



Boston - September 12-14

15th International Workshop on Bio-Design Automation
Boston, Massachusetts, USA
September 11th-14th 2023

Foreword

Welcome to IWBDA 2023!

The IWBDA 2023 Organizing Committee welcomes you to the Fifteenth International Workshop on Bio-Design Automation (IWBDA). IWBDA brings together researchers from an array of biological and computational domains including synthetic biology, systems biology, and design automation. The focus of IWBDA is on concepts, methodologies, and tools for the analysis, design, and synthesis of engineered 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 researchers in specific domains of biology. As the engineering of biological systems develops from a process driven by experimentation to one driven by standardization and engineering principles, there is a tremendous opportunity to engage with new ideas and new communities. IWBDA offers a forum for cross-disciplinary discussion, with the aim of seeding and fostering collaboration between biological and computational research communities.

This year, the program consists of 4 workshops, 20 contributed talks, and 9 short talks (in lieu of posters): The talks are organized into 7 sessions:

- Sequence Manipulation and Gene Expression
- Biosecurity
- Modeling
- Genetic Design
- Automating Evolution
- Machine Learning
- Improving Workflows through Software

In addition, we are very pleased to have Dr. Nicole Wheeler from the University of Birmingham as a distinguished invited speaker. We would like to thank all the participants for their contributions to IWBDA and highlight the efforts of the Program Committee and 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.

Organizing Committee

Organizing Committee

General Chair, Finance Chair - Traci Haddock, Asimov

Program Committee Chair - Natasa Miskov-Zivanov, University of Pittsburgh

Publication Chair - Nicholas Roehner, Raytheon BBN

Local Chair - Daniel Fang, University of Colorado Boulder

Diversity Chair - Lukas Buecherl, University of Colorado Boulder

Co-Workshop Chair - Carolus Vitalis, University of Colorado Boulder

Co-Workshop Chair - Gonzalo Vidal, Newcastle University

Co-Web Chair - Aaron Adler, Raytheon BBN

Co-Web Chair - Prashant Vaidyanathan, Oxford Biomedica

Bio-Design Automation Consortium

President - Aaron Adler, Raytheon BBN

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

Treasurer - Traci Haddock, Asimov

Board Member - Douglas Densmore, Boston University

Clerk - Prashant Vaidyanathan, Oxford Biomedica

Program Committee

Natasa Miskov-Zivanov	University of Pittsburgh
Yasmine Ahmed	University of Pittsburgh
Chris J. Myers	University of Colorado Boulder
Aaron Adler	Raytheon BBN
Jake Beal	Raytheon BBN
Lukas Buecherl	University of Colorado Boulder
Traci Haddock	Asimov
Emilee Holtzapple	University of Pittsburgh
Ernst Oberortner	DOE Joint Genome Institute
Ayush Pandey	University of California, Merced
William Poole	California Institute of Technology
Nicholas Roehner	Raytheon BBN
Kenza Samlali	Concordia University
Radhakrishna Sanka	Nona Research Foundation
Daniel Schindler	Max Planck Institute for Terrestrial Microbiology
Cheryl Telmer	Carnegie Mellon University
Prashant Vaidyanathan	Oxford Biomedica
Carolus Vitalis	University of Colorado Boulder
Gaoxiang Zhou	University of Pittsburgh

Program

All times EST.

Monday, September 11 - 665 Commonwealth Ave, 17th Floor

- 8:00-9:00 Registration and breakfast
- 09:00-09:10 Welcome and kick-off of Bio-Design Week
- 09:10-10:30 Workshop: SBOL Data Model
- 10:30-11:00 Coffee break
- 11:00-12:30 Workshop: SBOL Visual
- 12:30-13:30 Lunch
- 13:30-15:00 Workshop: Programmatic genetic design automation using LOICA
- 15:00-15:30 Coffee break
- 15:30-17:00 Nona Works Team Building and Mixer

Tuesday, September 12 - 665 Commonwealth Ave, 17th Floor

- 08:00-09:00 Registration and breakfast
- 09:00-09:10 Welcome to IWBDA
- 09:10-10:15 IWBDA Talks: Sequence Manipulation and Gene Expression**
 - 09:10-09:30 *Energy Aware Technology Mapping*. Erik Kubaczka, Tobias Schwarz, Jérémie Marlhens, Maximilian Ge, Nicolai Engelmann, Christian Hochberger and Heinz Koeppl
 - 09:30-09:50 *Local RNA Feedback: More Logic, Less Leakage*. Nicolai Engelmann, Maik Molderings and Heinz Koeppl
 - 09:50-10:10 *Using learned representations of genetic circuits to evaluate sequence-level mutations*. Olivia Gallup and Harrison Steel
- 10:10-10:15 Lightning Talks
 - 10:10-10:15 *Multi-Site Mutagenic Protein Library Design with Controlled Annealing Temperature*. Yehuda Binik, Ayesha Chaudry, Akira Takada, Georgios Papamichail and Dimitris Papamichail

- 10:15-10:45 Coffee break

10:45-11:45 Keynote: Dr. Nicole Wheeler

Dr. Nicole Wheeler is a Turing Fellow at the University of Birmingham and also serves as a technical consultant for the Nuclear Threat Initiative. Dr Wheeler's work focuses on the development of computational screening tools for identifying DNA from emerging biological threats, establishing genomic pathogen surveillance in resource-limited settings, One Health surveillance of antimicrobial resistance, and the ethical development of artificial intelligence

(AI) for health applications. She has a background in biochemistry and microbial genomics, complemented by experience in developing machine learning methods for predicting the effects of genetic variation on the virulence of pathogens. She has provided expertise on machine learning for genomic pathogen surveillance for several international programs, including a world-first AI-driven One Health AMR surveillance system. She is also actively involved in public outreach and the development of governance frameworks to ensure the safe and responsible development of technologies for health improvement.

11:45-13:30 Lunch

13:30-14:30 IWBDA Talks: Biosecurity

- 13:30-13:50 *Biological Malware Detector*. Muntaha Samad, Dan Wyschogrod and Jacob Beal
- 13:50-14:10 *DNA Editing Game (DEGA) Theory*. Nicholas Roehner
- 14:10-14:30 *Splicing-based Biocontainment Devices*. Allison Taggart, Miles Rogers and Jacob Beal

14:30-15:00 Coffee break

15:00-16:30 Biosecurity Panel and Discussion Session

We are pleased to announce that IWBDA will host a biosecurity panel with panelists Dr. Nicole Wheeler (see Keynote above) and Dr. Peter Carr, moderated by Dr. Jacob Beal. This panel will explore the relationship between design tools and biosecurity.

Dr. Peter Carr is a Senior Scientist at the Massachusetts Institute of Technology's Lincoln Laboratory, where he leads the Synthetic Biology research program. His research interests include genome engineering, rapid prototyping of both hardware and wetware, DNA synthesis and error correction, and biosecurity. He is the Director of Judging for the International Genetically Engineered Machine (iGEM) competition and a founding member of the Synthetic Biology Center at MIT. He received his bachelor's degree in Biochemistry from Harvard, and his PhD in Biochemistry and Molecular Biophysics from Columbia University.

Dr. Jacob Beal is an Engineering Fellow at Raytheon BBN and is the lead developer for FAST-NA Scanner, a signature-based biosecurity screening tool used by multiple DNA synthesis companies. He also co-led the IGSC's Regulated Pathogen Database update, and is co-organizing international standards for testing biosecurity sequence screening systems.

16:30-17:00 Break and travel time to dinner

17:00-20:00 Dinner at Sunset Cantina

Wednesday, September 13 - 201 Brookline Avenue, 4th Floor

08:30-09:00 Breakfast

09:00-10:05 IWBDA Talks: Modeling

- 09:00-09:20 *A Collection of Biological Models for the Development of Infinite-State Stochastic Model Checking Tools.*
Lukas Buecherl, Payton J. Thomas, Mohammad Ahmadi, Josh Jeppson, Andrew Gerber, Eric Reiss, Chris Winstead, Hao Zheng, Zhen Zhang and Chris J. Myers
- 09:20-09:40 *A neural-mechanistic hybrid approach improving the predictive power of genome-scale metabolic models.*
Bastien Mollet, Jean-Loup Faulon, Léon Faure and Wolfram Liebermeister
- 09:40-10:00 *Knowledge-Based Pathway Extraction and Verification.* Gaoxiang Zhou and Natasa Miskov-Zivanov
- 10:00-10:05 Lightning Talks
 - 10:00-10:05 *Model Checking of Interval Discrete-Time Markov Chain for Biochemical Pathways.* Krishnendu Ghosh

10:05-10:35 Coffee break

10:35-11:40 **IWBDA Talks: Genetic Design**

- 10:35-10:55 *Guided Design of Genetic Circuits Exploiting Stochastic Model Verification.* Lukas Buecherl, Mohammad Ahmadi, Hao Zheng and Chris J. Myers
- 10:55-11:15 *Design-Build-Test-Learn of Sponge RNAs for Synthetic Gene Circuits.* Scott Stacey, Harrison Steel and Antonis Papachristodoulou
- 11:15-11:35 *Rule-based generation of synthetic genetic circuits.* Masayuki Yamamura, Ryoji Sekine, Kazuteru Miyazaki, Sota Okuda, Naoki Kodama and Daisuke Kiga
- 11:35-11:40 Lightning Talks
 - 11:35-11:40 *Analysis of ASR inducible promoter in different conditions in Escherichia coli.* Maria Jose Mesa-Rodriguez, Domenica Cuneo-Campodonico, Martin Gutierrez and Alberto J. Donayre-Torres

11:40-13:00 Lunch

13:00-13:40 **Invited Talks**

- 13:00-13:20 *Kernel: Evolving Genetic Design.* Kevin LeShane
- 13:20-13:40 *A Genetic Construct Simulator for Faster Design.* Alina Ferdman

13:40-14:45 Discussion session: Growing our Community

14:45-15:15 Coffee break

15:15-15:45 Discussion summary

15:45-16:45 **IWBDA Talks: Automating Evolution**

- 15:45-16:05 *Long-term evolution of bacteria for maximal growth rate.* Antoine Vigouroux and Johan Paulsson
- 16:05-16:25 *Engineering Continuous Directed Evolution with Single Cell Optogenetic Selection and Microfluidics.* Jess James, Sebastian Towers, Idris Kempf, Jingyu Wang, Jakob Foerster and Harrison Steel

- 16:25-16:45 *An Automated Platform for Accelerating Adaptive Laboratory Evolution.* Marco Corrao and Harrison Steel

16:45-16:50 Late Breaking Lightning Talks

- 16:45-16:50 *Biology that starts at a computer.* Dave Vance

16:50-17:00 Break and travel to the reception

17:00-19:30 Networking Reception sponsored by Asimov

Thursday, September 14 - 665 Commonwealth Ave, 17th Floor

08:30-09:00 Breakfast

09:00-09:45 Workshop: How to Write a Technical Note for ACS Synthetic Biology

09:45-10:45 Nona Works: Presentations

10:45-11:15 Coffee break

11:15-12:20 IWBDA Talks: Machine Learning

- 11:15-11:35 *SeqImprove: Machine Learning Assisted Curation of Genetic Circuit Sequence Information.* Zach Sents, Duncan Britt, William Mo and Chris J. Myers
- 11:35-11:55 *Automated model curation using LLMs: Integration of ChatGPT with the DySE framework.* Emilee Holtzapple, Tanvi Verma and Natasa Miskov-Zivanov
- 11:55-12:15 *Encoding Process Markers with Neural Networks to Simplify the Complexity of Engineering.* Haomiao Luo, Anya Zivanov and Natasa Miskov-Zivanov
- 12:15-12:30 Lightning Talks
 - 12:15-12:20 *How to build and train your ANN (In-Vivo) From zero to hero.* Tomás Fuentes Araya and Martín Gutiérrez
 - 12:20-12:25 *AI algorithms for classification and generation of spatial/temporal patterns in cell colonies.* Valeria Navarrete, Freddy Aguilar and Martín Gutiérrez
 - 12:25-12:30 *Using Machine Learning to Infer RNA Velocity Fields.* Taos Transue and Payton Thomas

12:30-13:30 Lunch and Nona Works Voting

13:30-14:25 IWBDA Talks: Improving Workflows through Software

- 13:30-13:50 *A Report on SynBio Data Management Practices.* Carolus Vitalis, Sai Samineni, Chris Myers and Pedro Fontanarrosa
- 13:50-14:10 *Software for Synthetic Biology Workflows: How to Improve Your Productivity and Impact.* Chris J. Myers, Lukas Buecherl, Daniel Fang, Pedro Fontanarrosa, William Mo, Sai P. Samineni, Gonzalo Vidal, Carolus Vitalis, Guillermo Yanez-Feliu and Timothy J. Rudge

- 14:10-14:25 Lightning Talks
 - 14:10-14:15 *DBTL Engineering cycle automation: Improving basic parts characterization in the Learn stage by Automation of the Test stage.* Yadira Boada, Anna Pushkareva, Harold Díaz-Iza, Andrés Arboleda-García, Jesús Picó and Alejandro Vignoni
 - 14:15-14:20 *PUDU: Simple Liquid Handling Robot Control for Synthetic Biology Workflows.* Gonzalo Andrés Vidal Peña, Carolus Vitalis, Matt Burridge, Lukas Buecherl, David Markham, Chris Myers and Timothy Rudge
 - 14:20-14:25 *A fully in-silico workflow for treating colon cancer with engineered cells: a study case.* Cristobal Hofmann, Francisco Salcedo and Martin Gutierrez

14:25-14:45 Awards and Closing Remarks

Keynote Presentation

Dr. Nicole Wheeler



Speaker Biography

Dr. Nicole Wheeler is a Turing Fellow at the University of Birmingham and also serves as a technical consultant for the Nuclear Threat Initiative. Dr Wheeler's work focuses on the development of computational screening tools for identifying DNA from emerging biological threats, establishing genomic pathogen surveillance in resource-limited settings, One Health surveillance of antimicrobial resistance, and the ethical development of artificial intelligence (AI) for health applications. She has a background in biochemistry and microbial genomics, complemented by experience in developing machine learning methods for predicting the effects of genetic variation on the virulence of pathogens. She has provided expertise on machine learning for genomic pathogen surveillance for several international programs, including a world-first AI-driven One Health AMR surveillance system. She is also actively involved in public outreach and the development of governance frameworks to ensure the safe and responsible development of technologies for health improvement.

Regular Talks

1	Using learned representations of genetic circuits to evaluate sequence-level mutations <i>Olivia Gallup and Harrison Steel</i>	13
2	Engineering Continuous Directed Evolution with Single Cell Optogenetic Selection and Microfluidics <i>Jess James, Sebastian Towers, Idris Kempf, Jingyu Wang, Jakob Foerster and Harrison Steel</i>	15
3	A neural-mechanistic hybrid approach improving the predictive power of genome-scale metabolic models <i>Bastien Mollet, Jean-Loup Faulon, Léon Faure and Wolfram Liebermeister</i>	18
4	An Automated Platform for Accelerating Adaptive Laboratory Evolution <i>Marco Corrao, Harrison Steel and Antoine Vigouroux</i>	22
5	Energy Aware Technology Mapping <i>Erik Kubaczka, Tobias Schwarz, Jérémie Marlhens, Maximilian Gehri, Nicolai Engelmann, Christian Hochberger and Heinz Koepl</i>	26
6	Design-Build-Test-Learn of Sponge RNAs for Synthetic Gene Circuits <i>Scott Stacey, Harrison Steel and Antonis Papachristodoulou</i>	30
7	Local RNA Feedback: More Logic, Less Leakage <i>Nicolai Engelmann, Maik Molderings and Heinz Koepl</i>	35
8	Rule-based generation of synthetic genetic circuits <i>Masayuki Yamamura, Ryoji Sekine, Kazuteru Miyazaki, Sota Okuda, Naoki Kodama and Daisuke Kiga</i>	39
9	A Collection of Biological Models for the Development of Infinite-State Stochastic Model Checking Tools <i>Lukas Buecherl, Payton J. Thomas, Mohammad Ahmadi, Josh Jeppson, Andrew Gerber, Eric Reiss, Chris Winstead, Hao Zheng, Zhen Zhang and Chris J. Myers</i>	43
10	Guided Design of Genetic Circuits Exploiting Stochastic Model Verification <i>Lukas Buecherl, Mohammad Ahmadi, Hao Zheng and Chris J. Myers</i>	52
11	SeqImprove: Machine Learning Assisted Curation of Genetic Circuit Sequence Information <i>Zach Sents, Duncan Britt, William Mo, Chris J. Myers and Jeanet Mante</i>	56
12	A Report on SynBio Data Management Practices <i>Carolus Vitalis, Sai Samineni, Chris Myers and Pedro Fontanarrosa</i>	60
13	Software for Synthetic Biology Workflows: How to Improve Your Productivity and Impact <i>Chris J. Myers, Lukas Buecherl, Daniel Fang, Pedro Fontanarrosa, William Mo, Sai P. Samineni, Gonzalo Vidal, Carolus Vitalis, Guillermo Yanez-Feliu, Timothy J. Rudge and Jeanet Mante</i>	65
14	DNA Editing Game (DEGA) Theory <i>Nicholas Roehner</i>	68
15	Biological Malware Detector <i>Muntaha Samad, Dan Wyschogrod and Jacob Beal</i>	71
16	Long-term evolution of bacteria for maximal growth rate <i>Antoine Vigouroux, Johan Paulsson and Sadik Yildiz</i>	73
17	Splicing-based Biocontainment Devices <i>Allison Taggart, Miles Rogers and Jacob Beal</i>	76
18	Automated model curation using LLMs: Integration of ChatGPT with the DySE framework <i>Emilee Holtzapple, Tanvi Verma and Natasa Miskov-Zivanov</i>	78
19	Knowledge-Based Pathway Extraction and Validation <i>Gaoxiang Zhou and Natasa Miskov-Zivanov</i>	81

- 20 Encoding Process Markers with Neural Networks to Simplify the Complexity of Engineering CAR T Cells
Haomiao Luo, Anya Zivanov and Natasa Miskov-Zivanov 84

Short Talks

1	DBTL Engineering cycle automation: Improving basic parts characterization in the Learn stage by Automation of the Test stage <i>Yadira Boada, Anna Pushkareva, Harold Díaz-Iza, Andrés Arboleda-García, Jesús Picó and Alejandro Vignoni</i>	88
2	Multi-Site Mutagenic Protein Library Design with Controlled Annealing Temperature <i>Yehuda Binik, Ayesha Chaudry, Akira Takada, Georgios Papamichail and Dimitris Papamichail</i>	93
3	PUDU: Simple Liquid Handling Robot Control for Synthetic Biology Workflows <i>Gonzalo Andrés Vidal Peña, Carolus Vitalis, Matt Burridge, Lukas Buecherl, David Markham, Chris Myers and Timothy Rudge</i>	96
4	Analysis of ASR inducible promoter in different conditions in Escherichia coli <i>Maria Jose Mesa-Rodriguez, Domenica Cuneo-Campodonico, Martin Gutierrez and Alberto J. Donayre-Torres</i>	99
5	How to build and train your ANN (In-Silico) From zero to hero <i>Tomás Fuentes Araya and Martín Gutiérrez</i>	103
6	AI algorithms for classification and generation of spatial/temporal patterns in cell colonies <i>Valeria Navarrete, Freddy Aguilar and Martín Gutiérrez</i>	108
7	A fully in-silico workflow for treating colon cancer with engineered cells: a study case <i>Cristobal Hofmann, Francisco Salcedo and Martin Gutierrez</i>	112
8	Using Machine Learning to Infer RNA Velocity Fields <i>Taos Transue and Payton Thomas</i>	115
9	Model Checking of Interval Discrete-Time Markov Chain for Biochemical Pathways <i>Krishnendu Ghosh</i>	119

Using learned representations of genetic circuits to evaluate sequence-level mutations

Olivia Gallup

olivia.gallupova@eng.ox.ac.uk

Department of Engineering, University of Oxford
Oxford, UK

Harrison Steel

harrison.steel@eng.ox.ac.uk

Department of Engineering, University of Oxford
Oxford, UK

1 INTRODUCTION

Evolution often presents a major obstacle to scaling and maintaining useful engineered biological systems. However, while the study of evolution has been a key focus for biology for many years, it has recently also been framed as an engineering challenge in synthetic biology [1] - something to be worked with rather than against. Just as some genetic parts function robustly despite mutations [5], others can oscillate between different behaviours over short evolutionary trajectories [8]. Controlling local evolutionary stability enables the design of systems capable of enduring unpredictable environments without compromising their function. Computational modelling of how mutation affects genetic parts is also becoming more tractable with the growing suite of tools for predicting biological interactions [4]. Notably, significant progress in protein design has demonstrated the effectiveness of data-driven methodologies in finding matches to a target specification, despite the sequence-based search space being too vast to fully explore. The embeddings learned by such models hold biologically meaningful representations and can identify key patterns corresponding to specific phenotypes.

