection for the adjacent node zone will be initiated. This step is repeated until all node zones are assigned a color.

This approach allows each cell inside a colony to determine -through bacterial communication- the optimal color output. This is, how it contributes in the formation of the graph while coloring it according to restrictions that dynamically create a graph with nodes of the same color apart from each other. Our solution was designed to color node zones and then select a leader to continue adding a new node zone to the colony without relating to a specific graph and only considering the cell colony as a coloring landscape. However, to encode a graph into a cell colony, spatial resolution is needed and the following rules determine a mapping to a given graph: 1) A leader cell accounts for a node zone, which represents a node in a given graph. 2) The QS signal and its range complete the node zone definition, but also configure

the edges of the graph. 3) Any non-adjacent nodes should be placed furthest apart, but taking into account existing node connections in the graph.

4 TEST RESULTS

Several implementations of the algorithm were tested, and different results obtained. A first variant of the algorithm was designed to activate when a global environment signal is present. Upon activation, the circuit allows for selection of one or more colors for each bacterium, leading to an indeterminate color value. A picture of the execution of this version is shown in Figure 1a.

An improvement over this implementation leads to more specific constraints on when to assign a color, and band intersection zones restrict the amount of colors in such zones to one. The effect of these constraints is that now single color zones are well differentiated and set apart, although zone boundaries are mostly unlimited and depending on their location and surroundings. An example of the second implementation is shown in 1b.

The next implementation enforces all constraints established in the second design, and establishes node zones by restricting cell growth. Also, color is fixed as the design uses CRISPR-Cas9 to eliminate other possible colors and commit to the selected color choice. The sequence of solution calculation for this implementation is: 1) The circuit initiates by instantiating two leaders that are assigned different colors. 2) A third color is added to a different leader adjacent to the previously selected ones. 3) Leaders keep being chosen and assigned a color of which they do not receive signals, completing the colony coloring. A depiction of the result of this algorithm implementation is shown in Figure 1c.

Communication between cells was improved to allow growth inside the colony while trying to maintain the solution acceptable, giving rise to new implementations of the algorithm. The results for these implementations are represented by the nodes within the graph expanding unevenly, losing their shape and usually forming long lines of cells, which makes the edges between nodes harder to recognize. An sample execution is shown in Figure 1d. This leads to two main problems inside the node representation: the first one is that bacteria are split into chains, creating the previously mentioned lines that expand outwards of the formerly defined node zone, thus reconstructing the node into a fringe shape; the second reason is the growth rate. If it is higher than the speed at which the colony is colored, it is locked into a stationary state. This is due to the nodes expanding faster than the rate at which the signal calling for a new color node is transmitted, so the farthest bacteria never receive the signal.

5 FINAL REMARKS AND FURTHER WORK

The proposed algorithm is based on cell-cell communication and CRISPR-Cas9 system to color node zones in a pattern such that it can solve the Graph 3-Coloring NP-Complete problem. Further investigation is required to assess whether this technique can be extended to Graph N-Coloring problems where N > 3. One limitation of our solution is given by the availability of QS signals and potential crosstalk that could occur between them. However, studies on orthogonality of QS signals was done in [7] and establishes a good starting point. The initial designs account for several patch patterns that adjust to multiple graph morphologies and therefore successfully color them according to an appropriate solution to the Graph 3-Coloring problem. Our solution currently handles planar map graphs, as `gro` is a 2D simulator. More research is needed to verify whether this proposal extends to 3D situations.

In our simulations, the coloring of the graph within the colony cannot be predicted from the beginning, although by analyzing the simulations step by step, it is possible to assume what the color assignation in the next step will be. This is because the definition of the coloring process in the colony is conditioned by its environment and the bacterial communication happening within the colony (graph). This means that if we take the colony as a whole, its execution for solving the coloring problem is non-deterministic. Each cell is intended to exhibit a determined output depending on its own environmental state, which is possible thanks to QS, and the set of plasmids that encode their logic.

REFERENCES

[1] Adleman, L. M. Molecular computation of solutions to combinatorial problems. *science 266*, 5187 (1994), 1021–1024.

[2] Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., and Weiss, R. A synthetic multicellular system for programmed pattern formation. *Nature 434*, 7037 (2005), 1130–1134.

[3] Gutiérrez, M., Gregorio-Godoy, P., Perez del Pulgar, G., Muñoz, L. E., Sáez, S., and Rodríguez-Patón, A. A new improved and extended version of the multicell bacterial simulator gro. *ACS synthetic biology 6*, 8 (2017), 1496–1508.

[4] Jang, S. S., Oishi, K. T., Egbert, R. G., and Klavins, E. Specification and simulation of synthetic multicelled behaviors. *ACS synthetic biology 1*, 8 (2012), 365–374.

[5] Lipton, R. J. Dna solution of hard computational problems. *Science 268*, 5210 (1995), 542–545.

[6] Nicolau, D. V., Lard, M., Korten, T., van Delft, F. C. M. J. M., Persson, M., Bengtsson, E., Månsson, A., Diez, S., Linke, H., and Nicolau, D. V. Parallel computation with molecular-motor-propelled agents in nanofabricated networks. *Proceedings of the National Academy of Sciences 113*, 10 (2016), 2591–2596.

[7] Scott, S. R., and Hasty, J. Quorum sensing communication modules for microbial consortia. *ACS synthetic biology 5*, 9 (2016), 969–977.

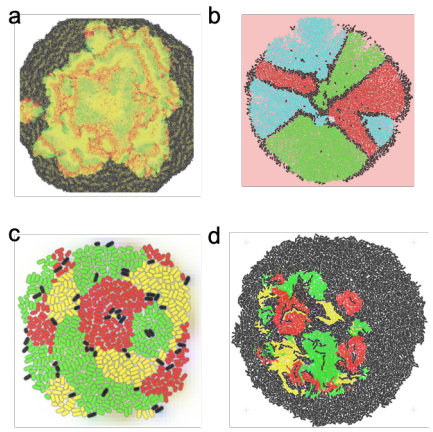


Figure 1: All tested versions of the circuit: a. First design of the solution: the oscillation of colors leads to a vague solution in which node zones can be set apart, but no clear boundaries are found. b. Color constraints are enforced in the second design. Single color well differentiated zones are obtained. c. Static version of the colony (without growth) in which well defined color patches are placed and describe a correct solution for the Graph-3-Coloring problem. d. While the calculated solution is still correct, sometimes gaps and shape deformities are introduced when the algorithm takes growth into account.