The rise of multi-modal models [10], driven by advancements in self-supervised learning, have facilitated the combination of diverse biological data into enriched predictive embeddings. Combining embeddings is particularly relevant for automating design in synthetic biology, which heavily relies on various forms of biological abstractions, such as gene networks, biochemical networks, logic gates, or parts vs. sequence-based representations. Evaluating mutational robustness can be achieved at the genetic parts level by utilising part-specific tools like the Ribosome Binding Site (RBS) Calculator [7], sequence similarity, or neural network models for promoter design to predict how a sequence mutation might change the strength of a part. More comprehensive, part-agnostic tools can transcend the parts level to predict interaction energies or secondary structures of nucleotide sequences [6]. This development enables the design of DNA or RNA-based circuits, such as riboswitches/ribozymes, DNA computing devices, or transcriptional regulation mediators, entirely in silico. While protein interaction predictors are still in their developmental phase, DNA/RNA circuits currently

serve as the optimal model system for studying evolution [9] and the development of new artificial intelligence methods for biodesign.

2 RESULTS

Here, we explore the use of neural networks in predicting the mutational robustness of simple genetic circuits and, more broadly, in generating circuits at the sequence level based on a desired functionality. A summary can be found in the graphical abstract in Figure 1. The components of the model circuit consist of an arbitrary number of RNA sequences governed by their binding rates' interaction matrix. RNA species are generated randomly as circuits, and RNA binding simulators like RNAsstructure [6] or IntaRNA [3] determine the interaction energy between RNA molecules. The simulated interactions are subsequently evaluated as a rate to model circuit dynamics with a step signal input. We perform a simulation of a set of circuits and assess their dynamics in comparison to mutated versions of the same circuit, taking into account key signal response metrics like fold change, sensitivity, precision, or adaptation overshoot as used frequently in biodesigns. The variation in these metrics allows us to quantitatively measure the potential benefits and intensity of mutations on a specific circuit.

Initial dimensionality reduction techniques (eg. TSNE, UMAP) have demonstrated that these metrics cluster based on a circuit's interaction strengths, suggesting that more advanced dimensionality reduction could improve identification of key structures or topologies. The circuits and their dynamics metrics are thus input into a neural network trained to predict the impact of a mutation to learn pertinent embeddings, in a manner akin to how convolutional neural networks learn kernels for identifying specific object classes in images [2]. This investigation of neural networks as a tool for characterising and designing genetic circuits further underscores the need for research into architectures suitable for synthetic biology data representations. The proposed frameworks for 1) simulating RNA circuits with a signal input (https://github.com/olive004/synbio_morpher) and 2) encoding these for mutational robustness predictions that aim to optimise biological systems for both function and

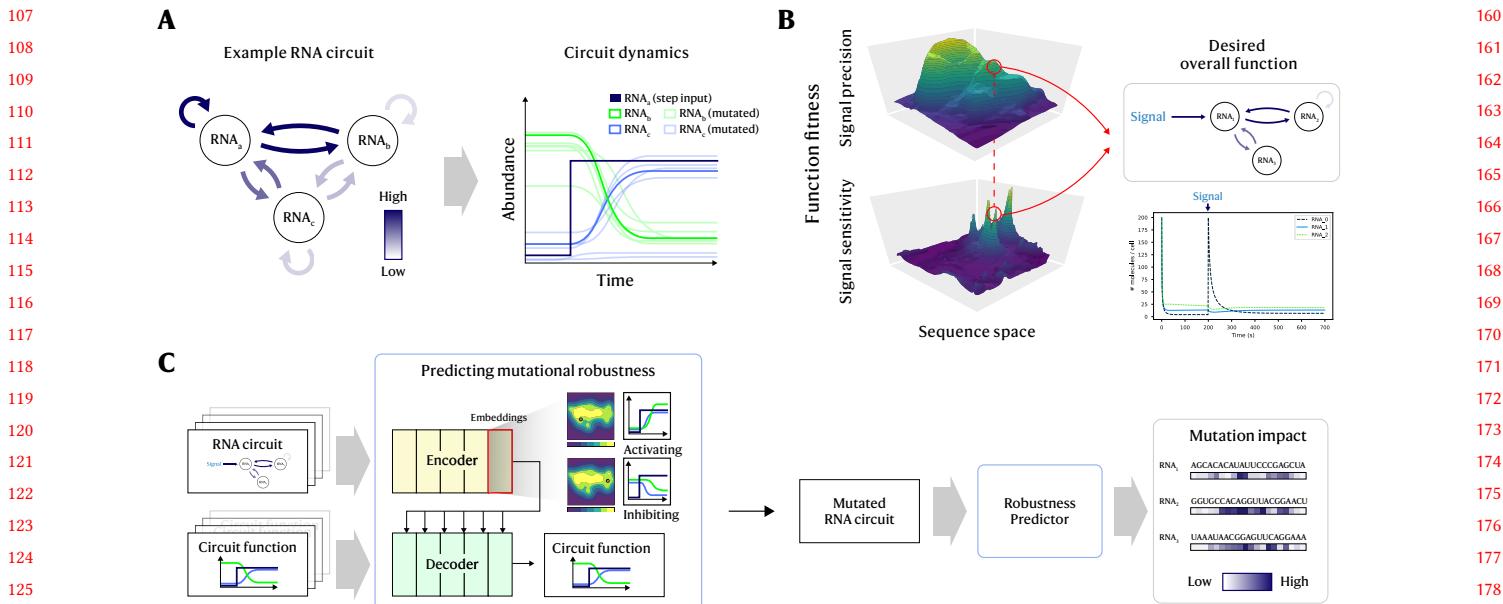


Figure 1: Project overview. A) A simple genetic circuit (RNA circuit) is used as the model for probing evolutionary robustness and simulating circuit dynamics. The RNA binding interactions and self-interactions (relative binding strength given by arrow color) parameterise the circuit's function and estimated to enable dynamics simulations after estimating reaction rates. A circuit is perturbed from steady state by adding a step input to one of the RNA species (in practice this would in the form of inducers or signalling molecules). B) An illustrative 3D landscape representing the high-dimensional sequence space vs. function. Functionality can be defined across many different circuit dynamics metrics (phenotypes), including signal precision and sensitivity, as well as overshoot, fold change, intermediate concentrations, response time, and other relevant metrics. The joint optima across the functions of interest can be mapped back to reaction rates and candidate circuits. C) Neural network for learning relevant embeddings to map from function to sequence. Circuits and their signal response are input to train a network to predict the impact of mutations on a specific circuit. Once trained, the predictor can be applied iteratively to a circuit to assess the effect of mutations along the sequence.

evolutionary stability. These findings will aid in computer-aided biodesign and take a step in the direction of integrating evolution into synthetic biology's tool arsenal. In the future, biological embeddings created with real data and tuned with prior knowledge could allow a more flexible design of new biological systems, along with more efficient screening *in silico* and faster development of applications. Computational tools will also soon become suited to studying evolution on a joint genotype and phenotype scale for systems more complex than RNA circuits and promote computational synthetic biology to the forefront of biodesign.

REFERENCES

- [1] CASTLE, S. D., GRIERSON, C. S., AND GOROCHEWSKI, T. E. Towards an engineering theory of evolution. *Nature Communications* 12, 1 (Dec. 2021), 3326.
- [2] HOFMANN, N., HASSELGREN, J., AND MUNKBERG, J. Joint Neural Denoising of Surfaces and Volumes. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6, 1 (May 2023), 1–16.
- [3] MANN, M., WRIGHT, P. R., AND BACKOFEN, R. IntaRNA 2.0: enhanced and customizable prediction of RNA-RNA interactions. *Nucleic Acids Research* 45, W1 (July 2017), W435–W439.
- [4] PASOTTI, L., AND ZUCCA, S. Advances and Computational Tools towards Predictable Design in Biological Engineering. *Computational and Mathematical Methods in Medicine* 2014 (2014), 369681.
- [5] PAYNE, J. L., AND WAGNER, A. Mechanisms of mutational robustness in transcriptional regulation. *Frontiers in Genetics* 6 (2015).
- [6] REUTER, J. S., AND MATHEWS, D. H. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics* 11, 1 (Mar. 2010), 129.
- [7] SALIS, H. M. The ribosome binding site calculator. *Methods in Enzymology* 498 (2011), 19–42.
- [8] TONNER, P. D., PRESSMAN, A., AND ROSS, D. Interpretable modeling of genotype-phenotype landscapes with state-of-the-art predictive power, June 2021. Pages: 2021.06.11.448129 Section: New Results.
- [9] WAGNER, A. Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society B: Biological Sciences* 275, 1630 (Oct. 2007), 91–100. Publisher: Royal Society.
- [10] WANG, W., BAO, H., DONG, L., BJORK, J., PENG, Z., LIU, Q., AGGARWAL, K., MOHAMMED, O. K., SINGHAL, S., SOM, S., AND WEI, F. Image as a Foreign Language: BEiT Pretraining for Vision and Vision-Language Tasks.

160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212

Engineering Continuous Directed Evolution with Single Cell Optogenetic Selection and Microfluidics

Jessica James, Sebastian Towers, Idris Kempf, Jingyu Wang, Jakob Foerster, Harrison Steel

University of Oxford, United Kingdom

jessica.james@eng.ox.ac.uk, sebastian.towers@reuben.ox.ac.uk, harrison.steel@eng.ox.ac.uk

1 INTRODUCTION

Directed evolution is a process that allows us to generate novel proteins with little to no need for rational design [1], however it is currently a labour intensive process, requiring many iterations of library generation, transformation, fitness evaluation and sequencing. As directed evolution is a process for which the outcome improves with each iteration performed, there is significant interest in reducing the manual labour required. Automation and machine learning methods have therefore been identified as key tools in the future of directed evolution [10, 11]. Advancements in molecular biology techniques, notably *in vivo* mutagenesis methods [3, 6, 8], have also eliminated the need for iterative library generation, transformation and sequencing, making automation far simpler to implement.

The directed evolution system proposed here aims to (1) leverage the advantages of *in vivo* mutagenesis to build a high-throughput, continuous method for directed evolution and (2), combine optogenetics, microfluidics and state-of-the-art microscopy to perform automated selection at the level of single cells based on long-term periods of observation (Fig 1). It will be possible to individually observe each variant of a population and determine *in silico* which members to select for the next generation. Such fine-grained control over selection opens the door to integration with more complex optimisation algorithms that steer evolution towards novel functions and designs.

2 POSSIBILITIES FOR SELECTION STRATEGIES

The standard approach to selection in directed evolution is to select the highest performing variant(s) of each generation. Given that real fitness landscapes can be quite rugged [9], this strategy is prone to getting trapped in local optima. With the new capabilities afforded by the system proposed here, we are able to explore alternative selection strategies (Fig 2). The strategies presented here are geared towards continuous directed evolution, therefore they have the additional benefit of requiring no sequencing information.

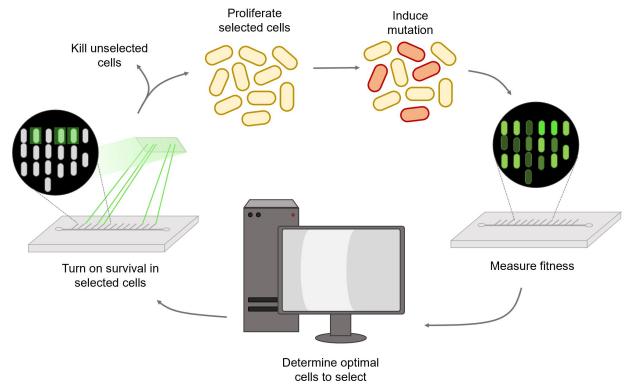


Figure 1: Directed evolution with single-cell optogenetic selection on a microfluidics chip.

Selection Functions

In this system, the fitness values of all cells are measured and then selection is applied on a cell-by-cell basis. It is therefore not limited to the standard approach of selecting cells above a defined fitness threshold, and one can apply "selection functions" to a population (i.e. probability of selection as a function of fitness). With such capabilities, it is possible to tune the degree of exploration and exploitation employed as one navigates a protein fitness landscape. In theory, such approaches could also be implemented with flow cytometry.

Population Splitting

It is possible to improve exploration of a protein fitness landscape by dividing a population into sub-populations. This is due to the fact that any single population moves around a fitness landscape in a cluster, and by splitting it into sub-populations it is possible for each cluster to explore a different trajectory. Alternatively, improvements can be made by splitting in time (i.e. favouring several short-term experiments over a single long-term experiment).

Multi-Dimensional Selection

Finally, given that using microfluidics, individual cells can be observed for long time periods and tested with multiple stimuli [7], this set up is ideal for selection across multiple dimensions (something that is currently sub-optimal with

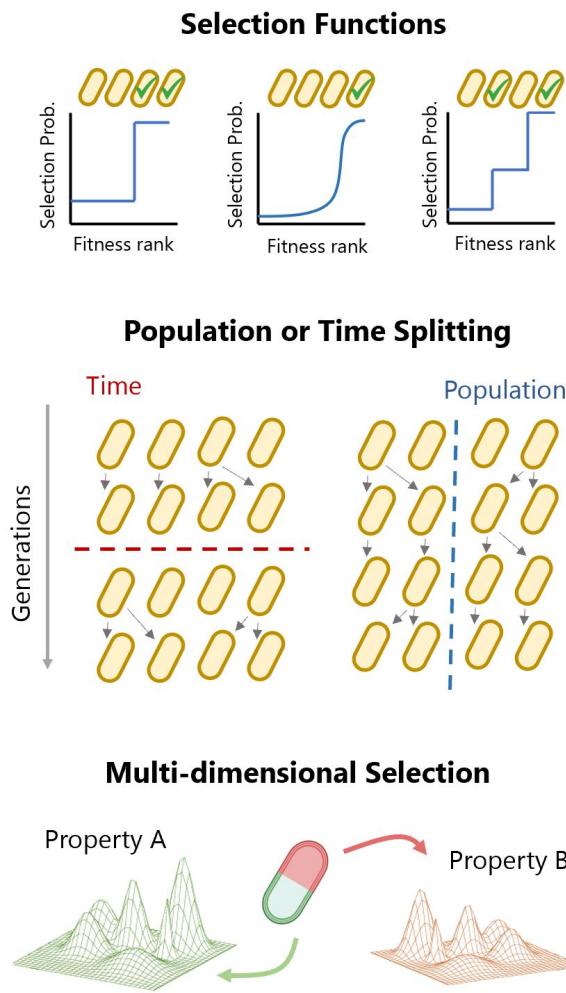


Figure 2: Selection functions are a variable way to implement selection as a function of fitness. Population or time splitting can increase effective exploration of a protein fitness landscape. Using the system proposed here, it is possible to perform selection optimally across multiple dimensions in parallel.

flow cytometry) [2]. For instance, this is highly applicable to biosensor development, where one wishes to optimise simultaneously for specificity and sensitivity.

Simulating Selection Strategies

In order to evaluate the effectiveness of these directed evolution strategies, we simulated genes as vectors with associated fitnesses *in silico*. Using either the tuneable NK fitness landscape model [4], or empirical fitness landscape data [5, 9], we performed iterative rounds of selection (according to a selection function), proliferation and mutation. An example of a combined optimisation approach using a selection

function and population splitting on the real GB1 empirical fitness landscape [9] is displayed in Figure 3.

3 CONCLUSIONS

Advances in continuous directed evolution have massively expanded the potential throughput of evolution-driven protein design. Previously, minimal attention has been paid to the selection strategies employed in continuous directed evolution. Here we propose a system that matches the cell-by-cell selection offered by flow cytometry, and improves upon it with the long-term observation benefits of microfluidics. Several challenges remain before this method can be applied to a real laboratory setting, the most notable of which are successful engineering of reversible optogenetic selection and accurate cell recognition and targeting with light beams. Once up and running, however, this method will allow us to apply more complex optimisation methods to directed evolution, increasing yields while significantly reducing the manual labour required compared to traditional sequencing-based approaches. Optimised directed evolution methods can also be used to develop novel proteins from rational design starting points, opening the door to functionalities far beyond the current repertoire of nature.

REFERENCES

- [1] ARNOLD, F. H. Design by Directed Evolution. *Accounts of Chemical Research* 31, 3 (Mar. 1998), 125–131. Publisher: American Chemical Society.
- [2] F. M. MACHADO, L., CURRIN, A., AND DIXON, N. Directed evolution of the PcaV allosteric transcription factor to generate a biosensor for aromatic aldehydes. *Journal of Biological Engineering* 13, 1 (Nov. 2019), 91.

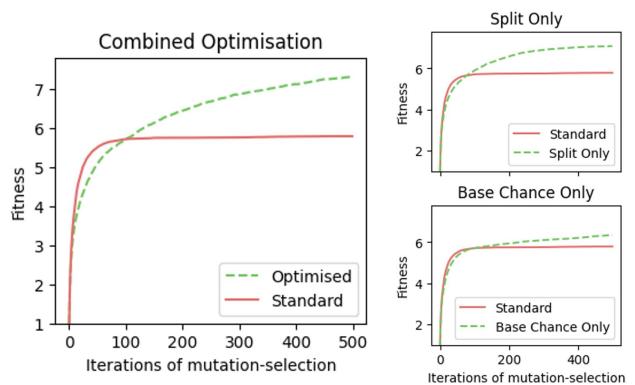


Figure 3: Comparing directed evolution strategies on the GB1 empirical landscape [9]. Standard = top 5% selection, Optimised = top 5% selection + 20% random inclusion + 5 population splits (total population size 100). Average max fitness over 1000 simulations.

- [3] HALPERIN, S. O., TOU, C. J., WONG, E. B., MODAVI, C., SCHAFFER, D. V., AND DUEBER, J. E. CRISPR-guided DNA polymerases enable diversification of all nucleotides in a tunable window. *Nature* 560, 7717 (Aug. 2018), 248–252. Number: 7717 Publisher: Nature Publishing Group.
- [4] KAUFFMAN, S., AND LEVIN, S. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology* 128, 1 (Sept. 1987), 11–45.
- [5] LITE, T.-L. V., GRANT, R. A., NOCEDAL, I., LITTLEHALE, M. L., GUO, M. S., AND LAUB, M. T. Uncovering the basis of protein-protein interaction specificity with a combinatorially complete library. *eLife* 9 (Oct. 2020), e60924. Publisher: eLife Sciences Publications, Ltd.
- [6] MENGISTE, A. A., WILSON, R. H., WEISSMAN, R. F., PAPA III, L. J., HENDEL, S. J., MOORE, C. L., BUTTY, V. L., AND SHOULDER, M. D. Expanded MutaT7 toolkit efficiently and simultaneously accesses all possible transition mutations in bacteria. *Nucleic Acids Research* (Jan. 2023), gkad003.
- [7] POTVIN-TROTTIER, L., LURO, S., AND PAULSSON, J. Microfluidics and single-cell microscopy to study stochastic processes in bacteria. *Current Opinion in Microbiology* 43 (June 2018), 186–192.
- [8] RIX, G., WATKINS-DULANEY, E. J., ALMHJELL, P. J., BOVILLE, C. E., ARNOLD, F. H., AND LIU, C. C. Scalable continuous evolution for the generation of diverse enzyme variants encompassing promiscuous activities. *Nature Communications* 11, 1 (Nov. 2020), 5644. Number: 1 Publisher: Nature Publishing Group.
- [9] WU, N. C., DAI, L., OLSON, C. A., LLOYD-SMITH, J. O., AND SUN, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife* 5 (July 2016), e16965.
- [10] WU, Z., KAN, S. B. J., LEWIS, R. D., WITTMANN, B. J., AND ARNOLD, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences* 116, 18 (Apr. 2019), 8852–8858. Company: National Academy of Sciences Distributor: National Academy of Sciences Institution: National Academy of Sciences Label: National Academy of Sciences Publisher: Proceedings of the National Academy of Sciences.
- [11] YU, T., BOOB, A. G., SINGH, N., SU, Y., AND ZHAO, H. In vitro continuous protein evolution empowered by machine learning and automation. *Cell Systems* 0, 0 (May 2023). Publisher: Elsevier.

A neural-mechanistic hybrid approach improving the predictive power of genome scale metabolic models

Léon Faure¹, Bastien Mollet², Wolfram Liebermeister³, Jean-Loup Faulon¹

¹MICALIS Institute, INRAE, AgroParisTech, University of Paris-Saclay, ²UMR MIA, INRAE, AgroParisTech, Université Paris-Saclay, ³MaIAGE, INRAE, University of Paris-Saclay
`{leon.faure,bastien.mollet,wolfram.liebermeister,jean-loup.faulon}@inrae.fr`

INTRODUCTION

Constraint-based metabolic models have been used for decades to predict the phenotype of microorganisms in different environments with significant successes. However, quantitative predictions are limited unless labor-intensive measurements of media uptake fluxes are performed. Furthermore, even with precise measurements, those models do not account for all the mechanisms that regulate the metabolism. Data-driven models could make better predictions at the expense of generalization, interpretation and a significant increase of experimental data for training. In this paper, we show how hybrid models composed of a neural network part coupled with a mechanistic part can limit the flaws of each approach. This trainable architecture provides a way to improve phenotype predictions for common tasks in system biology and metabolic engineering. For example, our models were tested on growth rate predictions of *Escherichia coli* and *Pseudomonas putida* grown in different media and on phenotype predictions of gene knocked-out *E. coli* mutants. Our neural-mechanistic models outperform constraint-based models and require training set sizes orders of magnitude smaller than classical machine learning methods.

RESULTS:

Presentation of the artificial metabolic network (AMN) models:

The data-driven part of our models was chosen to be artificial neural networks (ANN). Thus, the mechanistic part was forced to be compatible with gradient back-propagation in order to allow the ANN's training. Flux balance analysis (FBA) was chosen to be the mechanistic part of the model. However, the fastest method to solve FBA is the simplex algorithm which is incompatible with gradient back-propagation.

Back-propagation compatible FBA solvers:

Three alternative methods were proposed to solve FBA's constrained linear problem. Our first method (Wt-solver), inspired by previous work on signaling networks [1], recursively updates \mathbf{m} , the vector of metabolite production fluxes, and \mathbf{v} , the vector of all reaction fluxes using matrices derived from the metabolic network stoichiometric matrix S and from

a weight matrix, W_r , representing consensual flux branching ratios found in example flux distributions (i.e., reference FBA-simulated data or experimental measurements). The second method (LP-solver), derived from a method proposed by Yang et al.[2], handles linear problems using exact constraint bounds (EBs) or upper bounds (UBs) for uptake fluxes. This method makes use of Hopfield-like networks, which is a long-standing field of research inspired by the pioneering work of Hopfield and Tank[3]. The third approach (QP-solver), is loosely inspired by the work on Physics-Informed Neural Networks (PINNs)[4], which has been developed to solve partial differential equations matching a small set of observations. With PINNs, solutions are first approximated with a neural network and then refined to fulfill the constraints imposed by the differential equations and the boundary conditions. These methods were validated on FBA simulations obtained with Cobrapy[5] as illustrated in Fig 2f,g.

AMNs can be trained on experimental datasets with good predictive power:

To train AMNs on an experimental dataset, we grew *E. coli* DH5-alpha in 110 different media compositions, with M9 supplemented with combinations of 4 amino acids as a basis and 10 different carbon sources as possibly added nutrients. The resulting experimental dataset of media compositions: C_{med} , and measured growth rates: V_{ref} , was used to train all AMN architectures (Wt, LP, QP) as illustrated in Fig 1c. The results are presented in Fig 2a-c and the Q^2 are above 0.77 for all three methods whereas R^2 with FBA alone presented in Fig 2d was 0.51.

To demonstrate capabilities of AMNs beyond this task, we extracted from the ASAP database a dataset of 17,400 growth rates for 145 *E. coli* mutants. Each mutant had a KO of a single metabolic gene and was grown in 120 media with a different set of substrates. Our AMNs training sets were therefore composed of medium compositions and reaction KOs, both encoded as binary vectors, alongside the measured growth rates. The AMN regression performance reaches $Q^2=0.81$. For comparison, a decision tree algorithm (XGboost) yielded a regression performance of 0.75, with the same cross-validation scheme and dataset. To further showcase AMN capabilities,

in particular when multiple fluxes are measured, we evaluated the performance of an AMN on a dataset from Rijsewijk et al [6]. With this dataset, composed of 31 fluxes measured for 64 single regulator gene KO mutants of *E. coli* grown in two media compositions, our AMN reaches a variance averaged Q^2 value of 0.91 in 10-fold cross-validation.

AMNs can be used in a reservoir computing framework to enhance the predictive power of traditional FBA solvers:

In order to overcome FBA's limitation of unknown uptake fluxes, AMN models were used to predict these fluxes using a method inspired by reservoir computing that we called 'AMN-Reservoir' (Fig 1d). Once an AMN has been trained on a large dataset of FBA-simulated data where exact uptake fluxes values are known, we can fix its parameters and exploit it in subsequent further learning in order to find uptake fluxes values that can be used as entries in a classical FBA framework. We benchmarked our AMN-Reservoir approach with two datasets. One of them was produced by our team and is composed of 110 *E. coli* growth rates, and the second one stems from a growth assay of *P. putida* grown in 296 different conditions (Nogales et al.[7]). Overall, our results indicate that the usage of AMN-Reservoirs substantially increases the predictive capabilities of FBA without additional experimental work. Indeed, after applying the AMN-Reservoir procedure to find the best uptake fluxes, we increased the R on *E. coli* growth rates from 0.51 to 0.97 and increase the accuracy on *P. putida* growth assays from 0.81 to 0.96.

DISCUSSION:

Making FBA suitable for machine learning as we have done in this study opens the door to improve GEMs. For instance, in addition to estimating uptake fluxes, AMNs could be used to estimate the coefficients of the biomass reaction based on measurements. With AMNs, a trainable layer containing the biomass coefficients could be added, adapting the biomass reaction to any experimental setup. Another possible application of AMNs is to enhance GEMs reconstruction based on quantitative prediction performance. Indeed, the method we developed for KOs could be adapted to screen putative reactions in a metabolic model so that its predictions match experimental data. Manual curation must be performed after this task in order to incorporate existing literature and database. Beyond improving constraint-based mechanistic models and black-box ML models, AMNs can also be used for industrial applications. Indeed, since arbitrary objective functions can be designed and AMNs can be directly trained on experimental measurement data, AMNs can be used to optimize media for the bioproduction of compounds of interest or to find optimal gene deletion and insertion strategies,

which are typical tasks in metabolic engineering projects. In the latter case, reactions would be turned off via a trainable layer, which would be added prior to the mechanistic layers of our AMNs. Another potential application is the engineering of microorganism-based decision-making devices for the multiplexed detection of metabolic biomarkers or environmental pollutants. Here, AMNs could be used to search for internal metabolite production fluxes enabling one to differentiate positive samples containing biomarkers or pollutants from negative ones.

This document is a summary of a work published under the DOI: <https://doi.org/10.1038/s41467-023-40380-0>

REFERENCES

- [1] Avlant Nilsson, Joshua M Peters, Nikolaos Meimetis, Bryan Bryson, and Douglas A Lauffenburger. Artificial neural networks enable genome-scale simulations of intracellular signaling. *Nat. Commun.*, 13(1):3069, June 2022.
- [2] Yongqing Yang, Jinde Cao, Xianyun Xu, Manfeng Hu, and Yun Gao. A new neural network for solving quadratic programming problems with equality and inequality constraints. *Math. Comput. Simul.*, 101:103–112, July 2014.
- [3] J J Hopfield and D W Tank. “neural” computation of decisions in optimization problems. *Biol. Cybern.*, 52(3):141–152, 1985.
- [4] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through Physics-Informed neural networks: Where we are and what’s next. *J. Sci. Comput.*, 92(3):88, July 2022.
- [5] Ali Ebrahim, Joshua A Lerman, Bernhard O Palsson, and Daniel R Hyduke. COBRApy: COnstraints-Based reconstruction and analysis for python. *BMC Syst. Biol.*, 7:74, August 2013.
- [6] Bart R B Haverkorn van Rijsewijk, Annik Nanchen, Sophie Nallet, Roelco J Kleijn, and Uwe Sauer. Large-scale ^{13}C -flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in *Escherichia coli*. *Mol. Syst. Biol.*, 7:477, March 2011.
- [7] Juan Nogales, Joshua Mueller, Steinn Gudmundsson, Francisco J Canalejo, Estrella Duque, Jonathan Monk, Adam M Feist, Juan Luis Ramos, Wei Niu, and Bernhard O Palsson. High-quality genome-scale metabolic modelling of *Pseudomonas putida* highlights its broad metabolic capabilities. *Environ. Microbiol.*, 22(1):255–269, January 2020.

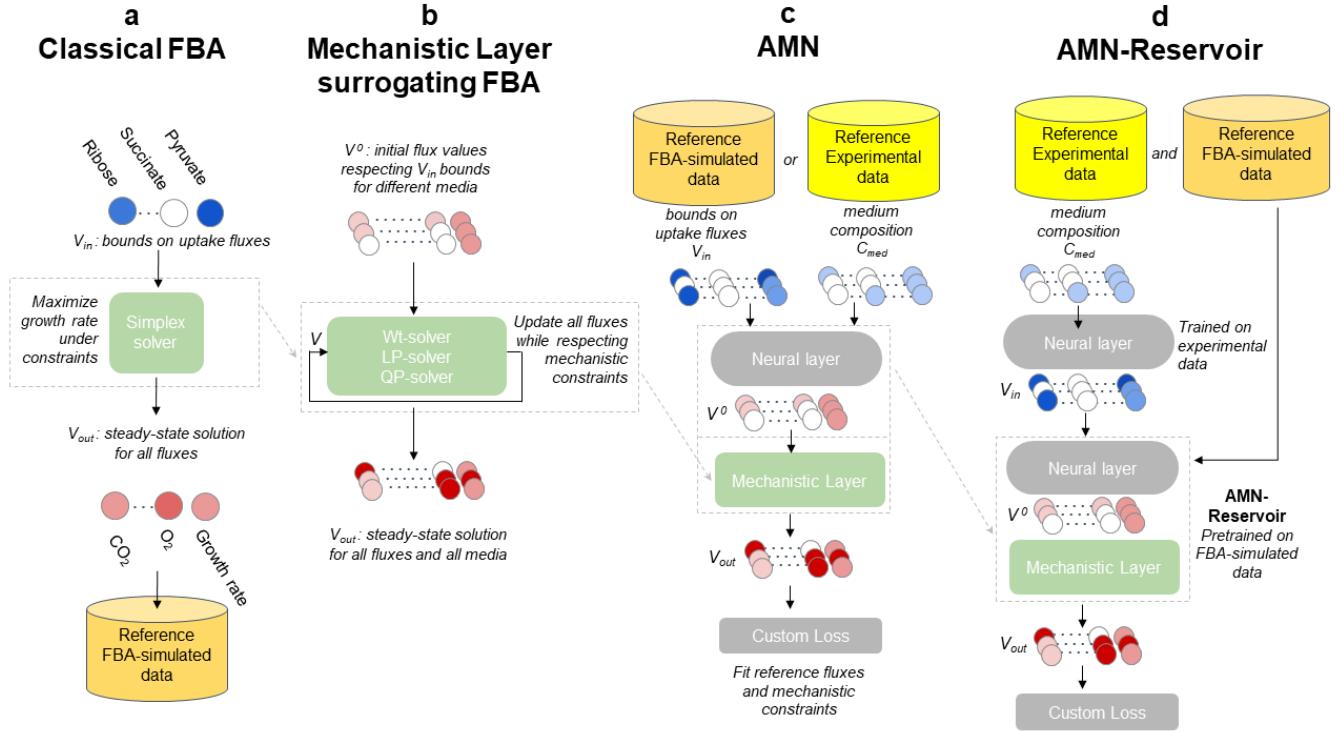


Figure 1: Computing and learning frameworks for FBA, alternative Mechanistic Models, AMN, and AMN-Reservoir. a. Computing framework for classical FBA. b. Computing framework for MM methods surrogating FBA. All three solver (Wt, LP and QP) can handle multiple growth media at once. c. Learning framework for AMN hybrid models. d. Learning framework for an ‘AMN-Reservoir’. The first step is to train an AMN on FBA-simulated data (as in panel c), after which parameters of this AMN are frozen. In the second step, a neural layer is added prior to V_{in} taking as input media compositions, C_{med} , and learning the relationship between the compositions and bounds on uptake fluxes.