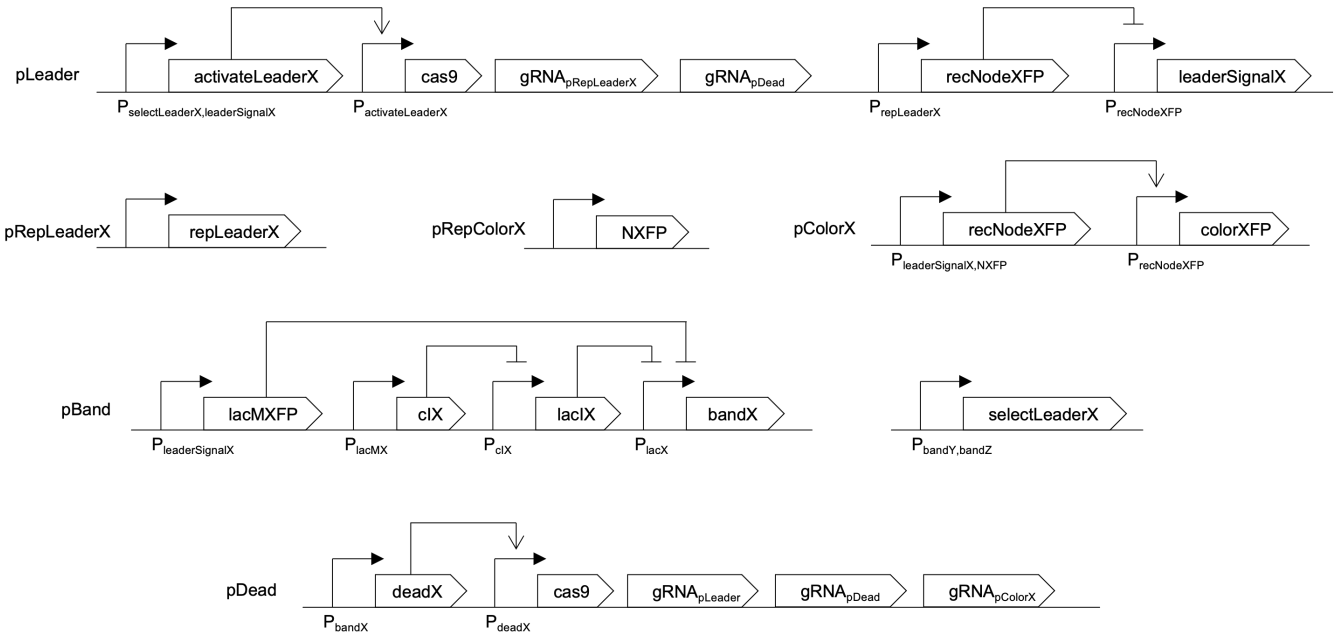


Figure 2: Global circuit design. The circuits presented are the same for each color, where 'X' refers to the color itself, while 'Y' and 'Z' are referring to the two remaining colors. The algorithm implementation starts by selecting leaders, which is correlated to the 'pLeader' plasmid (that activates a leader), and therefore, it removes both the plasmid that allows the cell to express a given color and the one who represses the leader signal, thus allowing the transmission of this signal. The cells that are closer to the leader will receive this signal and activate the 'pBand' plasmid matching the specified color, causing color expression while also activating the 'pDead' plasmid. It removes the circuits associated with the possibility of becoming a leader, the color repressor and 'pDead' itself, thus keeping the cells expressing the color protein constantly. Finally, through the activation of the 'pBand' plasmids corresponding to colors 'Y' and 'Z', it is possible to determine when there is an intersection between the bands and then a new leader for the 'X' color is selected.

Exploring Advantages and Limitations of Discrete Modeling of Biological Network Motifs

Difei Tang
University of Pittsburgh
Pittsburgh, United States
dit18@pitt.edu

Gaoxiang Zhou
University of Pittsburgh
Pittsburgh, United States
gaz11@pitt.edu

Natasa Miskov-Zivanov
University of Pittsburgh
Pittsburgh, United States
nmzivanov@pitt.edu

1 INTRODUCTION

Computational modeling is an important part of studying complex systems and it plays a critical role in interpreting biological experiments and behaviors. It can also provide a guideline for biologists when designing time-efficient and low-cost experiments.

Several modeling approaches have been applied in biological system studies. Ordinary differential equations (ODEs) can be used to model biochemical reactions when kinetic parameters are available [1]. Also, reaction rule-based modeling allows for modeling interactions between different molecules in cells, and this modeling approach is usually specified in the BioNetGen language (BNGL) [2]. However, quantitative modeling approaches mentioned above might not be suitable when the systems are too large, since it is challenging to find all kinetic parameters for building the model. Additionally, when the collected data or information about the system is uncertain or missing, these methods will become impractical. Therefore, logical and discrete modeling was introduced to solve these problems. In logical models, only Boolean variables and logic state transition functions (i.e., element update rules) are needed. Interactions between model elements can be simply represented by a logical rule between regulators and the regulated element. However, some large systems may require more than two values to represent their states, thus making logical modeling less practical. To address this issue, the DiSH (Discrete, Stochastic, Heterogenous model simulation) simulator was proposed [3], extending the logical modeling to multiple levels with several new features and notations.

In this study, we mainly focus on the utility of discrete modeling introduced by DiSH simulator and compare the simulator outcomes with simple Boolean network and ODEs.

In DiSH simulator, discrete rules and score are applied to the model and its elements if they have multiple levels of value. These rules are different from normal logical rules. For example, N^{th}complement is used to represent logical NOT in the discrete domain where N is the maximum discrete value that a variable can take. AND and OR logical operators are replaced by maximum and minimum functions. Additionally, if the element has both activators and inhibitors, scores for activation and inhibition will be calculated using their logical function and if activator score is greater than inhibitor score, the next value of this element will increase by 1 compared to its current value. On the contrary, the value will decrease by 1 if the inhibitors take the higher score [4].

For comparison with other modeling methods, we translate the rule-based BNGL model to lists of ODEs by Copasi [5] and then convert these ODEs into a spreadsheet format, which is required by DiSH simulator as an input. These models created using different modeling approaches are then simulated using DiSH simulator and other tools.

2 METHODS

To evaluate the efficiency of discrete modeling in DiSH simulator and compare the outcomes with other modeling methods, we used the following as inputs: (1) A simple binding model (ABC) in BNGL using unstructured molecules to represent each species; (2) Basic disease outbreak model (SIR) used in 2020 BioNetGen Workshop Tutorial [6]; (3) A simple Ras pathway model (RAS) built from a diagram described in [7]. Simple network structures for these three models are shown in Figure 2 (left) using Cytoscape [8] to draw, and the positive regulation from source node to target node is indicated with arrow shape, while the negative regulation is the T shape edge.

We first import the BNGL model into SBML format and then use Copasi to extract the ODEs using its mathematics related functionality. Next, we create the discrete model based on the Jacobian matrix of the ODEs and this conversion consists of several steps: (1) Regulation type of element in ODE is determined by signs of the Jacobian elements and these elements are classified into positive and negative regulators. Also, the element itself is considered and assigned to one of these two classification groups. (2) The order of elements in ODE should remain the same in both positive and negative regulators. (3) If one term of ODE is formed by element multiplication, the logical function between them is defined as AND gate, while the addition of each term is defined as OR gate. (4) If the regulation of one element is represented as the denominator of a function and multiplied by other elements, we put this element in positive regulators and use NOT logic to represent it. (5) The initial values in ODE are rescaled to discrete level. For example, Figure 1 shows the ODEs created from the SIR BNGL model. Table 1 shows the corresponding discrete model in the format supported by DiSH simulator, "Positive" and "Negative" columns denote the positive and negative regulators of each element respectively, and the rescaled initial values are filled in the "initial" column.

$$\frac{dS}{dt} = -k_b SI$$

$$\frac{dI}{dt} = k_b SI - k_g I$$

$$\frac{dR}{dt} = k_g I$$

Figure 1: Equations of SIR model translated from BNGL model

Table 1: SIR model in the format supported by DiSH

Element	Positive	Negative	Initial
S		(S, I)	10
I	(S, I)	I	1
R	I		0

3 RESULTS

For the first two models, we use BioNetGen to run the simulation with a built-in ODE simulator and the simulation step is set to 200. The corresponding discrete models are simulated using DiSH simulator with random update scheme and we simulate them with 200 steps and 200 runs. For the Ras pathway model, we use Scipy python package [9] to run the ODE simulation and also the discrete version of this model is simulated by DiSH simulator. Additionally, we set the levels of discrete model as 11, which means the range of value is [0, 10]. Figure 2 shows a plot summary of simulation results. For the two simple BNGL models, we find that the trend of the trajectories of discrete modeling is almost the same as ODE. The third model simply illustrates the signal transition in Ras pathway from the activation of RTKs to the downstream elements, and due to the feedback effect, the value of RTKs, RAF and ERK is suppressed by its negative regulators. Discrete modeling for this pathway shows the similar feedback regulation, but the steady states of elements are different with the ODE approach, due to the complex kinetic parameters we used for ODE models.