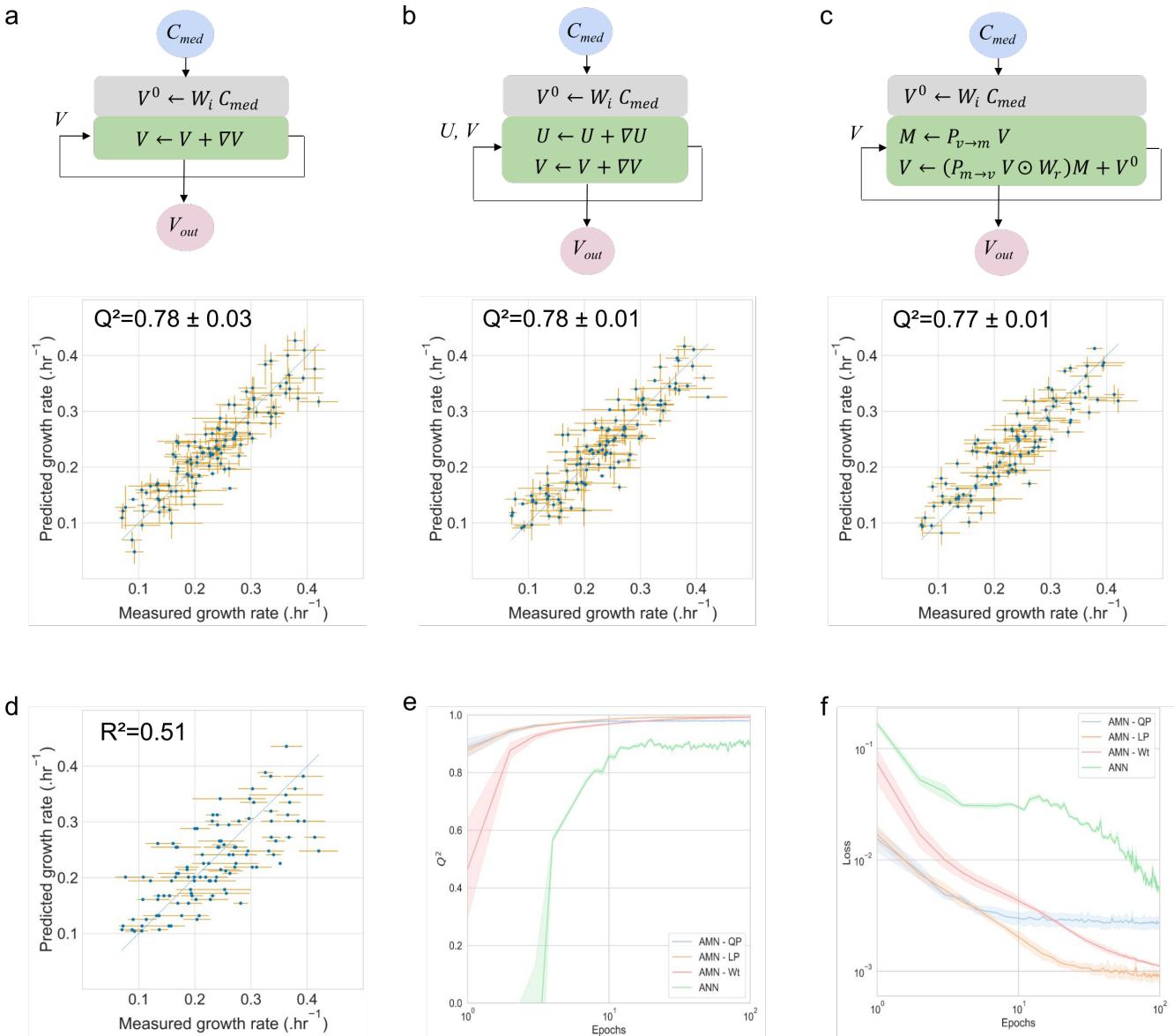


Figure 2: Benchmarking growth rate predictions by AMNs with experimental measurements. In all panels, the experimental measurements were carried out on *E. coli* grown in M9 with different combinations of carbon sources (strain DH5-alpha, model iML1515). Training and 10-fold stratified cross-validation were performed 3 times with different initial random seeds. All points plotted were compiled from predicted values obtained for each cross-validation set. In all cases, both axes show mean values (measured and predicted), and error bars denote standard deviations. For the measured data, means and standard deviations were computed based on 3 replicates, whereas for predictions, means and standard deviations were computed based on the 3 repeats of the 10-fold cross-validation. Source data are provided as a Source Data file (cf. Data availability).

a. Architecture and performance of AMN-QP. The neural layer (grey box) is composed of an input layer of size 38 (c_{med}), a hidden layer of size 500, and an output layer of size 550 corresponding to all fluxes (v) of the iML1515 reduced model. The mechanistic layer (green box) follows the neural layer and minimizes the loss between measured and predicted growth rate, as well as the losses of the metabolic network constraints. The model was trained for 1000 epochs with dropout = 0.25, batch size = 5, and the ‘Adam’ optimizer with a 10-3 learning rate.

b. Architecture and performance of AMN-LP. This model hyperparameters are identical to those of panel a. The neural layer computes the initial values for the 550 reaction fluxes (vector v), the initial values for the 1083 metabolite shadow prices (vector u) are set to zero.

c. Architecture and performance of the AMN-Wt architecture. The model hyperparameters are those of the previous panels and the size of the W_r matrix is 550x1083 (sizes of v and u vectors).

d. CobraPy predictions on experimental growth rates.

e,f. AMN performances on data simulated with CobraPy. The architecture of the models is given by Fig 1c with V_{in} as input, and a neural layer composed of one hidden layer of size 500. For all models, dropout = 0.25, batch size = 5, the optimizer is Adam, the learning rate is 10^{-3} .

An Automated Platform for Accelerating Adaptive Laboratory Evolution

Marco Corrao

Department of Engineering Science,
University of Oxford
Oxford, United Kingdom
marco.corrao@eng.ox.ac.uk

Antoine Vigouroux

Harvard Medical School
Boston, USA
antoine_vigouroux@hms.harvard.edu

Harrison Steel

Department of Engineering Science,
University of Oxford
Oxford, United Kingdom
harrison.steel@eng.ox.ac.uk

1 INTRODUCTION

Adaptive laboratory evolution (ALE) has become an extensively used tool to complement rational engineering approaches in synthetic biology [4]. In a typical ALE experiment, microbial populations are propagated under the concurrent influence of mutagenic and selective forces, thus reproducing natural selection under controlled laboratory conditions. Evolutionary methods can tackle system-level adaptation tasks, which remain challenging by targeted engineering alone, and have been successfully used to increase the fitness and robustness of synthetic strains [4]. At the same time, ALE can act as a reverse-engineering tool, where identified causal mutations are used to inform further rational design efforts [3]. Thus, evolutionary engineering will likely become a valuable tool to integrate into the synthetic biology design, build, test, and learn cycle. While serial batch cultivation remains the most widely used experimental protocol for ALE, continuous growth systems, such as chemostats and turbidostats, have shown great potential in recent years as alternative platforms, allowing for increased throughput, control and automation [4]. However, a major constraint limiting the feasibility of long-term continuous evolution is the ability of most microbial species to form biofilms that adhere to the reaction vessel wall. Strains with improved adhesion capabilities are inherently selected for in continuous culture (as they are never removed from the vessel) and their presence progressively hampers the longevity of the experiment by disturbing optical readouts, clogging pipes, or generating a secondary selective pressure prevailing over the one of experimental interest [6].

Here, we present an automated, programmable bioreactor platform designed to overcome this issue. Compared to previously developed solutions [1, 2, 7, 8], our device allows for multiple parallel replicates, precise liquid dosing, tuneable UV-induced mutagenesis and others. Based on the capabilities of this platform, an experimental framework is proposed for future implementation, where the evolution experiment is managed and optimized continuously by means of feedback control.

2 PLATFORM DESIGN

The platform is designed to maintain up to 8 parallel cultures in exponential growth for arbitrarily long periods of time. To counterselect the emergence of wall-adhering phenotypes, an alternating growth system is implemented (Figure 1). Each reactor is composed of two parallel growth vessels. The culture is grown in one of the two poles and diluted to maintain its density around a desired setpoint (turbidostat growth). After a desired number of dilutions have been performed, a fraction of the culture is transferred into the other pole, where growth is resumed. The transfer ratio (the volumetric fraction of the culture transferred over) is a design parameter that can be controlled according to need. If the transfer ratio is set equal to the dilution ratio, then the device operates as a continuous turbidostat. However, a lower transfer ratio may be desired to reduce media consumption (which, for fast-growing cultures, may be significant over a long experiment) at the cost of losing additional genetic diversity at each transfer step. Once transfer has occurred, the initial pole is sterilized with a bleach solution and rinsed with water, ensuring the removal of any wall-adhering cell. The process is thus repeated indefinitely, enabling long-term growth while maintaining reactors and pipes clean.

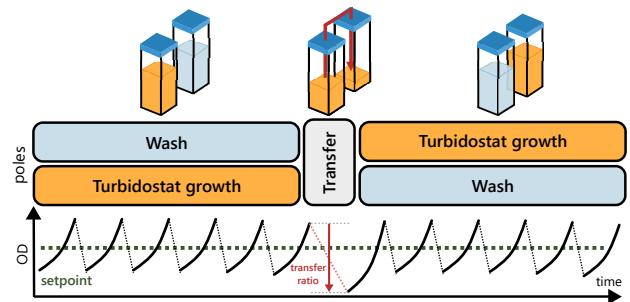


Figure 1: Counter-selection of biofilm formation by means of alternating growth/wash cycles. The culture is grown in one pole while the other is washed with bleach and water (top). While growing, the culture is diluted periodically to maintain its density around a desired setpoint (bottom). After a desired number of dilutions, a fraction of the culture (transfer ratio) is moved to the other pole and the cycle is repeated.

Liquid handling in the device is driven by pressurised air and controlled by multiple electric valves (Figure 2a). Reagents (including media, water, and bleach) are drawn from downstream reservoirs and flown through a custom-made heat exchanger (Figure 2b) that brings them to the desired temperature for the experiment. This mechanism allows keeping the reservoirs of reagents at cold temperatures, limiting, for example, degradation of growth media or antibiotics. Reagents are dosed, mixed, and optionally aerated in an intermediate mixing bottle (IMB, Figure 2b). Flow of liquid in and out of the IMB can be monitored in real time through a load cell connected to the bottle. This approach allows for precise dosing of liquids and can be used, for example, to tune media composition over the course of the experiment. Finally, liquids are sent to the reactors for dilution and cleaning.

The turbidostat module (Figure 2d) is composed of 16 (8 vessels x 2 poles) quartz cuvettes embedded into a custom-designed board. Each reactor includes an LED-photodiode pair for optical density (OD) reading, a UV LED for tuneable control of mutagenesis, as well as a rotor to enable stirring through a magnetic stir bar. Flow of liquid in (dilutions), out (waste/sampling), and across (transfers) reactors is handled through additional valves placed on the board.

Finally, an operating software interface built in Python is used to communicate with the hardware and allows for GUI-based operation and monitoring of the machine (Figure 2e). The software implements a range of basic operations, which can be combined to construct arbitrarily complex custom protocols in a streamlined fashion. The software is also responsible for controlling in real time the execution of different tasks, including media heating and liquid dosing. This is done in a closed-loop fashion based on feedback from a range of sensors present in the device, granting the system precision, reproducibility, and fault detection abilities.

3 FUTURE OUTLOOK

We designed and built an easy-to-program bioreactor tailored to the needs of laboratory evolution experiments. The platform enables a high degree of control over growth conditions, including temperature, medium composition, stirring rate, and UV irradiation. Because these factors can be tuned dynamically, our evolution system is suited to operate in a closed-loop configuration, which has been highlighted as a promising paradigm for accelerating ALE workflows [5, 9]. As a future development, we propose such an operational scheme (Figure 3), wherein growth rate estimates based on OD readings are fed back to a control algorithm. These observations are integrated with prior knowledge of the system's dynamics, which may come in the form of i) previously measured responses to environmental variables (e.g. known dose-response curves for the condition of interest) and ii) an

approximate model of adaptation dynamics for the trait of interest. This prior knowledge grants the controller predictive ability over the system's dynamics and can be used to inform its decision-making process. Based on this information, the controller determines the optimal environmental conditions required to sustain adaptation of the culture at high rate, which are finally sent to the device for actuation. In the simplest case, mutagenesis (in the form of UV irradiation) may be fixed and medium composition tuned to maintain the culture in a constant inhibition state [5, 9]. However, our platform will allow the exploration of more complex control designs, including simultaneous dynamic tuning of mutagenesis and selective pressure.

Enabled by technology like this one, automated evolutionary engineering can be used to design strains with desired levels of stress tolerance, media adaptation, or combinations thereof, and characterise the evolutionary stability of rationally designed biological parts. Such an approach has great potential for a number of applications, including bioproduction, waste remediation, and biosafety.

ACKNOWLEDGEMENTS

The authors would like to thank Sadik Yildiz and Johan Paulsson for their contribution to the development of the platform.

REFERENCES

- [1] DE CRÉCY, E., METZGAR, D., ALLEN, C., PÉNICAUD, M., LYONS, B., HANSEN, C. J., AND DE CRÉCY-LAGARD, V. Development of a novel continuous culture device for experimental evolution of bacterial populations. *Applied Microbiology and Biotechnology* 77, 2 (2007), 489–496.
- [2] ESPESO, D. R., DVOŘÁK, P., APARICIO, T., AND DE LORENZO, V. An automated DIY framework for experimental evolution of *Pseudomonas putida*. *Microbial Biotechnology* 14, 6 (2021), 2679–2685.
- [3] LACROIX, R. A., SANDBERG, T. E., O'BRIEN, E. J., UTRILLA, J., EBRAHIM, A., GUZMAN, G. I., SZUBIN, R., PALSSON, B. O., AND FEIST, A. M. Use of Adaptive Laboratory Evolution To Discover Key Mutations Enabling Rapid Growth of *Escherichia coli* K-12 MG1655 on Glucose Minimal Medium. *Applied and Environmental Microbiology* 81, 1 (2015), 17–30.
- [4] MAVROMMATI, M., DASKALAKI, A., PAPANIKOLAOU, S., AND AGGELIS, G. Adaptive laboratory evolution principles and applications in industrial biotechnology. *Biotechnology Advances* 54, April 2021 (2022), 107795.
- [5] TOPRAK, E., VERES, A., YILDIZ, S., PEDRAZA, J. M., CHAIT, R., PAULSSON, J., AND KISHONY, R. Building a morbidostat: an automated continuous-culture device for studying bacterial drug resistance under dynamically sustained drug inhibition. *Nature Protocols* 8, 3 (Mar. 2013).
- [6] VAN DEN BERGH, B., SWINGS, T., FAUVART, M., AND MICHELS, J. Experimental Design, Population Dynamics, and Diversity in Microbial Experimental Evolution.
- [7] YILDIZ, M. S. Bacterial evolution under optimal conditions. Master's thesis, Universidad de Los Andes, 2014.
- [8] YILDIZ, M. S. *Understanding the Limits on Exponential Growth of Bacteria Through Long Term Evolution*. PhD thesis, Harvard University Graduate School of Arts and Sciences, 2021.
- [9] ZHONG, Z., WONG, B. G., RAVIKUMAR, A., ARZUMANIAN, G. A., KHALIL, A. S., AND LIU, C. C. Automated Continuous Evolution of Proteins in Vivo. *ACS Synthetic Biology* 9, 6 (2020), 1270–1276.

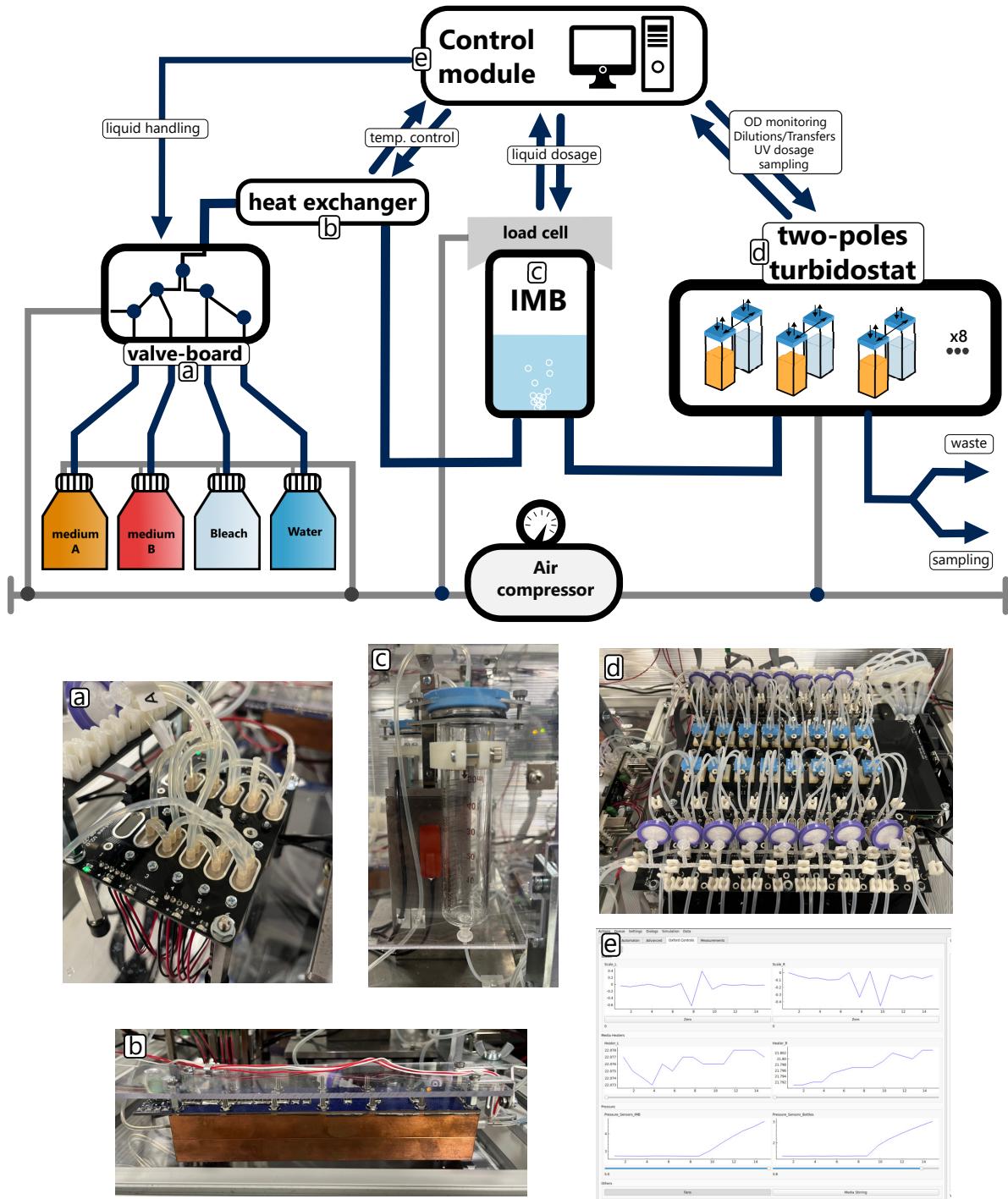


Figure 2: Schematic and photos of the platform's main components and connections, as described in the main text. The letter in each photo matches the corresponding part in the diagram. a) One of the valve-boards used to direct the flow of liquids across the device. b) The heat exchanger, used to bring cold medium to the desired temperature for the experiment. c) One of the intermediate mixing bottles (IMBs), where liquids are mixed, dosed, and aerated. d) The turbidostat module, showing the 8 pairs of reactors and the relative connections. e) A screenshot taken from the software GUI used to monitor and control the platform.

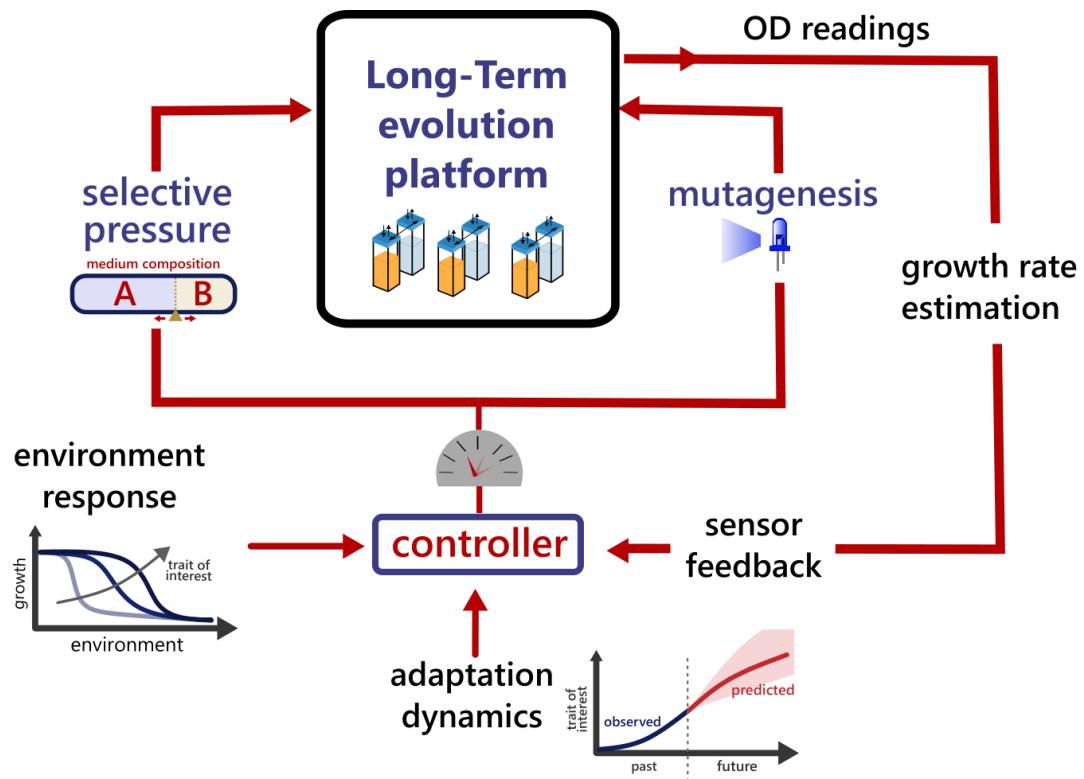


Figure 3: Proposed experimental framework for automated evolutionary engineering based on the platform. The optical density of the evolving cultures is used to estimate their growth rate in real time, which is in turn fed to a control algorithm. The controller combines observed data with prior knowledge of the system's dynamics, which can come from previously measured environmental response curves (bottom left) and/or a chosen model of adaptation dynamics for the trait of interest (bottom). Even if approximate, this prior knowledge gives the controller some predictive ability over the expected system's future trajectories, which can improve the control action. Hence, the controller computes an optimal level of mutagenesis (in the form of UV irradiation) and selective pressure (in the form of growth conditions) required to maintain a high rate of adaptation in the experiment. Finally, this output is sent to the device and actuated, closing the loop.

Energy Aware Technology Mapping

Erik Kubaczka, Tobias Schwarz, Jérémie Marlhens, Maximilian Gehri, Nicolai Engelmann,
Christian Hochberger, Heinz Koeppl

TU Darmstadt, Germany

{erik.kubaczka,jeremie.marlhens,maximilian.gehri,nicolai.engelmann,heinz.koeppl}@tu-darmstadt.de
{tobias.schwarz,hochberger}@rs.tu-darmstadt.de

1 INTRODUCTION

The advancements of modern Genetic Design Automation (GDA) tools allow for the *in silico* simulation and optimization of genetic logic circuits [4, 6, 14, 20, 24]. However, aspects like cell-to-cell variability, host context-effects or the resource competition of artificial genetic circuits with their hosts limit the expressiveness of the models considered for simulation. In addition, function energy trade-offs are known [7, 12, 15], as for example [15] reports increases in information transmission due to energy dissipation by non-equilibrium processes. While the widely employed equilibrium models [6] seem suitable for prokaryotes, they are not capable of capturing non-equilibrium characteristics which are especially relevant in the context of eukaryotes [28].

To overcome this need, we utilize non-equilibrium steady state (NESS) models of gene expression [7, 10] and derive the associated energy consumption. Our proposed model can be used with various promoter architectures, enabling the adaption to distinct settings. Used by a GDA tool, the model allows to assess both, the response characteristics as well as the associated energy consumption of the represented genetic logic gate in dependence to a given transcription factor (TF). Through its modular design, it is suitable for the *in silico* evaluation and optimization of genetic logic circuits with respect to energy and function.

While both of these are valid objectives of a technology mapping process for genetic logic circuits, the known trade-offs and the results in Figure 1 indicate exclusiveness of these optimization goals. Therefore, we further propose a multi-objective optimization approach for circuit designs, allowing for constraint-based or Pareto optimization w.r.t. both objectives. Additionally, a heuristic for the energy aware exploration of structural circuit variants is proposed.

2 NON-EQUILIBRIUM STEADY STATE GENE EXPRESSION MODEL

The consideration of energy within the *in silico* evaluation requires a model giving rise to the response characteristics as well as the associated energy consumption of a part. For this reason, we present the continuous-time Markov chain (CTMC) model of gene expression in Figure 2. This model is composed of a promoter model (Figure 2A shows an exemplary architecture), which modulates transcription, and

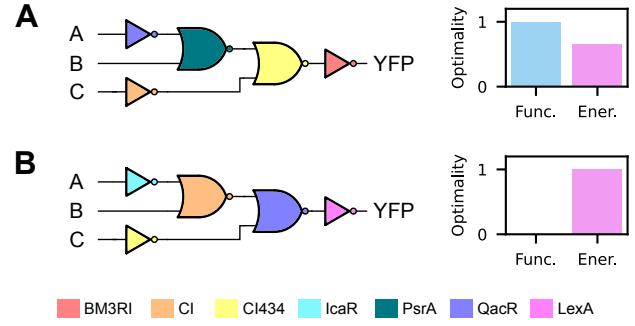


Figure 1: Function energy trade-off in the technology mapping of genetic logic circuits. It is well recognizable, that the optimization of function (Func.) in A or energy expenditure rate (Ener.) in B manifest in the sole optimization of the respective objective. In both cases, the other quantity is clearly not close to optimum, indicating a function energy trade-off. The figure presents optimality scores, where higher is better and 1 denotes the optimum, in order to facilitate comparison between the maximization of the functional score and the minimization of the energy expenditure rate. For the derivation of functional score and energy expenditure rate, we use the proposed model and the technology mapping framework ARCTIC. Both are available at <https://www.rs.tu-darmstadt.de/ARCTIC>. The model is parameterized to follow the transfer characteristic of the genetic logic gates in [4].

the dynamics of mRNA (M) and protein (P) molecules as visualized in Figure 2B.

In the following, we consider the four state promoter architecture in Figure 2A with state space $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$. The transcriptional active (x_3 and x_4) and inactive states (x_1 and x_2) are motivated by the open and closed conformation of the coding DNA segment [11]. Moreover, TF binding and unbinding appear to only modulate the rate of change between these states, rather than causing the transitions [10, 15, 16], wherefore we treat these events separately. Depending on the choice of rate constants, this architecture can model activating ($k_{23} \geq k_{14}$ and $k_{32} \leq k_{41}$) or repressing ($k_{23} \leq k_{14}$ and $k_{32} \geq k_{41}$) behaviour of the TF. By having multiple transcriptionally active and inactive states, this model allows for non-exponential waiting times between conformations, transcriptional bursting, and out-of-equilibrium simulations.

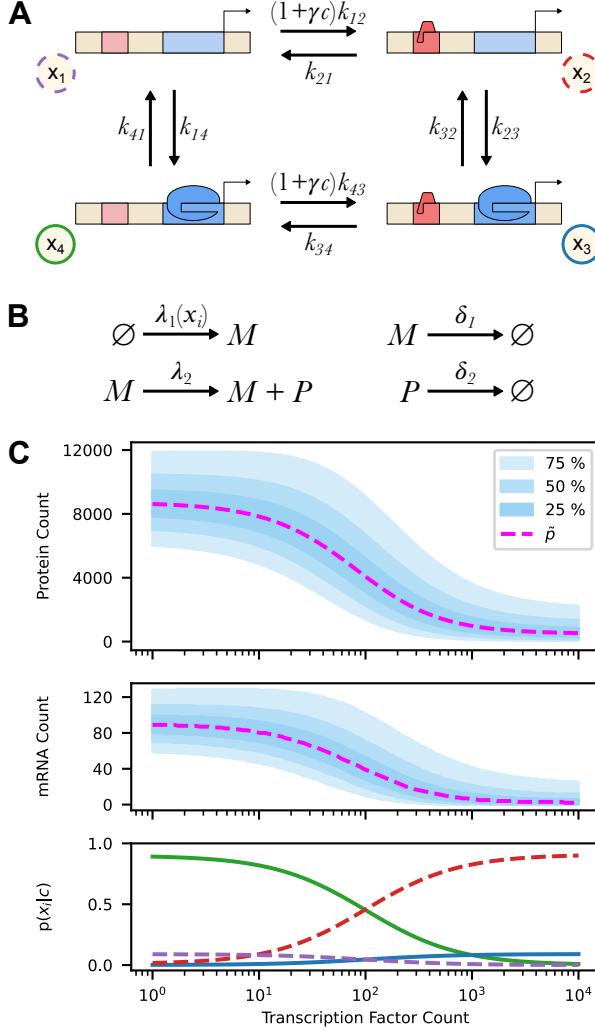


Figure 2: Visualization of the proposed gene expression model. A: The four state promoter architecture models the binding state of a TF (left vs. right) and the transcriptional activity (top vs. bottom) of the gene. The TF count c modulates the transition between bound and unbound repressor, with scaling factor γ . B: The mRNA (M) and protein (P) dynamics with promoter state x dependent transcription rate $\lambda_1(x)$, translation rate λ_2 , and mRNA and protein degradation rates δ_1 and δ_2 . C: The distribution of the protein counts, the mRNA counts, and the promoter states in dependence on the TF count c . The rates of the promoter architecture have been chosen as an inverter characteristic. In particular, we use $\gamma = 1$, the per second rates $k_{12} = 10^{-3}$, $k_{21} = 10^{-2}$, $k_{23} = 10^{-4}$, $k_{32} = 10^{-3}$, $k_{34} = 1$, $k_{41} = 10^{-3}$, $k_{43} = 10^{-5}$, $k_{14} = 10^{-4}$, $l_m = 3 l_p = 663$ nt, $l_m = 3 l_p = 663$ nt, and $\lambda_1 = 40$ nt s $^{-1}$, $\lambda_2 = 6$ aa s $^{-1}$, $\delta_1 = 5.83 \cdot 10^{-4}$ s $^{-1}$, and $\delta_2 = 2.83 \cdot 10^{-4}$ s $^{-1}$ from Table 1 for S. Cerevisiae. The colors of the promoter state probabilities refer to the state labels in Figure 2A.

With these characteristics, the promoter architecture, and in turn the overall model, is applicable to model genetic logic circuits in both, prokaryotes and eukaryotes [9, 28].

To adequately represent the response characteristic of a gate, we introduce M and P as the random variables for mRNA and protein count, with $m, p \in \mathbb{N}_{\geq 0}$ being particular realizations. The distributions of M and P depend on the TF count c , as well as the promoter state dependent transcription rate denoted by $\lambda_1(x_i)$. Moving on, the rate constants of translation, mRNA degradation, and protein degradation are λ_2 , δ_1 , and δ_2 . With $\bar{p}(x_i|c)$ being the promoter's conditional steady state marginal distribution for being in state x_i with TF count c , the mean dynamics of our proposed model read

$$E[P|c] = \frac{\lambda_2}{\delta_2} E[M|c] \quad E[M|c] = \frac{1}{\delta_1} \sum_{x_i \in \mathcal{X}} \lambda_1(x_i) \bar{p}(x_i|c)$$

with exemplary rates given in Table 1. In addition, we use the second order moments from the model to approximate the discrete distribution of protein and mRNA count with a negative binomial distribution via moment matching. Figure 2C visualizes the response characteristics of the model with the TF acting as repressor. Despite the dependence on a single TF, this model is suited for multiple inputs, as the outputs of precursors can be combined in an additive manner [4, 22].

We continue with the derivation of the expected energy consumption rate of our NESS gene expression model. This rate is composed of the energy dissipation rate of the promoter, the mRNA dynamics, and the protein dynamics. Following [15, 18], the energy dissipation rate of the promoter for maintaining the NESS is given by the entropy production rate [17, 23, 25] of the corresponding CTMC. Due to the ring structure of the four state architecture, the entropy production rate in units of $k_B T s^{-1}$ simplifies to

$$\epsilon_p(c) = (J_+(c) - J_-(c)) \log \left(\frac{J_+(c)}{J_-(c)} \right) \quad (1)$$

where k_B is the Boltzmann constant, T the temperature, and $J_+(c)$ and $J_-(c)$ denote the forward and reverse probability fluxes between two neighbouring states, respectively. In detail, the fluxes are exemplarily given by

$$J_+(c) = k_{41} \bar{p}(x_4|c) \quad J_-(c) = k_{14} \bar{p}(x_1|c)$$

where is $\bar{p}(x_i|c)$ the steady state distribution of the promoter.

To compute the energy consumption rate of transcription and mRNA degradation, we take the product of the energy per reaction and the mean reaction rate. Due to being in the steady state, the mRNA synthesis and degradation rates are actually equal and given by $\delta_1 E[M|c]$. Introducing l_m as the length in nucleotides of the mRNA, the energy per mRNA for synthesis and degradation is of the form $e_m l_m + \bar{e}_m$, where e_m is the length-dependent and \bar{e}_m the length-invariant energy demand [1, 8]. The energy expenditure

Table 1: Published estimates of transcription, translation, degradation, and energy utilization Rates. The actual transcription and translation rates depend on the organisms, as well as the protein to produce. Therefore, we include the length of relevant coding sequences [4, 22], as well as the length-dependent transcription and translation rates for E. Coli and S. Cerevisiae as exemplary model organisms for prokaryotes and eukaryotes, respectively [1–3, 5, 13, 19, 21, 27]. For a protein of l_p amino acids and a corresponding mRNA of $l_m = 3 l_p$ nucleotides, the actual rates per mRNA and protein are given by $\lambda_1 = \tilde{\lambda}_1 / l_m$ and $\lambda_2 = \tilde{\lambda}_2 / l_p$. Additionally, we acknowledge that the constant component for mRNA transcription could also be applied to the energy required for transcribing the non-coding regions of DNA to mRNA, including caps and other modifications.

Organism	$l_m = 3 l_p [nt]$	mRNA				Protein			
		$\tilde{\lambda}_1 \left[\frac{nt}{s} \right]$	$\delta_1 \left[\frac{10^{-4}}{s} \right]$	$e_m \left[\frac{k_B T}{nt} \right]$	$\bar{e}_m [k_B T]$	$\tilde{\lambda}_2 \left[\frac{aa}{s} \right]$	$\delta_2 \left[\frac{10^{-4}}{s} \right]$	$e_p \left[\frac{k_B T}{aa} \right]$	$\bar{e}_p [k_B T]$
E. Coli	561 - 714	10 - 100	10 - 60	16	≈ 0	10-20	≈ 3	29	39
S. Cerevisiae	588 - 741	30 - 320	6 - 12	16	≈ 0	3 - 10	≈ 2	42	52

of the protein dynamics follows the same rationals except for using l_p as the protein's length in amino acids. Using $n_m(c) = E[M|c]$ and $n_p(c) = E[P|c]$, the TF dependent expected energy consumption rates of transcription ($\epsilon_{tx}(c)$) and translation ($\epsilon_{tl}(c)$) are

$$\epsilon_{tx}(c) = (e_m l_m + \bar{e}_m) \delta_1 n_m(c) \quad \epsilon_{tl}(c) = (e_p l_p + \bar{e}_p) \delta_2 n_p(c)$$

and we define the total energy expenditure rate in units of $k_B T s^{-1}$ as

$$\epsilon_g(c) = \epsilon_p(c) + \epsilon_{tx}(c) + \epsilon_{tl}(c). \quad (2)$$

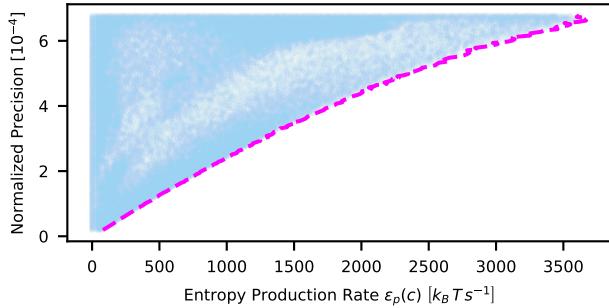


Figure 3: Results of the boundary exploration in the entropy production rate vs. normalized precision space for our gene expression model as shown in Figure 2. It is well observable, that increased entropy production corresponds to an increase in the minimum precision. The entropy production rate is calculated according to Equation (1) and the normalized precision follows the approach of [15] adapted to a repressor architecture, while the precision itself is given by the inverse of the standard deviation. For the exploration, we consider the value range $[10^{-3}, 10^3]$ for all k_{ij} as well as for γ . Additionally, we enforce $k_{23} \leq k_{14}$ and $k_{32} \geq k_{41}$ to constrain the TF to act as a repressor. For the mRNA and protein dynamics, we use the same values as in Figure 2, which are $l_m = 3 l_p = 663 \text{ nt}$ and $\tilde{\lambda}_1 = 40 \text{ nt s}^{-1}$, $\tilde{\lambda}_2 = 6 \text{ aa s}^{-1}$, $\delta_1 = 5.83 \cdot 10^{-4} \text{ s}^{-1}$, and $\delta_2 = 2.83 \cdot 10^{-4} \text{ s}^{-1}$ from Table 1 for S. Cerevisiae.

Modelling genetic logic gates g in a genetic circuit G with the proposed model, the average and maximum expected energy consumption rates E and E_{max} of the circuit are

$$E = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} E_i \quad E_{max} = \max_{i \in \mathcal{I}} E_i \quad E_i = \sum_{g \in G} \epsilon_g(c_{gi}) \quad (3)$$

with \mathcal{I} being the set of input combinations and i a particular input, for which E_i is the energy consumption rate and c_{gi} the TF input for gate g . Depending on the type of application and the constraints enforced, both, E and E_{max} , are valid objective functions for an optimization focusing on energy minimization.

To give an intuition on the interplay between energy dissipation rate and function, Figure 3A presents the results of a boundary exploration in the entropy production rate and normalized precision space. The proposed model is implemented in python and part of the technology mapping framework ARCTIC (<https://www.rs.tu-darmstadt.de/ARCTIC>).