4 CONCLUSIONS

The conversion between BNGL, ODE and discrete models that we describe here demonstrates that logical models can be enhanced into discrete models with similar functions, capable of predicting the biochemical network's dynamics. Our results show that for small models with simple reactions, but intertwined feedback loops, this modeling method can nearly reproduce the same dynamic behavior as differential equations. Our next step is to further investigate the combination of logical functions and explore new logical rules to optimize this discrete modeling method and test with larger networks.

REFERENCES

[1] Materi, W. and D.S. Wishart, Computational systems biology in drug discovery and development: methods and applications. Drug discovery today, 2007. 12(7-8): p. 295-303.

[2] Faeder, J.R., M.L. Blinov, and W.S. Hlavacek, Rule-based modeling of biochemical systems with BioNetGen, in Systems biology. 2009, Springer. p. 113-167.

[3] Sayed, K., et al. DiSH simulator: Capturing dynamics of cellular signaling with heterogeneous knowledge. in 2017 Winter Simulation Conference (WSC). 2017. IEEE.

[4] Telmer, C.A., et al. Dynamic system explanation: DySE, a framework that evolves to reason about complex systems-lessons learned. in Proceedings of the Conference on Artificial Intelligence for Data Discovery and Reuse. 2019.

[5] Hoops, S., et al., COPASI—a complex pathway simulator. Bioinformatics, 2006. 22(24): p. 3067-3074.

[6] https://mmbios.pitt.edu/workshops/2020-workshops

[7] Lu, H., et al., SHP2 inhibition overcomes RTK-mediated pathway reactivation in KRAS-mutant tumors treated with MEK inhibitors. Molecular cancer therapeutics, 2019. 18(7): p. 1323-1334.

[8] Smoot, M.E., et al., Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics, 2011. 27(3): p. 431-432.

[9] Virtanen, P., et al., SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods, 2020. 17(3): p. 261-272..

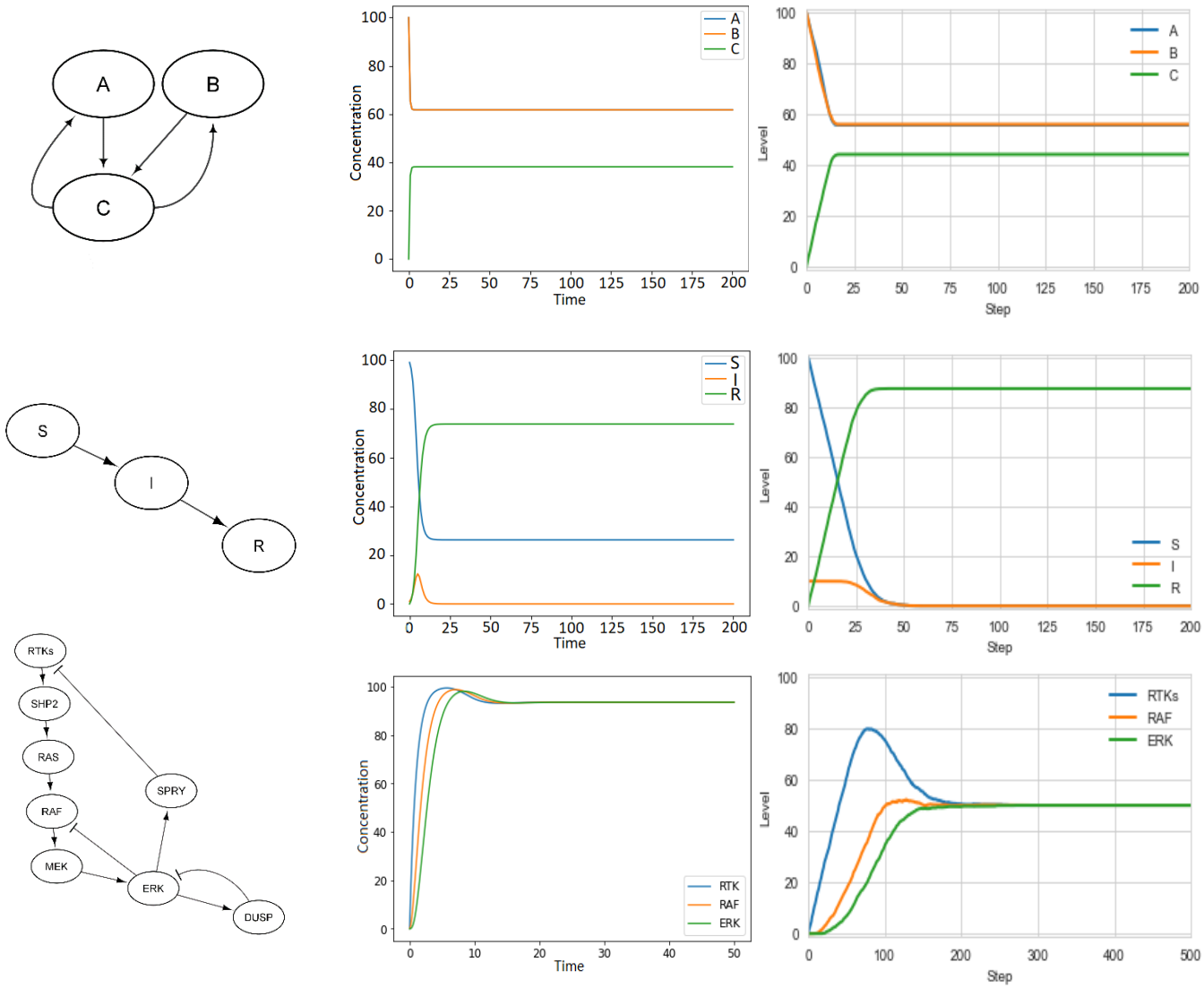


Figure 2: From left to the right: simple network figures (left), simulation results for three models using ODEs (middle) and discrete modeling approach (right). It is worth noting that y-axis of ODEs (middle) shows the concentration levels of different molecules, while in discrete modeling (right), it is the percentages of maximum possible level. From top to the bottom: Three models in this study, ABC (Top), SIR (middle), and RAS (bottom).

The Context Matrix: A Framework for Context-Aware Synthetic Biology

Camillo Moschner*
Department of Engineering,
University of Cambridge
Cambridge, United Kingdom
cm967@cam.ac.uk

Charlie Wedd*
Department of Engineering,
University of Cambridge
Cambridge, United Kingdom
cdw42@cam.ac.uk

Somenath Bakshi
Department of Engineering,
University of Cambridge
Cambridge, United Kingdom
sb2330@cam.ac.uk

1 INTRODUCTION

Synthetic biology seeks to design and build biological systems which perform useful functions, typically following a *Design-Build-Test-Learn* (DBTL) cycle. There have been many efforts to improve the DBTL cycle, such as through standardisation and automation [1, 4]. However, the design process remains particularly difficult to standardise. A major difficulty is in acquiring and evaluating knowledge of the vast numbers of potential design strategies and genetic parts available, from which many candidate designs could produce the intended functionality. Attempts at predictive design are further confounded by contextual issues: the same synthetic genetic construct can give a different functional performance in the context of a different host cell background, or even using the same host cell genotype but grown in a different physical or chemical environment. Many of these effects have previously been identified and characterised [2], but a unified, user-friendly framework for the consideration of contextual issues in the synthetic biology DBTL cycle has yet to be developed.