3 TECHNOLOGY MAPPING

To allow for an energy aware technology mapping, we propose integrating the model into an existing simulated annealing gate assignment scheme [24]. It efficiently generates a mapping of genetic gates from a library to the circuit topology by leveraging functional proximity of gates in its neighborhood-generation heuristic. By optimizing for the E-Score [24], population-wide separation of Boolean output states of the circuit is achieved, taking cell-to-cell variability into account. We propose three objective configurations to handle the arising trade-off of function and energy. First, optimization of the E-Score under an energy threshold to find the best functioning circuit with constrained energy. Second, minimizing the energy while demanding a lower E-Score threshold to find the most energy efficient circuit that still functions. Third, a true multi-objective optimization to explore the Pareto set [26]. Exemplary results for function maximization and energy expenditure rate minimization are presented in Figures 1A and 1B, respectively.

To broaden the search space of energy efficient circuit designs, we propose to systematically take into account structural variants [24] of circuit topologies. As the number of variants and thus mapping effort may be large depending on the gate library and desired circuit function, an order of exploration of topologies is proposed. It prefers structures which feature less Boolean 1 states among all signals of the circuit with respect to all Boolean input assignments, representing a heuristic measure for the expected energy.

4 CONCLUSION

Within this work, we propose a model for non-equilibrium steady state gene expression. This model is capable of predicting the steady state distribution of protein counts as well as the associated energy consumption rate in dependence to a TF acting as input. Allowing for different promoter architectures, it is possible to adapt this model to various settings. Examples are multiple bindings sites for TFs, multiple TF species with differing binding characteristics, and transcriptionally active states with varying transcription rates. Consequently, the adaption to different organisms or different levels of coarse-graining is possible. The usage of our presented model for the *in silico* evaluation of genetic logic circuits empowers energy aware technology mapping including the distinct and joint optimization of function and energy expenditure. To this end, we present means to combine these two optimization goals either by constrained optimization or multi-objective optimization and present an approach to increase the search space exploration efficiency with respect to structural variants.

REFERENCES

- [1] ALBERTS, B., JOHNSON, A., LEWIS, J., RAFF, M., ROBERTS, K., AND WALTER, P. *Molecular Biology of the Cell*, 6th ed. Garland Science, New York, NY, 2017.
- [2] BERNSTEIN, J. A., KHODURSKY, A. B., LIN, P.-H., LIN-CHAO, S., AND COHEN, S. N. Global analysis of mRNA decay and abundance in Escherichia coli at single-gene resolution using two-color fluorescent DNA microarrays. *Proceedings of the National Academy of Sciences* 99, 15 (2002), 9697–9702.
- [3] CAO, Z., AND GRIMA, R. Analytical distributions for detailed models of stochastic gene expression in eukaryotic cells. *Proceedings of the National Academy of Sciences* 117, 9 (2020), 4682–4692.
- [4] CHEN, Y., ZHANG, S., YOUNG, E. M., JONES, T. S., DENSMORE, D., AND VOIGT, C. A. Genetic circuit design automation for yeast. *Nature Microbiology* 5, 11 (Nov 2020), 1349–1360.
- [5] COULON, A., FERGUSON, M. L., DE TURRIS, V., PALANGAT, M., CHOW, C. C., AND LARSON, D. R. Kinetic competition during the transcription cycle results in stochastic RNA processing. *eLife* 3 (oct 2014), e03939.
- [6] ENGELMANN, N., SCHWARZ, T., KUBACZKA, E., HOCHBERGER, C., AND KOEPPPL, H. Context-Aware Technology Mapping in Genetic Design Automation. *ACS Synthetic Biology* 12, 2 (Feb 2023), 446–459.
- [7] ESTRADA, J., WONG, F., DEPACE, A., AND GUNAWARDENA, J. Information Integration and Energy Expenditure in Gene Regulation. *Cell* 166, 1 (2016), 234–244.
- [8] GAO, F., DANSON, A. E., YE, F., JOVANOVIC, M., BUCK, M., AND ZHANG, X. Bacterial Enhancer Binding Proteins-AAA+ Proteins in Transcription Activation. *Biomolecules* 10, 3 (Feb. 2020), 351.
- [9] GOLDING, I., PAULSSON, J., ZAWILSKI, S. M., AND COX, E. C. Real-Time Kinetics of Gene Activity in Individual Bacteria. *Cell* 123, 6 (Dec. 2005), 1025–1036.
- [10] GRAH, R., ZOLLER, B., AND TKAČIK, G. Nonequilibrium models of optimal enhancer function. *Proceedings of the National Academy of Sciences* 117, 50 (2020), 31614–31622.
- [11] HASENAUER, J., WOLF, V., KAZEROONIAN, A., AND THEIS, F. J. Method of conditional moments (MCM) for the Chemical Master Equation. *Journal of Mathematical Biology* 69, 3 (Sep 2014), 687–735.
- [12] HAUSNER, J., MAYO, A., KEREN, L., AND ALON, U. Central dogma rates and the trade-off between precision and economy in gene expression. *Nature Communications* 10, 1 (Jan 2019), 68.
- [13] HOUSELEY, J., AND TOLLERVEY, D. The Many Pathways of RNA Degradation. *Cell* 136, 4 (Feb. 2009), 763–776.
- [14] JONES, T. S., OLIVEIRA, S. M. D., MYERS, C. J., VOIGT, C. A., AND DENSMORE, D. Genetic circuit design automation with Cello 2.0. *Nature Protocols* 17, 4 (Apr 2022), 1097–1113.
- [15] LAMMERS, N. C., FLAMHOLZ, A. I., AND GARCIA, H. G. Competing constraints shape the nonequilibrium limits of cellular decision-making. *Proceedings of the National Academy of Sciences* 120, 10 (2023), e2211203120.
- [16] LAMMERS, N. C., KIM, Y. J., ZHAO, J., AND GARCIA, H. G. A matter of time: Using dynamics and theory to uncover mechanisms of transcriptional bursting. *Current Opinion in Cell Biology* 67 (2020), 147–157. Differentiation and disease.
- [17] LEBOWITZ, J. L., AND SPOHN, H. A Gallavotti–Cohen-Type Symmetry in the Large Deviation Functional for Stochastic Dynamics. *Journal of Statistical Physics* 95, 1 (Apr 1999), 333–365.
- [18] MEHTA, P., AND SCHWAB, D. J. Energetic costs of cellular computation. *Proceedings of the National Academy of Sciences* 109, 44 (2012), 17978–17982.
- [19] MILO, R., AND PHILLIPS, R. *Cell Biology by the Numbers*. Garland Science, Dec. 2015.
- [20] 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 (07 2009), 2848–2849.
- [21] NATH, K., AND KOCH, A. L. Protein Degradation in Escherichia coli: I. MEASUREMENT OF RAPIDLY AND SLOWLY DECAYING COMPONENTS. *Journal of Biological Chemistry* 245, 11 (1970), 2889–2900.
- [22] 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.
- [23] QIAN, H., AND GE, H. *Stochastic Chemical Reaction Systems in Biology*. Springer International Publishing, 2021.
- [24] SCHLADT, T., ENGELMANN, N., KUBACZKA, E., HOCHBERGER, C., AND KOEPPPL, H. Automated Design of Robust Genetic Circuits: Structural Variants and Parameter Uncertainty. *ACS Synthetic Biology* 10, 12 (Dec 2021), 3316–3329.
- [25] SCHNAKENBERG, J. Network theory of microscopic and macroscopic behavior of master equation systems. *Rev. Mod. Phys.* 48 (Oct 1976), 571–585.
- [26] SERAFINI, P. Simulated Annealing for Multi Objective Optimization Problems. In *Multiple Criteria Decision Making* (New York, NY, 1994), Springer New York, pp. 283–292.
- [27] WATSON, J. D., BAKER, T. A., BELL, S. P., GANN, A. A. F., LEVINE, M., AND LOSICK, R. M. *Molecular Biology of the Gene*, 7th ed. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, USA, 2013.
- [28] ZOLLER, B., GREGOR, T., AND TKAČIK, G. Eukaryotic gene regulation at equilibrium, or non? *Current Opinion in Systems Biology* 31 (2022), 100435.

Design-Build-Test-Learn of Sponge RNAs for Synthetic Gene Circuits

Scott Stacey

University of Oxford

Oxford, United Kingdom

scott.stacey@eng.ox.ac.uk

Harrison Steel

University of Oxford

Oxford, United Kingdom

harrison.steel@eng.ox.ac.uk

Antonis Papachristodoulou

University of Oxford

Oxford, United Kingdom

antonis@eng.ox.ac.uk

1 INTRODUCTION

The field of synthetic biology holds great potential for revolutionising life sciences research, tackling global challenges and advancing the bioeconomy. However, synthetic biology is impeded from fulfilling this potential due to several key outstanding challenges. Among these are the challenges of unpredictability of engineering biology, context-dependence, cross-talk, noise, and host burden from synthetic circuits. Bio-Design Automation tools can play a crucial role in addressing some of these challenges and recent successes have included automated gene circuit design with CelloCAD [5]; and automated strain engineering for metabolic engineering [7]. However, despite the significant success of Bio-Design Automation tools, a lack of synthetic biological parts holds back much of synthetic biology including Bio-Design Automation.

Sponge RNAs (spRNAs) [1] are a class of sRNA that regulate gene expression through interacting with other sRNAs, rather than interacting with mRNAs. Discovered spRNAs are typically involved in important cellular processes such as regulating stress response, regulating central metabolism and global gene regulation. Yet, despite the apparent importance of spRNAs, spRNA mechanisms and the features they endow on gene regulatory networks are underexplored. As a result, spRNAs have not been used in synthetic gene circuits.

In this project we take a Design-Build-Test-Learn approach to develop spRNAs as a new component of the synthetic biology toolbox. This will elucidate natural spRNA functions, enable novel synthetic gene circuit design, and clarify sRNA and spRNA design rules for potential future Bio-Design Automation.

2 DESIGN: SYNTHETIC SRNA AND SPONGE RNA

For synthetic spRNA design, we chose a natural cognate sRNA-spRNA pair as a template. The natural MicC sRNA has been the scaffold for most synthetic sRNAs developed so far, however since the MicC sRNA does not have an identified spRNA, a novel scaffold for a synthetic sRNA based on a natural sRNA with a natural cognate sponge was identified.

The *E. coli* ChiX sRNA - ChbBC spRNA pair [6] was chosen for several reasons: the ChiX-ChbBC system is relatively well-studied compared to other sRNA-spRNA pairs; ChiX is known to bind efficiently to Hfq [2] allowing for strong knockdown; and the ChbBC sponge is part of the intergenic region of an operonic mRNA [6], so a synthetic ChbBC sponge might be amenable to being placed upstream of a fluorescent protein for easy characterisation.

The ChiX sRNA seed region was replaced with a sequence designed to target EGFP, the remainder of the natural sequence was maintained. Initially ten versions of synthetic ChiX (Fig. 1A) were designed with seed lengths of 18 to 25 nucleotides long targeting either upstream, downstream or at the RBS. Upon successful design of a synthetic ChiX, a cognate ChbBC-derived spRNA was designed. The seed region was replaced with the reverse complement of the ChiX seed region while maintaining naturally occurring mismatches, the rest of the natural sequence was maintained (Fig. 1B).

3 BUILD: SYNTHETIC SPONGE RNA CIRCUIT

We assembled a basic synthetic gene circuit using EcoFlex MoClo [4], with three inducible promoters controlling the expression of EGFP, Synthetic ChiX20D, and Synthetic ChbBC20D-mScarlet-I (FIG. 1D). The three promoters were chosen from the Marionette collection and the circuit was transformed into Marionette-Clo [3]. Additionally, other circuits were assembled including P_{VanCC}-EGFP-P_{LuxB}-Synthetic-ChiX20D, and level 1 transcription units consisting of one of the three promoters (P_{VanCC}, P_{LuxB}, P_{Tac}), RiboJ, BBa_B0064, EGFP, and L3S2P21 (Fig. 1C).

4 TEST: AUTOMATED CHARACTERISATION

We characterised the circuits using the Chi.Bio automated experimental platform [8]. Cultures were grown in Chi.Bio reactors in M9 minimal media up to 0.5 OD, at which point the Chi.Bio's turbidostat functionality maintained the cultures dithering around 0.5 OD (Fig. 2A). Dithering allows sustained temporal measurement of growth rate. Maintaining OD at 0.5 OD allows measurement of fluorescent proteins at a constant OD, which in turn allows accurate measurement of fluorescent protein dynamics and steady states. Finally

at time points during experiments, chemical inducers were added allowing characterisation of promoter activity and validation of sRNA and spRNA activity.

The three promoters were characterised by growing the three EGFP level 1 transcription units in Marionette-Clo in Chi.Bio and inducing gene expression through step changes in concentration of inducer (Fig. 2B-D). P_{VanCC}-EGFP-P_{LuxB}-Synthetic-ChiX20D was used to validate the sRNA, by inducing EGFP expression followed by inducing sRNA expression, knocking down EGFP expression (Fig. 2F). Finally, similar experiments were carried out with the synthetic spRNA circuit, where each transcription unit in the circuit was induced in turn, validating that the spRNA could recover EGFP expression after sRNA-mediated knockdown (Fig 2E).

5 LEARN: MODEL FITTING VIA OPTIMISATION

Mathematical models were fitted to explore the mechanisms and features of our synthetic spRNA circuit. Mature fluorescent protein counts per cell were estimated using measured fluorescence values. Unknown parameters were estimated using MATLAB R2023a's fmincon and globalsearch functions with a normal mean squared error cost function:

$$\text{Normal MSE} = \frac{1}{n} \sum_{i=1}^n \frac{(y_{\text{exp}i} - y_{\text{model}i})^2}{y_{\text{model}i}}$$

First, unknown parameters for the model describing EGFP expression from the three level 1 TUs were estimated:

$$\begin{aligned} \frac{dm}{dt} &= k_0 + (k_1 - k_0) \frac{I^n}{I^n + K^n} - (\delta_m + \mu)m \\ \frac{dp}{dt} &= \theta m - \mu p - \tau p \quad \frac{dg}{dt} = \tau p - \mu g \end{aligned}$$

This provided parameters which fit the data well (Fig 3A-C). Next three different models for the synthetic spRNA circuit were fit using the previously fitted parameters and fitting the new sRNA and spRNA specific parameters: 1. sRNA-mRNA and spRNA-sRNA annihilate 2. sRNA predares on mRNA and spRNA predares on sRNA and 3. sRNA-mRNA annihilate and spRNA predares on sRNA (Fig. 3D). The equations for model 2 are shown:

$$\begin{aligned} \frac{dm}{dt} &= k_0 + (k_1 - k_0) \frac{V^{n_o}}{V^{n_o} + K_v^{n_o}} - (\delta_m + \mu)m - \eta_2 sm \\ \frac{dp}{dt} &= \theta m - \mu p - \tau p \quad \frac{dg}{dt} = \tau p - \mu g \\ \frac{ds}{dt} &= k_2 + (k_3 - k_2) \frac{O^{n_o}}{O^{n_o} + K_O^{n_o}} - (\delta_s + \mu)s - \eta_2 sm - \eta_1 sz + \eta_{-2} c_2 \\ \frac{dz}{dt} &= k_4 + (k_5 - k_4) \frac{I^{n_I}}{I^{n_I} + K_I^{n_I}} - (\delta_z + \mu)z - \eta_1 sz + \eta_{-1} c_1 \\ \frac{dp_z}{dt} &= \theta z - \mu p_z - \tau_z p_z \quad \frac{dg_z}{dt} = \tau_z p_z - \mu g_z \end{aligned}$$

$$\frac{dc_1}{dt} = \eta_1 sz - \eta_{-1} c_1 - (\delta_s + \mu)c_1 \quad \frac{dc_2}{dt} = \eta_2 sm - \eta_{-2} c_2 - (\delta_s + \mu)c_2$$

Each model fits the experimental data well (Fig. 3E). However, they give qualitatively different predictions under specific experimental conditions (Fig. 3F), enabling experimental model invalidation and elucidation of possible mechanisms.

6 FUTURE DIRECTIONS

We are adapting absolute protein quantification methods for Chi.Bios, we will present data on this at the workshop. Next, we will use the models as guides for experimental investigations to (in)validate hypothesised sRNA and spRNA mechanisms, completing the first DBTL cycle iteration. Upon better understanding of the mechanism of the synthetic spRNA circuit, a new phase of Design will begin where modelling can be used to inform the Design of synthetic gene circuits such as feedback circuits or logic gates. Alternatively, *in silico*/cybergenetic control could be used to close the loop on the synthetic spRNA circuit or feedback circuit architectures could be designed and validated in an automated fashion using *in silico* control before biomolecular implementation. Multi-target spRNAs will also be investigated.

7 ACKNOWLEDGEMENTS

This work was funded by the Biotechnology and Biological Sciences Research Council [Grant No. BB/T008784/1].

REFERENCES

- [1] DENHAM, E. L. The Sponge RNAs of bacteria – How to find them and their role in regulating the post-transcriptional network. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* 1863, 8 (Aug. 2020), 194565.
- [2] MAŁĘCKA, E. M., STRÓŻECKA, J., SOBAŃSKA, D., AND OLEJNICZAK, M. Structure of bacterial regulatory RNAs determines their performance in competition for the chaperone protein Hfq. *Biochemistry* 54, 5 (Feb. 2015), 1157–1170.
- [3] MEYER, A. J., SEGALL-SHAPIRO, T. H., GLASSEY, E., ZHANG, J., AND VOIGT, C. A. Escherichia coli “Marionette” strains with 12 highly optimized small-molecule sensors. *Nature Chemical Biology* 15, 2 (Feb. 2019), 196–204.
- [4] MOORE, S. J., LAI, H.-E., KELWICK, R. J. R., CHEE, S. M., BELL, D. J., POLIZZI, K. M., AND FREEMONT, P. S. EcoFlex: A Multifunctional MoClo Kit for *E. coli* Synthetic Biology. *ACS Synthetic Biology* 5, 10 (Oct. 2016), 1059–1069.
- [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 (Apr. 2016).
- [6] OVERGAARD, M., JOHANSEN, J., MØLLER-JENSEN, J., AND VALENTIN-HANSEN, P. Switching off small RNA regulation with trap-mRNA. *Molecular Microbiology* 73, 5 (Sept. 2009), 790–800.
- [7] SINGH, A. H., ET AL. An Automated Scientist to Design and Optimize Microbial Strains for the Industrial Production of Small Molecules, Jan. 2023. *bioRxiv*. Section: New Results.
- [8] STEEL, H., HABGOOD, R., KELLY, C. L., AND PAPACHRISTODOULOU, A. In situ characterisation and manipulation of biological systems with Chi.Bio. *PLOS Biology* 18, 7 (July 2020), e3000794.