Here, we present the "context matrix" [5], a database of input factors and design principles to help users navigate contextual considerations in synthetic biology workflows.

2 THE CONTEXT MATRIX

The context matrix is a multi-dimensional list of input factors, categorised into the contexts of synthetic genetic construct, host cell and environment (Figure 1). The matrix describes the known contextual effects of each input factor on the output state of a biosystem, and can be used to quickly learn about factors which were previously unknown to the user. The organisation is intended to help users identify the key input factors for their biosystem, and ultimately aims to equip users with the knowledge to be able to consider context as another tool for synthetic biology design, rather than as an obstacle to be avoided. Here we present a "function-centric" view of synthetic biology, where the biosystem function is considered in the context of the synthetic construct, host and environment. We believe this to be a more helpful framing for context-aware synthetic biology than the more traditional construct-centric view, as it draws attention away from the construct and towards a more holistic view of the biosystem, where all three aspects are considered in concert to achieve the desired functional performance. The position of a biosystem within this contextual space is defined by the full combination of all input factors, which we term the "input landscape". The input landscape maps to a phenotypic output state, from which qualities of interest to the user can be measured, such as host cell fitness, or functional performance. An appreciation of the position of a biosystem within context space enables both better prediction (at the design stage) and troubleshooting (at the learn stage) of failure modes and output performance.

3 CONTEXT-AWARE SYNTHETIC BIOLOGY

The context matrix can be used as a tool to assist in the design, test and learn phases of the DBTL cycle.

At the design stage, the context matrix represents a database of known design strategies, and showcases that there may be many available routes to a desired function. In practice, this means there can exist very different biosystems in terms of the construct, host and environment used, but that perform the same function, and we term these "analogous engineered biosystems". The context matrix allows for comparison between analogous systems at the design stage, such that the most appropriate design for the required function can be chosen. This comparison will consider time and material constraints in light of the intended goal, and for industrial applications will need to be balanced against a rigorous techno-economic and life cycle assessment analysis.

At the test stage, the aim is often to compare many different engineered biosystems, which differ by construct, host or environment, and to evaluate which biosystem will give the best performance. Each of the three components of the context matrix is organised by input factors. This allows the experimenter to choose which factors to change, while simultaneously acting as a list of all other input factors which may need to be controlled. This quality of the matrix also makes it accessible to Design of Experiments (DoE) methods,

*Both authors contributed equally to this research.

which offer advantages for the efficient exploration of large input parameter spaces [3].

At the learn stage, the knowledge of previously studied contextual issues in the matrix can aid in troubleshooting failed designs and in the interpretation of experimental data.

4 COMMUNITY DEVELOPMENT

It is our aim for the context matrix to become a community-built resource, where its structure, contents and standards continually evolve and improve to meet the needs of the synthetic biology community. By maintaining and growing a repository of knowledge pertaining to the design, characterisation and understanding of engineered biosystems, the challenges presented by contextual issues may in time be overcome. The context matrix is hosted on GitHub (https://github.com/camos95/context_matrix). We welcome feedback on the context matrix to improve its structure, content and interface. To contribute, please contact us by email or submit a GitHub pull request. Contributions could include the submission of new input factors (or expanding or editing existing ones) and suggestions or code to improve the structure and usability of the matrix, but are not limited to this.

5 CONCLUSION

The field of synthetic biology has given rise to a wealth of knowledge for the design of biological systems. However, the complex interplay between the many components of biological systems often presents unexpected problems, and these are broadly defined as contextual issues. Here, we present a holistic framework to categorise known contextual issues and their solutions, which we call the context matrix. As the understanding of contextual issues in the field continues to grow, we envisage the context matrix to develop from a list of design principles into a full database, capable of taking a combination of input factors and predicting the performance of an engineered design. Ultimately, the context matrix aims to help develop an understanding of contextual issues which takes context from a challenging and unpredictable problem to an opportunity for the creation of novel designs with robust performance.

ACKNOWLEDGEMENTS

The full version of this work has been published by Moschner *et al.* [5].

FUNDING

This research in the SB laboratory was supported by the Wellcome Trust Award (grant number RG89305), a University Startup Award for Lectureship in Synthetic Biology (grant number NKXY ISSF3/46) and a Royal Society Research Grant Award (Award number G109931). CM was supported by the United Kingdom Biotechnology and Biological Sciences (BBSRC) University of Cambridge Doctoral Training Partnership 2 (BB/M011194/1). CW was supported by the United Kingdom Engineering and Physical Sciences Research Council (EPSRC) grant EP/S023046/1 for the EPSRC Centre for Doctoral Training in Sensor Technologies for a Healthy and Sustainable Future.

REFERENCES

[1] Carbonell, P., Jervis, A. J., Robinson, C. J., Yan, C., Dunstan, M., Swainston, N., Vinaixa, M., Hollywood, K. A., Currin, A., Rattray, N. J., Taylor, S., Spiess, R., Sung, R., Williams, A. R., Fellows, D., Stanford, N. J., Mulherin, P., Le Feuvre, R., Barran, P., Goodacre, R., Turner, N. J., Goble, C., Chen, G. G., Kell, D. B., Micklefield, J., Breitling, R., Takano, E., Faulon, J. L., and Scrutton, N. S. An automated Design-Build-Test-Learn pipeline for enhanced microbial production of fine chemicals. *Communications Biology 1* (2018), 66.

[2] Cardinale, S., and Arkin, A. P. Contextualizing context for synthetic biology - identifying causes of failure of synthetic biological systems. *Biotechnology Journal 7*, 7 (2012), 856–866.

[3] Gilman, J., Walls, L., Bandiera, L., and Menolascina, F. Statistical Design of Experiments for Synthetic Biology. *ACS Synthetic Biology 10*, 1 (2021), 1–18.

[4] Jessop-Fabre, M. M., and Sonnenschein, N. Improving reproducibility in synthetic biology. *Frontiers in Bioengineering and Biotechnology 7* (2019), 18.

[5] Moschner, C., Wedd, C., and Bakshi, S. The context matrix: Navigating biological complexity for advanced biodesign. *Frontiers in Bioengineering and Biotechnology 10* (2022), 1277.

[6] Scholz, S. A., Diao, R., Wolfe, M. B., Fivenson, E. M., Lin, X. N., and Freddolino, P. L. High-Resolution Mapping of the Escherichia coli Chromosome Reveals Positions of High and Low Transcription. *Cell Systems 8*, 3 (2019), 212–225.e9.

[7] Yeung, E., Dy, A. J., Martin, K. B., Ng, A. H., Del Vecchio, D., Beck, J. L., Collins, J. J., and Murray, R. M. Biophysical Constraints Arising from Compositional Context in Synthetic Gene Networks. *Cell Systems 5*, 1 (2017), 11–24.e12.