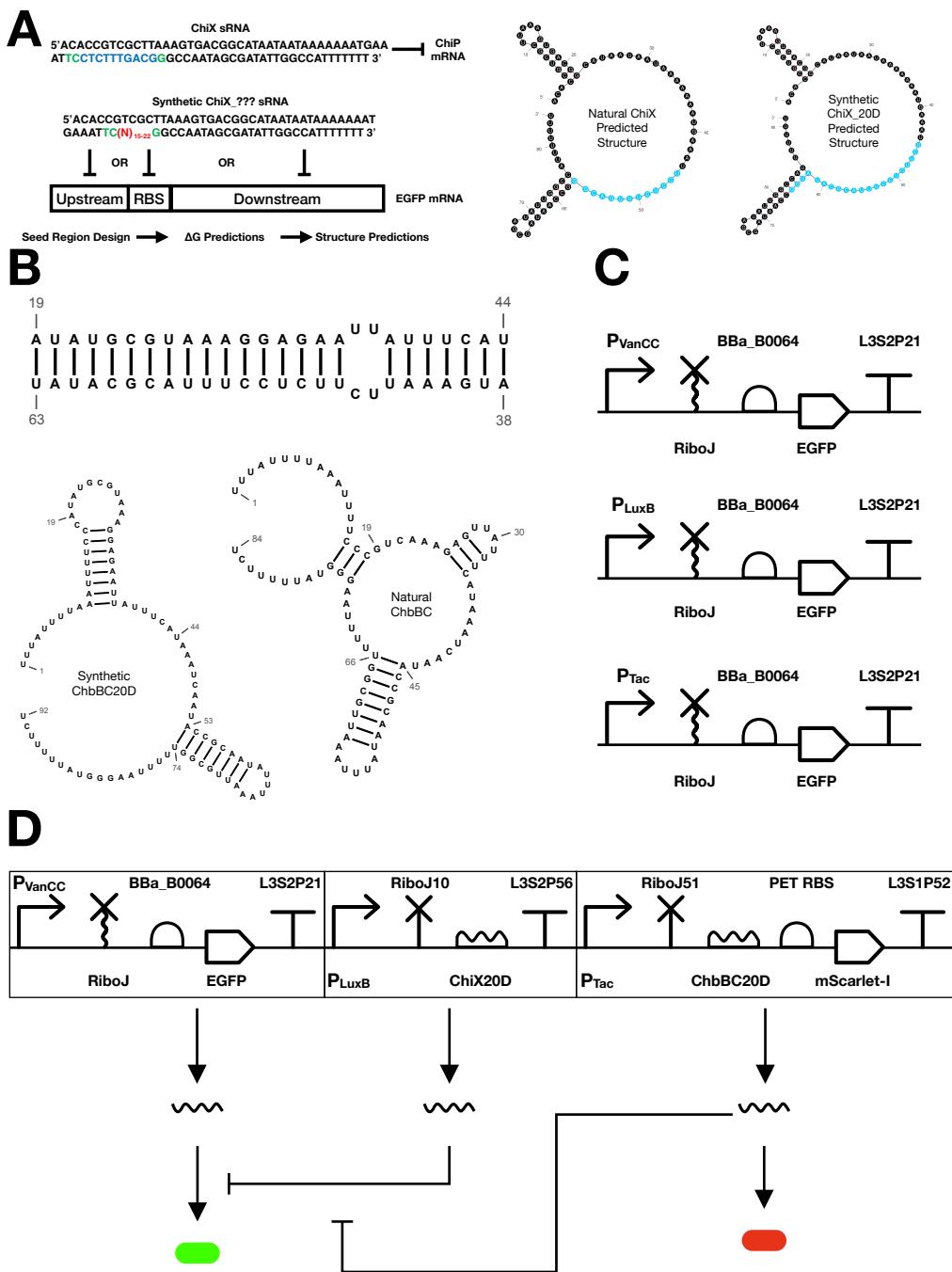


Figure 1: Design and Build: (A) Synthetic ChiX was designed by replacing the blue nucleotides in the seed region to target either upstream of, downstream of, or at the RBS. Binding thermodynamics with EGFP were predicted using intaRNA and structures were predicted using Vienna RNAFold. The natural structure is maintained in synthetic ChiX20D as shown on the right. (B) Synthetic ChbBC20D was designed by taking the reverse complement of the seed region in Synthetic ChiX20D and replacing the ChbBC seed region. The UU-CU mismatch was a maintained mismatch from the natural interaction. Below the predicted structures are displayed with equivalent bases numbered. The synthetic sequence has a different structure but the AU-rich stem loop between bases 45 and 66 is maintained, likely an Hfq binding site. (C) SBOL Visual representations of the three level 1 TUs assembled to allow standardised promoter characterisation. (D) SBOL Visual representation of the designed Synthetic spRNA circuit and below, a schematic of the regulatory interactions between mRNA and sRNA, and sRNA and spRNA.

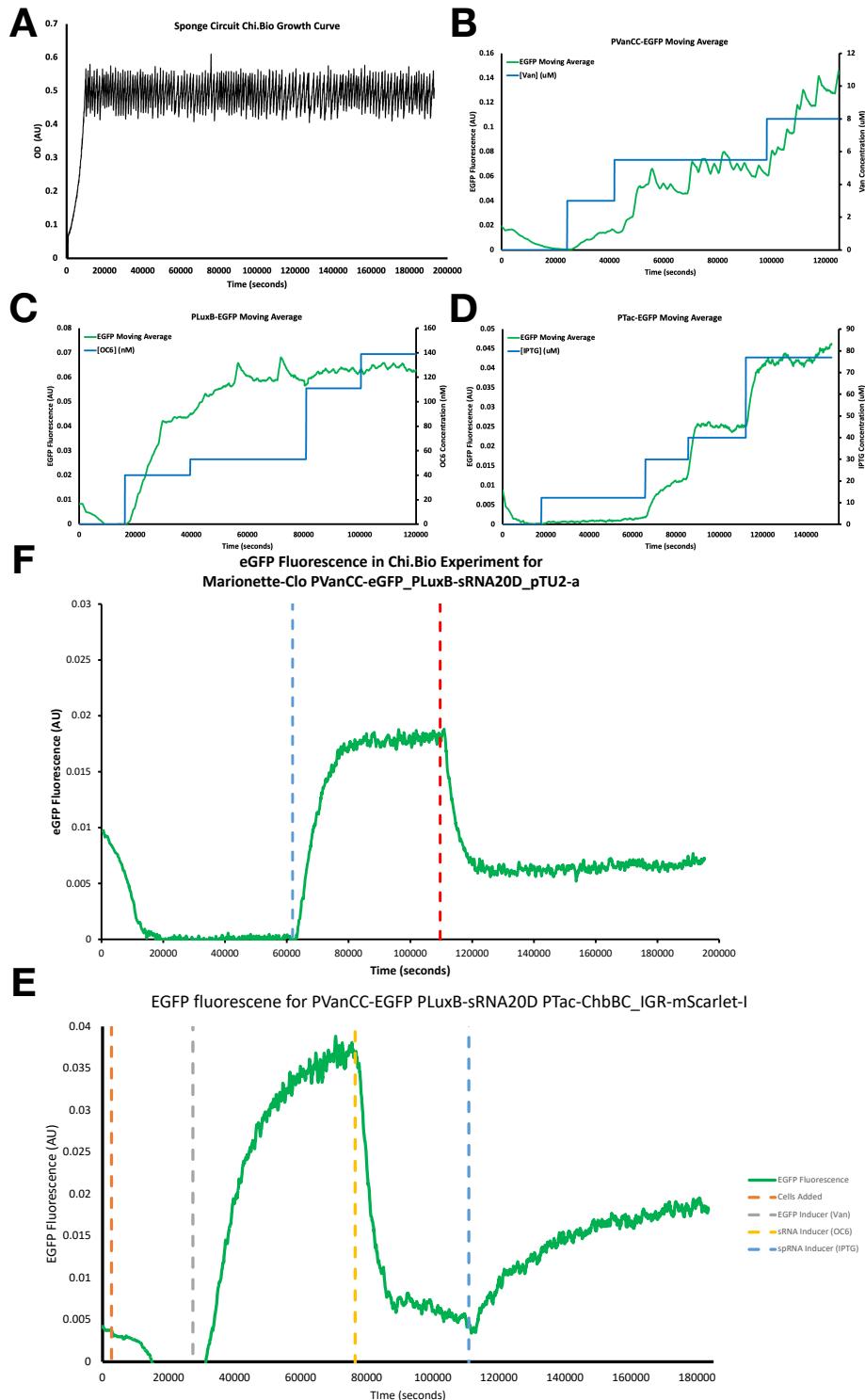


Figure 2: Test: (A) OD curve for an experiment in the Chi.Bio demonstrating turbidostat functionality. (B-D) EGFP fluorescence in level 1 TU in response to inducer step changes. (F) Validation of Synthetic-ChiX20D functionality by inducing EGFP, followed by inducing sRNA, demonstrating efficient knockdown. (E) Validation of Synthetic-ChbBC20D functionality by inducing EGFP, followed by sRNA, and then inducing spRNA demonstrating recovery of fluorescence.

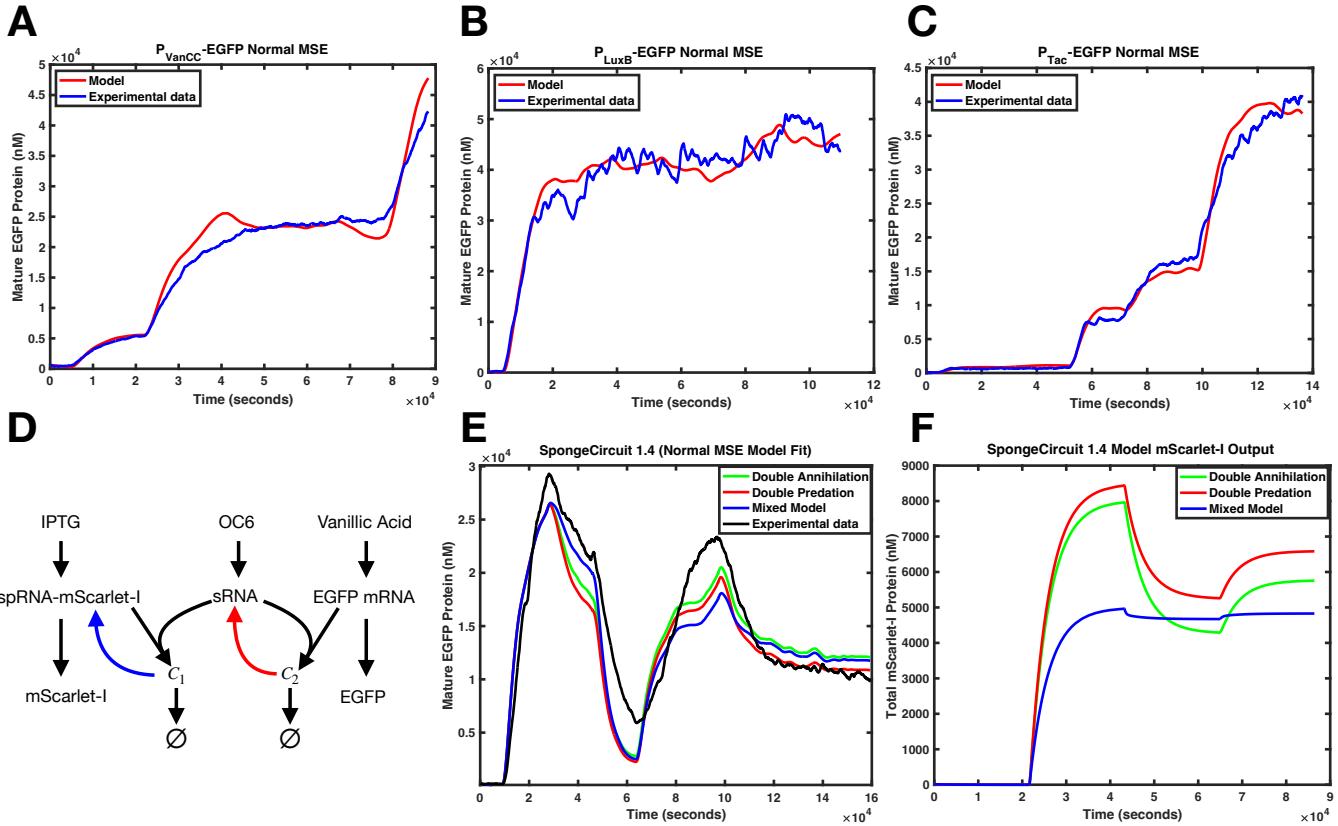


Figure 3: Learn: (A-C) Model fit Vs. experimental data for EGFP expressing level 1 TU. (D) Model architectures for sponge RNA circuit. In black are reactions present in all models. In blue a reaction present in both model 2 and 3. In red a reaction present in only model 2. (E) Model fit Vs. experimental data for the three model architectures describing the synthetic spRNA circuit. (F) Simulation of each model, where spRNA is induced, followed by sRNA induction, and finally followed by EGFP induction. Model 3 behaves qualitatively differently to the other two models.

Local RNA Feedback: More Logic, Less Leakage

Nicolai Engelmann, Maik Molderings, Heinz Koeppl

TU Darmstadt, Germany

{nicolai.engelmann,maik.molderings,heinz.koeppl}@tu-darmstadt.de

1 INTRODUCTION AND CONCEPT

The constituents of many cellular biochemical processes, such as the binding of regulator proteins to DNA binding-sites, appear to be related by dose-response curves that have sigmoid shape in a logarithmic argument [1]. This phenomenon has enabled synthetic biologists to engineer artificial genetic pathways that compute logic operations in cells [10]. Arithmetic operations on regulators such as subtractions, caused e.g. by titration effects [6], can increase or decrease the steepness of dose-response curves in the logarithmic domain. This has been observed in previous works [4, 19]. Qualitative effects like this have been exploited, e.g. in the engineering of genetic NOR gates, where the presence of two transcription factors (TFs) in high copy numbers that share the same target could effectively be understood as a logical OR operation [24]. Such a design has later been used as a building block for the genetic design software Cello [16].

Antisense RNA-Targeting and Gene Silencing. "Syphoning" regulating proteins can already happen at the level of RNA. Antisense RNA (asRNA) with high target specificity has wide usage in gene silencing in therapeutic applications [13, 15]. In these applications, non-coding RNA (ncRNA) that is complementary to regions of the target messenger RNA (mRNA) are transfected into cells. This ncRNA binds to the mRNA and forms double-strand RNA (dsRNA) that is rapidly degraded by the host as a security measure, e.g. against viral attacks [17]. Prokaryotic and eukaryotic cells alike maintain mechanisms that allow rapid degradation of dsRNA although they may work very differently.

Arithmetic operations using Antisense RNA-Feedback. Our goal is to exploit this syphoning architecture to mutually degrade input and output RNA of genetic logic gates. In this way, we obtain a subtraction on the biochemical species' that make up a gate's input and output. We use the dsRNA formation and degradation rates to increase the steepness of the dose-response curve and reduce output leakage. Our method is designed for *molecular NOR* gates, i.e. NOR gates that map chemical species to chemical species [9, 11, 24].

2 RNA FEEDBACK FOR NOT/NOR GATES

Usage of RNA mediated feedback within a synthetic circuit offers great advantages. RNA transduces signals directly, circumventing the translational process, which leads to a minimal additional metabolic burden to the circuit [12, 14, 18].

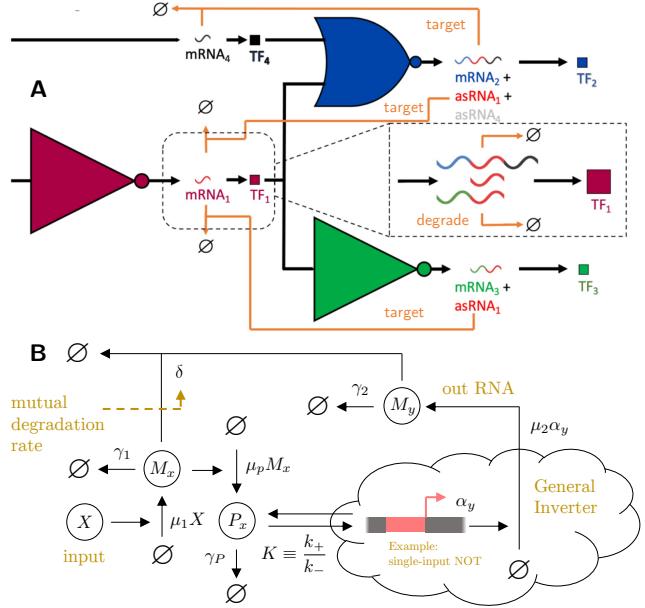


Figure 1: Illustration of the proposed feedback architecture. A) Feedback is implemented across gates in a molecular NOR-logic circuit by simultaneously transcribing antisense parts, that target mRNA of input TFs. B) Reaction network for a NOT gate with RNA feedback. Input RNA M_x is degraded together with output RNA M_y in a common nonlinear reaction with rate δ . Note, that the specific implementation of the inverter circuit is not important, because the feedback is positive in M_y .

We add the RNA feedback by using *trans* acting asRNAs that are adjacent of the coding mRNA of the output TF in a single transcript. Placement of start and stop codon surrounding the coding part of the mRNA region ensures translation of the TF. The antisense region is complementary to the coding region of the input TFs transcript as shown in Fig. 1A. Forming of dsRNA through Watson-Crick pairing results in repression of translation, due to degradation of the RNA-RNA duplex by diverse mechanisms present in different hosts [23, 26, 27] or blocking ribosome binding [2]. Inspired by regulation with antithetic integral feedback [5, 8], we propose, that mediating feedback by sequestering of sense and antisense RNA can lead to improvement in gate and therefore whole circuit function. While mRNA decay in eukaryotes is relatively slow, decay of mRNA in prokaryotes like *E. coli* takes place

within minutes [3, 22]. In order to improve gate function in *E. coli*, formation of dsRNA should happen within seconds. We hypothesize, that the feedback is fast enough, since dsRNA formation by sense and antisense RNA in *E. coli* is a naturally occurring event [25], therefore giving evidence, that dsRNA formation is faster than single stranded mRNA decay.