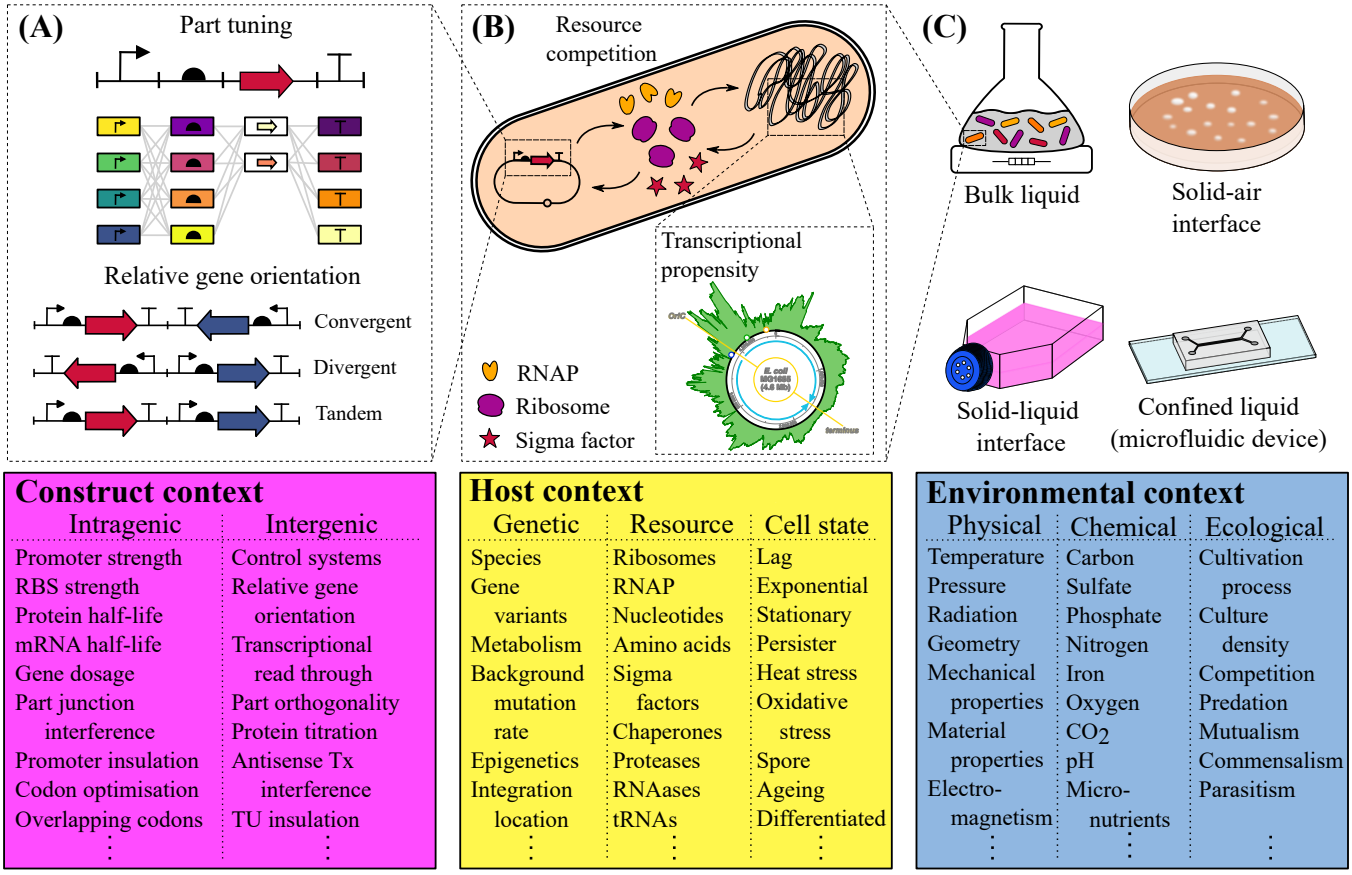


Figure 1: The three primary contexts of the function of an engineered biosystem. (A) The construct context can be divided into intragenic and intergenic contexts. Part tuning allows us to choose different genetic parts, and hence control some aspects of the intragenic context. Relative gene orientation is an example of intergenic context, and can significantly affect gene expression [7]. (B) The host context involves considerations of genetics, resource competition and the cell state. One genetic context is the genome integration location, which can significantly affect gene expression due to differing transcriptional propensities around the genome [6]. (C) The environmental context considers the chemical composition and physical conditions of the biosystem, and any ecological effects present. Several cultivation processes are shown, the choice of which can affect gene expression and population growth. Acronyms: TU = transcription unit, RNAP = RNA polymerase, RBS = ribosomal binding site. Figure from Moschner *et al.* [5].

PLATERO: A Plate Reader Calibration Protocol to work with different instrument gains and asses measurement uncertainty

Yadira Boada[1,3], Alba González-Cebrián[2], Joan Borràs-Ferrís[2], Jesús Picó[1], Alberto Ferrer[2], Alejandro Vignoni[1,*]

[1]Synthetic Biology and Biosystems Control Lab, Instituto de Automática e Informática Industrial, [2]Multivariate Statistical Engineering Group, Department of Applied Statistics and O.R. and Quality, Universitat Politècnica de València, València, Spain.

[3] Centro Universitario EDEM, Escuela de Empresarios, Muelle de la Aduana s/n, 46024, Valencia, Spain.

vignoni@isa.upv.es

1 BACKGROUND

One of the most common sources of information in Synthetic Biology (SynBio) is the bulk data of fluorescent proteins expressed in microorganisms. A plate reader is one of the measuring devices used for collecting and quantifying fluorescent measurements. But these data are highly dependent on the experiment characteristics, or the device features such as the gain setup for that experiment. Although some studies have proposed fluorescein as a robust green fluorescent calibrant [2], the gain dependence of the measurements remains unsolved. Also, the rising automation of the Design-Build-Test-Learn (DBTL) cycle is demanding the use of plate readers for monitoring synthetic gene circuits. Therefore, laboratories are generating tons of experimental data that can not be easily compared according to every lab protocols In this context, new data and calibration standards should encourage the exchange of information between labs because it is crucial to overcome both gene circuits characterization, and reproducibility problems [3].

Here, we propose PLATERO as Matlab Toolbox for calibration of fluorescence measurements (in this case for a green fluorescent protein-GFP) collected by the plate reader, which provides standardized data that also are independent of the instrument gain (see Figure 1). PLATERO's main features are (1) it allows us to compare data between experiments that have been carried out at different gain levels, and (2) it provides us with a quantification of the measurement uncertainty based on linearity and bias analysis (not fully showed here) [4]. Nevertheless, this pipeline can be easily modified and extended to other type of calibration model or other measurement instruments, or it can be adjusted to another colors using the appropriate dyes [1].

2 GAIN CALIBRATION MODEL

As in Figure 1A, the gain of a plate reader is one of the key parameters to set up before measuring a fluorescent protein. If the gain (G) is too low, small levels of fluorescence will not be detected by the instrument. Conversely, if G is too high, the measurement will be above of the upper limit of the sensor's range and leading to its saturation, so fluorescence cannot be measured either. Firstly, we assume that every $k-th$ measurement of a fluorescent protein F_{real} expressed by cells includes the background fluorescence (F_{BLK}) of the culture medium, and the fluorescence of the protein itself (F_p). That is, $F_{real}(k) = F_{BLK}(k) + F_p(k)$. Since this study is not working with living cells, the cell auto-fluorescence has been neglected. Then, F_m is the fluorescence measured by a 96-well plate reader that follows an exponential function of G, and it depends on the F_{real} as $F_m(k) = F_{real}(k) \cdot e^{b_1 \cdot G + b_2 \cdot G^2}$. A linear model did not correctly adjust the gains in the F_m data. To obtain the true value of fluorescence F_p accounting several experiments performed at different gain levels:

$$F_p(k) = \left(F_m(k) - F_{BLK,G}(k)\right) \cdot e^{-b_1 \cdot G - b_2 \cdot G^2} \qquad (1)$$

where $F_{BLK,G}$ is the background fluorescence at any gain G, b_1 and b_2 are the coefficients of the linear and the quadratic terms of the gain, respectively (see Table 1). These coefficients are statistically significant and they have been validated in a previous study. In PLATERO TOOLBOX, The function *gaincfs.m* computes these coefficients, and the function *checkblk.m* provides a fast analysis of the blank wells (potential outliers were excluded from F_{BLK} estimation).

3 STANDARIZATION OF UNITS

PLATERO also converts the arbitrary units of the F_p data (units of any plate reader) to standard equivalent concentration units of fluorescein (C_p). We used a reference fluorescein solution to calibrate the concentration levels of the green fluorescence measurements (see Figure 1B). The equivalent concentration units are similar to the Molecules Equivalent of Fluorescein (MEFL) using a conversion factor that considers the relationship between the number of molecules and the concentration (not included here). Therefore, for the $k-th$ we assumed a linear model between the fluorescence

F_p and the concentration level C_p knowing the volume of every well in the 96-well plate:

$$C_p(k) = c_0 + c_1 \cdot F_p(k) \quad (2)$$

where c_0 is the intercept, and c_1 is the scale factor for the conversion (see Table 1). One might expect a calibration curve containing the (0,0) point (no fluorescence measured at a concentration of 0 nM, i.e. $c_0 = 0$). However, it is important to have $c_0 \neq 0$ to capture the offset introduced by the plate reader. These coefficients were estimated using the *cfcoeff.m* function, and they have been validated as statistically significant in a previous study. *cfcoeff.m* also returns further information about the quality of the fitting. All the PLATERO functions are fully available in the open access repository https://github.com/sb2cl/PLATERO.

4 QUANTIFYING PREDICTION UNCERTAINTY

The accuracy of a measurement instrument (specifically referred to as *Bias*) reflects the difference between the observed measurements and their corresponding *true* values. We consider a simple model for the relative bias:

$$Bias \cdot \frac{1}{C_T(k)} = \frac{C_p(k) - C_T(K)}{C_T(k)} = d_0 \cdot \frac{1}{C_T(k)} + d_1 \quad (3)$$

where $C_T(k)$ is the *true* value of the $k-th$ measurement of concentration C_p (obtained from a master or gold standard), d_0 the intercept, and d_1 the slope of the model. All terms from Equation 3 can be estimated by the function *biasanalysis.m*.

Once the measurements have been calibrated as in sections 2 and 3, PLATERO also estimates the standard deviation of the plate reader's *Bias* (s_{Bias}). The degrees of freedom (DF) of the *Bias* error from Equation 3 together with the s_{Bias} are used to calculate the $(1-\alpha)\cdot 100\%$ confidence interval ($CI_{Cp}(k)$) for a given concentration C_p:

$$CI_{Cp}(k) = C_p(k) \pm t_{DF,\alpha/2} \cdot s_{Bias} \cdot C_T(k) \quad (4)$$

where C_T is the concentration value that undoes the scaling of the bias and gives the amplitude corresponding to a particular concentration level to the confidence interval. Ideally, this should be the true concentration level C_T. However, if C_T remains unknown in model exploitation, the C_p will be used instead. The function *cipred.m* quantifies the C_p prediction uncertainty.

5 USING PLATERO TO CALIBRATE A PLATE READER

The PLATERO protocol has been constructed and statistically validated using a data set of 1509 fluorescein samples coming from a serial dilution with at least five (5) concentrations (see Figure 1B). In each experiment, eight (8) replicates for each concentration were measured by two plate readers. PL1: the BioTek Cytation 3 plate reader at four different gains (starting with the smallest gain allowed by the instrument) G =50, 60, 70, and 80; and PL2: the TECAN infinite 200 plate reader at four different gains G =60, 70, 80, and 90.

After analyzing the data, we found that some of the samples were out of range for some of the gains as we expected. So, we excluded them from the model building step, but they were part of the validation data later on. From the non-excluded data, we used 70% for model building, and 30% for the model validation step that follows a randomly-selection of the samples coming from the 96-well plate. This avoids possible location effects due to the selection of wells in a specific order of rows or columns. The plate reader calibration generates the coefficients values listed in Table 1.

Table 1: Estimated coefficients (for any gain G)

Coefficients	Median PL1	Median PL2
b_1	0.24298	0.19789
b_2	$-9.933\cdot10^{-4}$	$7.0272\cdot10^{-4}$
c_0	$-1.1185\cdot10^{-3}$	$9.7161\cdot10^{-4}$
c_1	1.0576	19.01

Validating the plate reader calibration was the final step. We used PLATERO to predict the fluorescein concentration levels C_p of the measurements that were excluded from the calibration procedure (including the non-used samples and the non-used concentrations C_{pnu}). Then, we calculated the concentration predictions C_p related to the real concentration values showed in the X-*axis* of Figure 2A. Even for the samples excluded from the model construction (Figure 2A-left panel), we obtained good prediction quality, where the ratio $C_p/\text{Real}_{\text{concentration}}$ is approximately 1.

6 ACKNOWLEDGMENTS

Research funded by MCIN/AEI/10.13039/501100011033 grant number PID2020-117271RB-C21 and GVA grant CIAICO/2021/159. Y.B. thanks Grant PAID-10-21 Acceso al Sistema Español de Ciencia e Innovación-UPV; and also to Secretaría de Educación Superior, Ciencia, Tecnología e Innovación of Ecuador (Scholarship Convocatoria Abierta 2011).

REFERENCES

[1] Beal, J., and et al. Multicolor Plate Reader Fluorescence Calibration. *Synthetic Biology* (07 2022). ysac010.

[2] Boada, Y., Vignoni, A., Alarcon-Ruiz, I., Andreu-Vilarroig, C., Monfort-Llorens, R., Requena, A., and Picó, J. Characterization of gene circuit parts based on multiobjective optimization by using standard calibrated measurements. *ChemBioChem 20*, 20 (2019), 2653–2665.

[3] Fedorec, A. J. H., and et al. Flopr: An open source software package for calibration and normalization of plate reader and flow cytometry data. *ACS Synthetic Biology 9*, 9 (2020), 2258–2266.

[4] Montgomery, D. C. *Introduction to statistical quality control.* John Wiley & Sons, 2020.

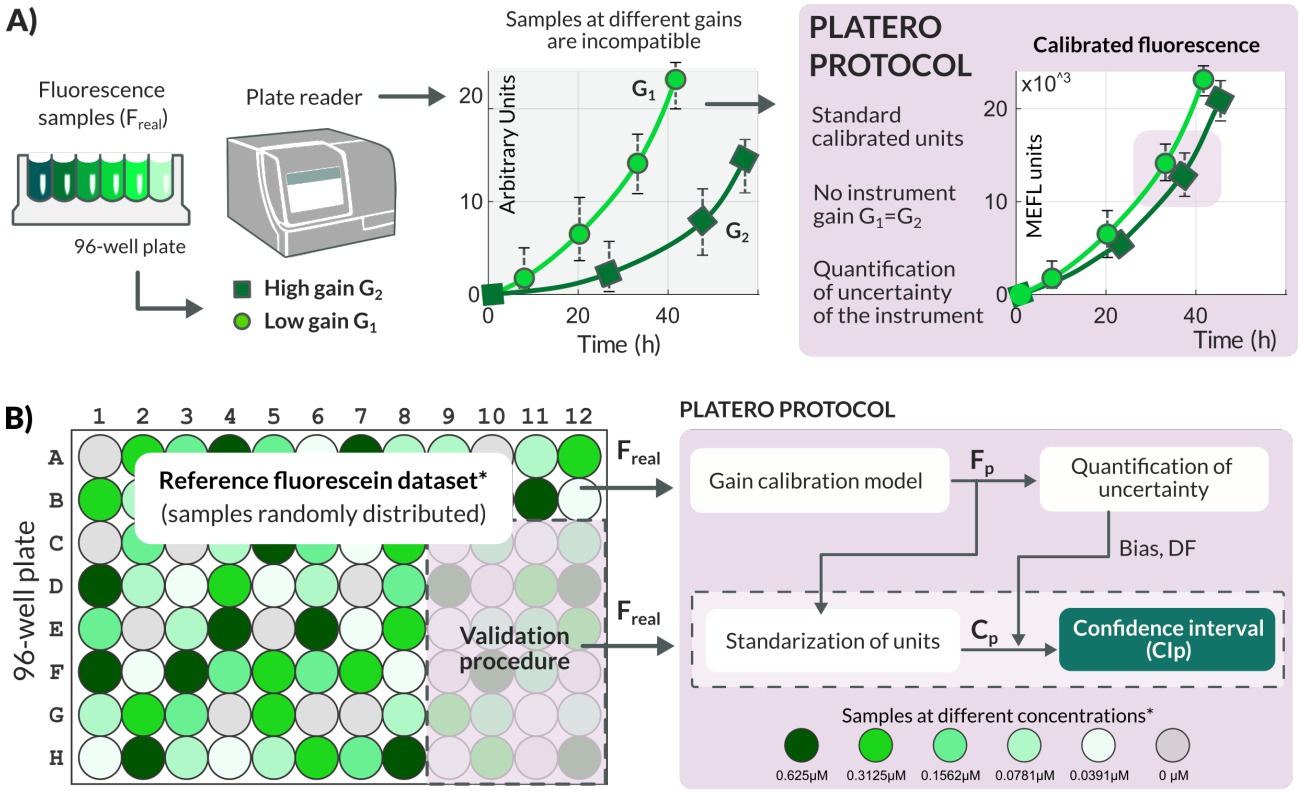


Figure 1: A) The PLATERO protocol brings the experimental measurements into a common gain-independent domain using standard fluorescence units as well. It also incorporates an instrument analysis that provides an estimate of the uncertainty that can be expected in the predicted fluorescence value. B) Schema representing the Model building and the Model validation steps of the PLATERO protocol. Particularly, eleven out of the sixteen wells (≈ 70%) for each concentration level were randomly selected for the gain calibration model. The remaining data were used only for validation.

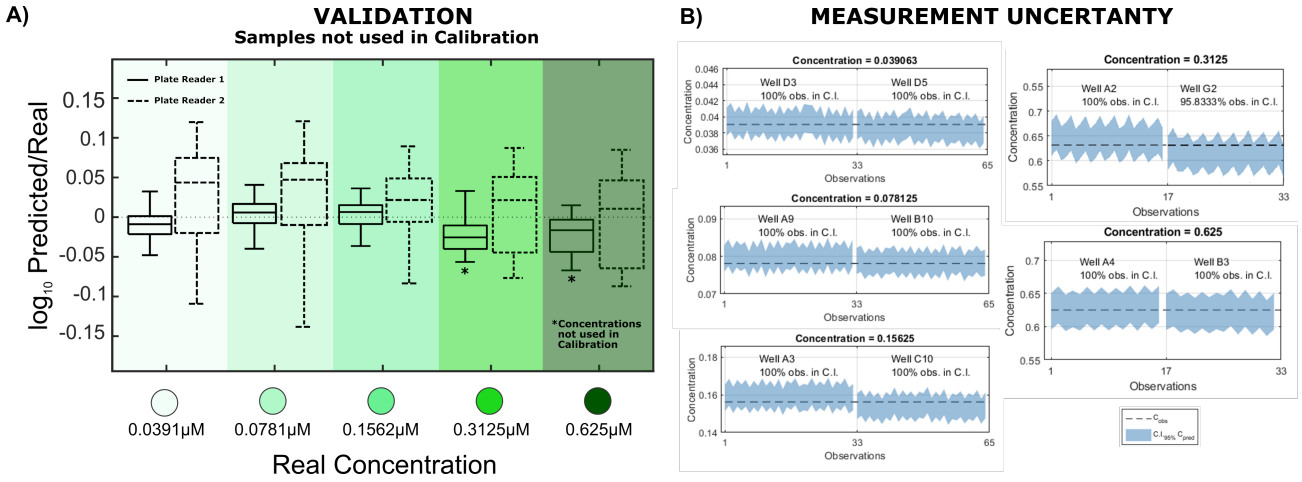


Figure 2: A) Model validation (with samples and concentrations C_{pnu} not used in the calibration) finds good performance and quality of the resulting gain calibration model. The box-plot illustrates the ratio between the predicted concentrations and the real ones. In each box, the center line indicates median, top and bottom edges show 25th and 75th percentiles, and whiskers extend from 9% to 91%. B) For the uncertainty quantification CI_{Cp}, the bias and the degrees of freedom obtained after validation were $s_{Bias} = 0.0022472$ and DF=1045, for PL1 and $s_{Bias} = 0.086015$ and DF=1758 for the PL2, respectively. Confidence intervals obtain with PLATERO for the concentrations used in the model construction. 100% of the real values lie into the corresponding CI_{Cp} for each prediction (wells D3, D5, A9, B10, A3, C10). Moreover, the non-used concentrations C_{pnu} that were estimated by PLATERO (wells A2, G2, A4, B3) are within the confidence interval with a confidence level 95%. This demonstrates PLATERO's high predictive capacity.

Steps Towards Functional Synthetic Biology

[1]Ibrahim Aldulijan [2,*]Jacob Beal [3]Sonja Billerbeck [4]Jeff Bouffard [5]Gaël Chambonnier [6]Nikolaos Delkis [7]Isaac Guerreiro [8]Martin Holub [9]Daisuke Kiga [10]Jacky Loo [11]Paul Ross [7]Vinoo Selvarajah [12]Noah Sprent [13]Gonzalo Vidal [14]Alejandro Vignoni

[1]Stevens Institute of Technology, [2]Raytheon BBN, [3]University of Groningen, [4]Concordia University, [5]Massachusetts Institute of Technology, [6]University of Thessaly, [7]iGEM Foundation, [8]Delft University of Technology, [9]Waseda University, [10]Aalto University, [11]BioStrat Marketing, [12]Imperial College London, [13]Newcastle University, [14]Universitat Politecnica de Valencia

*Corresponding author: jakebeal@ieee.org

1 INTRODUCTION

While synthetic biology has made great progress in methods for modular assembly of genetic sequences and in engineering biological systems with a wide variety of functions, current paradigms entangle sequence and functionality in a manner that makes abstraction difficult, reduces engineering flexibility, and impairs predictability and design reuse. Functional Synthetic Biology [1] proposes a roadmap to overcome these limits by focusing on behavior descriptions, predictability, flexibility, and risk reduction, so synthetic biologists can more effectively share successes and avoid failures.

The iGEM community, like other synthetic biology communities, faces challenges in effective sharing and reuse of biological devices. These are particularly acute for iGEM, since iGEM teams need to execute projects in only a few months and many team members have little prior experience. At the same time, barriers for adoption are lowered by the culture of openness, sharing, and reuse that is encouraged by iGEM. For these reasons, the iGEM Engineering Committee has been working to implement the early phases of the Functional Synthetic Biology roadmap in the context of iGEM's annual DNA distribution.

2 AGILE CURATION OF DNA DESIGN PACKAGES

As a first step, we have deployed an agile data curation workflow for community development of DNA design packages, leveraging distributed version control and continuous integration tooling. Each year, iGEM sends teams a distribution of DNA parts expected to be useful for their projects. For the 2022 season, iGEM developed an all new distribution, enlisting a larger community to aid in its design. To support the community design process, we built on work from the DARPA SD2 program [6] to deploy an agile data curation workflow on GitHub (Figure 1). With this workflow, contributors submit DNA design packages developed with spreadsheets and design files. These undergo community review and revision using the Gitflow workflow, while complementary automation tests packages for errors and collates package contents to produce a distribution plan and synthesis orders.

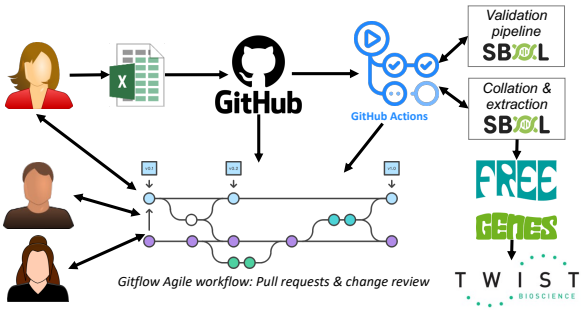


Figure 1: iGEM distribution agile data curation workflow.

Workflow automation is implemented using GitHub Actions, an integrated continuous integration and continuous delivery (CI/CD) framework. In our usage, continuous integration maps to checking specifications for coherence and correctness, while continuous deployment maps to compilation of all designs together into a complete plan for the distribution and the synthesis orders for building it.

Figure 2 shows details of this workflow. Excel templates provide a user-friendly interface to specify "packages" organizing groups of related parts (e.g., a collection of fluorescent reporters), and the build plans for how to combine part sequences into composites, flank them with prefixes and suffixes for BioBricks or Type IIS assembly, and insert them into plasmids for propagation and dissemination.

The workflow first exports Excel into two formats: CSV for git diff review, and SBOL3 [4] that specifies the parts (SBOL Components) and combinatorial build plans (SBOL CombinatorialDerviations). Parts are either fetched from public data sources by their identifiers (e.g., NCBI accession, BioBrick part number) or imported from files in the same directory as the sheet. The build plan is then validated to ensure it is coherent and fully specified. After validation, build plans are compiled to a full specification for each package. Each CombinatorialDerivation is expanded into a list of specific composite parts to produce, sequences are calculated for each construct, and a human-readable README file is generated summarizing the package and its contents. Finally, all packages are collated to produce the complete distribution,

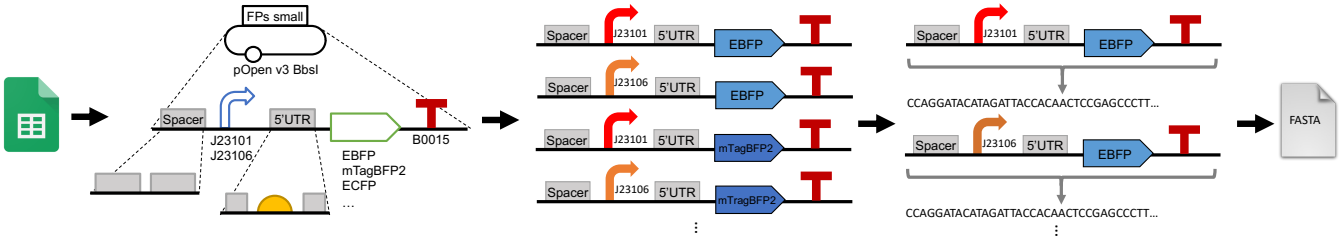


Figure 2: Production of synthesis orders from DNA package plans, as used by the iGEM Distribution repository: packages are specified in Excel sheets, from which are extracted SBOL3 documents specifying libraries of plasmid build plans. Each build plan is expanded into a list of all of the specific composite parts to produce. A sequence is then calculated for each construct, and the portion of each construct to be synthesized is exported to FASTA for placing a synthesis order.

and the SBOL is exported to GenBank for compatibility with other design tools, and to FASTA for ordering the plasmid inserts that are to be synthesized.

This workflow was able to be used effectively by the iGEM Engineering Committee in developing the iGEM 2022 distribution (available at https://github.com/iGEM-Engineering/iGEM-distribution), supporting a rapid pace of development and review by a large group of contributors. During the main development period of the distribution, from January 1st to February 16th, 2022, 15 contributors at 11 institutions in 8 different countries produced 571 commits, which were reviewed and merged in 87 pull requests, an average of nearly 2 contributions per day. The resulting distribution contains 16 packages organizing several hundred parts into thematic collections such as "CRISPR-Cas", "Fluorescent Reporters", "Small Molecule Inducers", and "Plant Parts." Critically, the learning curve also proved reasonable: most contributors were not programmers, and many had never used git before.

3 WORK IN PROGRESS

We are continuing to work towards the Functional Synthetic Biology vision, building on the lessons from the 2022 distribution. First, automated validation is being extended to include biological considerations, using pydna [5] to check assembly compatibility and using synthesis company APIs to check synthesizability. We have also been improving biologist-focused documentation for package development and use of git-based workflows, to support adoption of these methods by iGEM teams and the larger synthetic biology community.

Next, we are implementing a dependency management system for DNA design packages based on SBOL Enhancement Proposal (SEP) 054 (available at https://github.com/SynBioDex/SEPs). Analogous to software package management systems, this will allow DNA design packages to be broken out into their own repositories and maintained separately, then imported for use in the distribution or other packages. This is required for scalability to a large community and to minimize duplication and forking of materials.

Finally, we are running interlaboratory studies to develop reliable transcriptional toolkits. Prior work shows transcriptional regulators can be effectively insulated from genetic context (e.g., [2, 3]), but these results are not readily accessible or joined with predictive models. The committee is thus running studies to produce models quantifying insulated systems in replicable ERF/cell units. The first targets are constitutive promoters (for consistent expression levels) and fluorescent reporters (for debugging and quantification), to be followed by inducible promoters (for adjustable regulation and sensing). If successful, these will be collected in packages for distribution, making it simple for iGEM teams and other users to test new devices with known-reliable sensors, adjustable inputs, and reporters.

4 ACKNOWLEDGEMENTS

In addition to the listed authors, many other members of the iGEM Engineering Committee have made contributions that have supported the results described. This work was partially supported by AFRL/DARPA contract FA8750-17-C-0184, Concordia and Newcastle Universities, ERC Advanced Grant 883684, BBSRC award BB/M011178/1, Grant MINECO/AEI, EU DPI2017-82896-C2-1-R and MCIN/AEI/10.13039/501100011033 grant PID2020-117271RB-C21. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations. Views, opinions, and/or findings are should not be interpreted as representing the official views or policies of any of the funders.

REFERENCES

[1] Aldulijan, I., et al. Functional synthetic biology. *arXiv preprint arXiv:2207.00538* (2022).

[2] Carr, S. B., Beal, J., and Densmore, D. M. Reducing dna context dependence in bacterial promoters. *PloS one 12*, 4 (2017), e0176013.

[3] Lou, C., Stanton, B., Chen, Y.-J., Munsky, B., and Voigt, C. A. Ribozyme-based insulator parts buffer synthetic circuits from genetic context. *Nature biotechnology 30*, 11 (2012), 1137–1142.

[4] McLaughlin, J., et al. The synthetic biology open language (SBOL) version 3: simplified data exchange for bioengineering. *Frontiers in Bioengineering and Biotechnology* (2020), 1009.

[5] Pereira, F., Azevedo, F., Carvalho, Â., Ribeiro, G. F., Budde, M. W., and Johansson, B. Pydna: a simulation and documentation tool for dna assembly strategies using python. *BMC bioinf. 16*, 1 (2015), 1–10.

[6] Roehner, N., et al. Data representation in the DARPA SD2 program. In *IWBDA* (2021).