3 SIMPLE NOT-GATE QUANTITATIVE STUDY

Consider the reaction network of a simple NOT gate with feedback illustrated in Fig. 1B. It shows an arbitrary input species X that modifies the transcription rate of an RNA M_x , which is then translated to protein P_x . This protein acts as a repressing TF on the promoter that enables transcription of RNA M_y . The quantity α_y denotes the promoter activity. M_x and M_y contain the matching sense and antisense regions that lead to dsRNA formation and successive degradation by the host's responsible mechanism (e.g. RNase III in *E. coli*). Small letters are concentrations, e.g. $x \equiv [X]$. The steady-state equations obtained from the network's reaction rate equations are then

$$m_x = \frac{\mu_1 x}{\gamma_1 + \delta m_y} \quad \alpha_y = \frac{\kappa^n (1 - \beta)}{\kappa^n + (r_x m_x)^n} + \beta \quad m_y = \frac{\mu_2 \alpha_y}{\gamma_2 + \delta m_x}, \quad (1)$$

where $\beta \in [0, 1)$ determines the amount of leakage, $r_x \equiv \gamma_P^{-1} \mu_P$ is a coefficient that generally depends on the inverter circuit used, and n is a Hill coefficient. Finding an explicit closed-form expression for α_y or m_y in x is not possible in the general case because of the feedback connection. However, numerical evaluation is possible, e.g. using a fixed-point iteration on (1) or solving a polynomial root-finding problem. These methods also scale to circuits of many gates with potential feedback connections [7].

Zero-leakage in the Limit. Maximal gene expression is reached for $x = 0$. Then, (1) suggests $m_x = 0$, $p_x = 0$ and so $m_y = \frac{\mu_2}{\gamma_2}$. For the minimal expression, we let $x \rightarrow \infty$ and assume $m_y < \infty$ stays finite. Then, $m_x \rightarrow \infty$ and as a consequence $\alpha_y \rightarrow \beta$. Thus, we obtain $m_y = \lim_{m_x \rightarrow \infty} \frac{\mu_2 \beta}{\gamma_2 + \delta m_x} = 0$. In this idealistic setting, we thus approach very low steady-state leakage for an input x that is very large (Fig. 2A). Note, that a larger δ does not change the slope of the *asymptotic* decrease in m_y in a doubly logarithmic plot.

Logic Transition Region. The transition region in m_y becomes steeper the larger δ becomes, but also changes its location gradually (Fig. 2B). The x_{tr} that causes the largest change in m_y can be found numerically from (1) iterating over x . However, we can also give a less expensive asymptotic estimate for x_{tr} in the form of the root-finding problem

$$0 = [\mu_1 x_{tr} - \mu_2 \beta + \Delta \gamma] [\tau^n + m(x_{tr})^n] - \mu_2 \tau^n (1 - \beta) \quad (2)$$

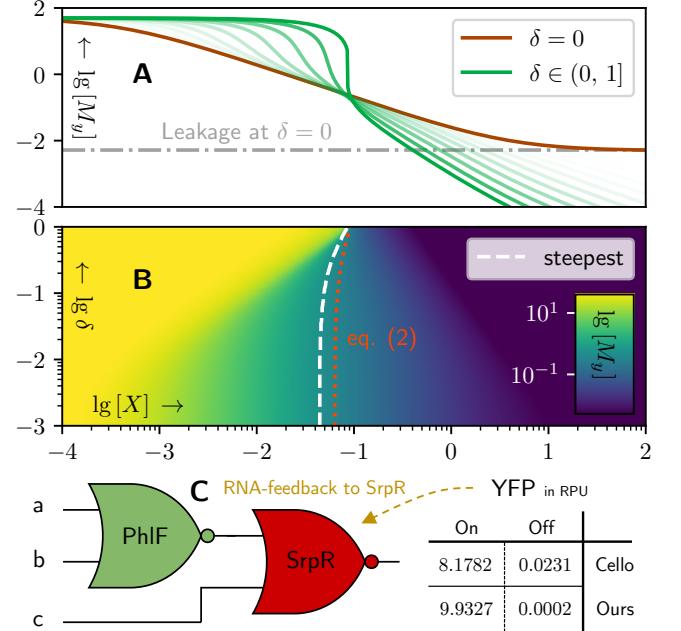


Figure 2: Change of transition function (dose-response) with change of δ . A) shows the logarithmic RNA concentration $\lg [M_y]$ against the one of the input species $\lg [X]$. Feedback ($\delta > 0$) increases steepness and reduces leakage in M_y . B) shows the narrowing transition region with increasing $\delta \in (0, 1]$. The location of the steepest slope shifts in the process and the shape is no longer logistic. An asymptotic approximation to locate the transition region is given in (2). C) Cello circuit 0x70 with optimal gate assignment and RNA-feedback from YFP to SrpR; lowest ON and highest OFF. We stress that this *in silico* experiment is under ideal conditions.

with $m(x_{tr}) \equiv -\frac{\gamma_1}{2\delta} + \sqrt{\left(\frac{\gamma_1}{2\delta}\right)^2 + \frac{\mu_1}{\delta} x_{tr}}$, $\Delta \gamma \equiv \gamma_2 - \gamma_1$ and $\tau \equiv r_x^{-1} \kappa$. Eq. (2) is based on the assumption that the larger δ becomes, the closer x_{tr} moves to the location where $m_x = m_y$. Eq. (2) also suggests that in the limit $\delta \rightarrow \infty$, we approach $x_{tr} = \frac{1}{\mu_1} (\gamma_1 - \gamma_2 + \mu_2)$. This corresponds to the case, where m_x matches the maximal m_y from a fully active promoter $\alpha_y = 1$.

4 CIRCUIT AND TECHNOLOGY MAPPING

Beyond Single-Input Gates. Mutual repression of translation between inputs and outputs of a logic gate can impact its logic function. Therefore, a gate with multiple inputs logically consistent with the single-transcript RNA feedback connection must implement a generalized inverter structure. Let $O \in \{0, 1\}$ be the Boolean output of a gate and $I_n \in \{0, 1\}$ for $n \in \mathcal{I} \subset \mathbb{N}$ its enumerated Boolean inputs. Let $\mathcal{T} \subset \mathcal{I}$ be the indices of the inputs targeted by RNA feedback. We must ensure $O = \overline{\sum_n I_n}$. Hence, $\sum_n I_n$ fully determines the O 's

213 value and any other inputs $n' \in \mathcal{I}$, $n' \notin \mathcal{T}$ become *don't care*.
 214 As a consequence, removing the inputs that are *don't care*,
 215 the resulting multi-input gate must implement NOR-logic.

216 *The RNA-Feedback NOR-logic Circuit.* Let the topology of a
 217 NOR-logic circuit be a directed graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$, where
 218 the vertices (i.e. gates) $\mathcal{V} \equiv \{v_1, \dots, v_K\}$ are enumerated by
 219 a totally ordered index set \mathcal{K} . The set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of directed
 220 edges (i.e. wires) in conjunction with the enumeration \mathcal{K}
 221 lets us define incoming $I_k \equiv \{m \mid (v_m \rightarrow v_k) \in \mathcal{E}, m \in \mathcal{I}\}$
 222 and outgoing $O_k \equiv \{m \mid (v_k \rightarrow v_m) \in \mathcal{E}, m \in \mathcal{I}\}$ index sets
 223 each for vertex v_k . We also define input $\mathcal{U} \subset \mathcal{I}$ and output
 224 $\mathcal{Y} \subset \mathcal{I}$ index sets for the whole circuit. While the topology
 225 \mathcal{G} is obtained in the logic synthesis step of the automated
 226 design pipeline, the technology mapping step introduces an
 227 injective labelling function $M : \mathcal{V} \rightarrow \mathcal{L}$ that maps each
 228 vertex to an element of a library \mathcal{L} , i.e. a set of gate param-
 229 eters from a gate library. Let the mapped gate parameters
 230 $M(v_k) = (\mu_k, \gamma_k, \beta_k, \delta_k, n_k, \kappa_k, r_k) \in \mathcal{L}$ be known and the
 231 variable quantities (α_k, m_k) be the gate's (output) promoter
 232 activity α_k and (output) RNA concentration m_k . Then, to
 233 obtain the circuit response α_y for $y \in \mathcal{Y}$ given fixed input
 234 $\alpha_u = \text{const.}$ for $u \in \mathcal{U}$, we need to solve the set of fixed-point
 235 equations

$$\begin{aligned} m_k &= \frac{\mu_k \alpha_k}{\gamma_k + \delta_k \sum_{l \in O_k} m_l + \sum_{l \in I_k} \delta_l m_l} & (3) \\ \alpha_k &= \frac{\kappa_k^n (1 - \beta_k)}{\kappa_k^n + (\sum_{l \in I_k} r_l m_l)^n} + \beta_k \end{aligned}$$

242 for each $k \in \mathcal{K}$. The calculated $m_y \propto [P_y]$ for each $y \in \mathcal{Y}$
 243 that are proportional to concentrations of reporter proteins
 244 $[P_y]$ can then be used to score the circuit [16, 21].

245 *Cello Circuit Case Study.* Translation to Cello's circuit frame-
 246 work faces several challenges. On the one hand do Cello's
 247 NOR gates functionally act as NAND gates on the level of
 248 chemical species. However, applying RNA feedback to NOT
 249 gates is still possible. On the other hand, Cello's gates fea-
 250 ture only parameters $\alpha_{\max,k}$, $\alpha_{\min,k}$, K_k and n_k . Matching the
 251 parameters from our NOT example in section 3 is underde-
 252 termined and thus leads to ambiguities. However, we see
 253 that $\alpha_{\min,k} = \beta \alpha_{\max,k}$ and $\alpha_{\max,k} \propto \frac{\mu_k}{\gamma_k} r_k$ (with any $\delta = 0$)
 254 we have $p_k = r_k m_k = \frac{\mu_{P,k} \mu_k}{\gamma_{P,k} \gamma_k} \alpha_k$). Although oversimplifying,
 255 we can fix the degradation rates $\gamma_{P,k}$ and γ_k to values repre-
 256 senting their "typical" order of magnitude for *E. coli* (in full
 257 knowledge that they may vary greatly in reality). We also set
 258 $\mu_{P,k} \equiv \omega_{P,k}$ and $\mu_{P,k} \equiv \alpha_{\max,k} \omega_k$ with the proportionality
 259 factors $\omega_{P,k}$ and ω_k representing "typical" orders of magni-
 260 tude for translation and transcription rates. This unifies r_k
 261 for all gates and we finally set $\kappa = K_k r_k$. With this setup,
 262 we took circuit 0x70 from [16] and equipped our simulation
 263 engine [20] with equation system (3) to check how different
 264

265 the circuit behaved with $\delta_y = 1$ for the YFP end-stage. The
 266 comparison of the circuit's closest ON-OFF output pair in
 267 the configurations with and without feedback ($\delta_y = 1$ and
 268 $\delta_y = 0$) is given in Fig. 2C.

REFERENCES

- [1] ALON, U. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2019.
- [2] BAYER, T. S., AND SMOLKE, C. D. Programmable ligand-controlled riboregulators of eukaryotic gene expression. *Nature biotechnology* 23, 3 (2005), 337–343.
- [3] BERNSTEIN, J. A., KHODURSKY, A. B., LIN, P.-H., LIN-CHAO, S., AND COHEN, S. N. Global analysis of mrna decay and abundance in escherichia coli at single-gene resolution using two-color fluorescent dna microarrays. *Proceedings of the National Academy of Sciences* 99, 15 (2002), 9697–9702.
- [4] BREWSTER, R. C., WEINERT, F. M., GARCIA, H. G., SONG, D., RYDENFELT, M., AND PHILLIPS, R. The transcription factor titration effect dictates level of gene expression. *Cell* 156 (2014), 1312–1323.
- [5] BRIAT, C., GUPTA, A., AND KHAMMASH, M. Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks. *Cell systems* 2, 1 (2016), 15–26.
- [6] 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.
- [7] ENGELMANN, N., SCHWARZ, T., KUBACZKA, E., HOCHBERGER, C., AND KOEPPLE, H. Context-aware technology mapping in genetic design automation. *ACS Synthetic Biology* 12, 2 (2023), 446–459. PMID: 36693176.
- [8] FREI, T., CHANG, C.-H., FILO, M., ARAMPATZIS, A., AND KHAMMASH, M. A genetic mammalian proportional–integral feedback control circuit for robust and precise gene regulation. *Proceedings of the National Academy of Sciences* 119, 00 (2022), e2122132119.
- [9] GANDER, M. W., VRANA, J. D., VOJE, W. E., CAROTHERS, J. M., AND KLAIVINS, E. Digital logic circuits in yeast with crispr-dcas9 nor gates. *Nature communications* 8, 1 (2017), 15459.
- [10] GARDNER, T. S., CANTOR, C. R., AND COLLINS, J. J. Construction of a genetic toggle switch in escherichia coli. *Nature* 403, 6767 (2000), 339–342.
- [11] GOÑI-MORENO, A., AND AMOS, M. A reconfigurable nand/nor genetic logic gate. *BMC systems biology* 6, 1 (2012), 1–11.
- [12] KAFRI, M., METZL-RAZ, E., JONA, G., AND BARKAI, N. The cost of protein production. *Cell reports* 14, 1 (2016), 22–31.
- [13] KAY, C., SKOTTE, N., SOUTHWELL, A., AND HAYDEN, M. Personalized gene silencing therapeutics for huntington disease. *Clinical genetics* 86, 1 (2014), 29–36.
- [14] LEHR, F.-X., HANST, M., VOGEL, M., KREMER, J., GÖRINGER, H. U., SUESS, B., AND KOEPPLE, H. Cell-free prototyping of and-logic gates based on heterogeneous rna activators. *ACS synthetic biology* 8, 9 (2019), 2163–2173.
- [15] MEISTER, G., AND TUSCHL, T. Mechanisms of gene silencing by double-stranded rna. *Nature* 431, 7006 (2004), 343–349.
- [16] NIELSEN, A. A., 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.
- [17] OBBARD, D. J., GORDON, K. H., BUCK, A. H., AND JIGGINS, F. M. The evolution of rnai as a defence against viruses and transposable elements. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1513 (2009), 99–115.
- [18] ROSENFIELD, N., AND ALON, U. Response delays and the structure of transcription networks. *Journal of molecular biology* 329, 4 (2003), 318

319	645–654.	372
320	[19] RYDENFELT, M., COX III, R. S., GARCIA, H., AND PHILLIPS, R. Statistical mechanical model of coupled transcription from multiple promoters due to transcription factor titration. <i>Physical Review E</i> 89, 1 (2014), 012702.	373
321		374
322	[20] SCHLADT, ENGELMANN, K. E. A. ARCTIC Design Automation Toolchain. https://gitlab.rs.e-technik.tu-darmstadt.de/arctic/arctic , 2021.	375
323		376
324	[21] SCHLADT, T., ENGELMANN, N., KUBACZKA, E., HOCHBERGER, C., AND KOEPLI, H. Automated design of robust genetic circuits: Structural variants and parameter uncertainty. <i>ACS synthetic biology</i> 10, 12 (2021), 3316–3329.	377
325		378
326	[22] SELINGER, D. W., SAXENA, R. M., CHEUNG, K. J., CHURCH, G. M., AND ROSENOW, C. Global rna half-life analysis in escherichia coli reveals positional patterns of transcript degradation. <i>Genome research</i> 13, 2 (2003), 216–223.	379
327		380
328	[23] SINGH, I. K., SINGH, S., MOGILICHERLA, K., SHUKLA, J. N., AND PALLI,	381
329		382
330		383
331		384
332		385
333		386
334		387
335		388
336		389
337		390
338		391
339		392
340		393
341		394
342		395
343		396
344		397
345		398
346		399
347		400
348		401
349		402
350		403
351		404
352		405
353		406
354		407
355		408
356		409
357		410
358		411
359		412
360		413
361		414
362		415
363		416
364		417
365		418
366		419
367		420
368		421
369		422
370		423
371		424

Rule-based generation of synthetic genetic circuits

Masayuki Yamamura¹, Kazuteru Miyazaki², Sota Okuda³,
Ryoji Sekine¹, Naoki Kodama⁴, Daisuke Kiga³

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

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

Introduction

Since the introduction of synthetic biology grounded in engineering principles, combining elements such as parts and devices has been perceived as critical. Synthetic genetic networks manually designed are an example of such a combination and were examined for their dynamical properties in both living cells and mathematical models.

In bio-design automation, model parameterization, as well as network motif selection, is also essential for a specification of cellular behavior. An automation tool designed for Boolean networks assigns biological components to nodes within these networks and determines those components' dynamic behavior to verify the designs' feasibility¹. Tools calculating other types of networks are also developed²⁻⁴, and databases for the models or simulation results are provided^{5,6}. Similar to those studies, our steps of bio-design automation for genetic circuit design are based on a logical inference engine that can call modules for numerical calculations of the dynamical properties of candidate circuits. In the calculation, our inference engine can also combine the listed parameters for each molecule and examine the property of the network. Furthermore, the inference engine can select, combine, and modify network motifs described in the rules in the engine (Fig 1). Because the engine can include alternative rules corresponding to each of the sub-goals to achieve the specification, the inference engine can generate multiple network structures, each of which has sets of appropriate parameters for the specification.

For an example of design using the engine, our IWBDA2022 abstract referred to our previous paper using a synthetic genetic circuit for reprogramming and diversification of gene-expression status of living cell⁷. In the manual design procedure in the paper, we have initially combined the toggle switch and gene

overexpression motifs in a cell because manual phase-space analysis of the toggle switch with and without the overexpression shows bifurcation between bistability and monostability both are required for the reprogramming and diversification process. For design automation of the circuit using an inference engine, we presented a pseudo code at IWBDA 2022 poster, and then developed a combination of Inside Prolog and C++ codes available in Zenodo⁸.

Using this flexibility and scalability of the Prolog inference engine, our modification of the IWBDA 2022 code here achieved an automated design of a genetic network that regulates intercellular-communication dependent diversification of cell type.

Results and Discussion

For a design of cell-population behavior regulated by cell-cell communication using the inference engine Prolog, specification is diversification of cell status on epigenetic landscape (Fig 2A). As well as our previous reprogramming and diversification by the gene overexpression, this diversification behavior dependent on the intercellular communication has been achieved in test tubes using our synthetic circuit in living cells⁹. In the circuit, the toggle switch motif was modified at two positions to be combined with the cell-cell communication motif (Fig 2B and 2C). By modifying the promoter sequence, the circuit bifurcates between monostability and bistability depending on the concentration of communication molecules, to satisfy the specification.

In design flows for the communication-dependent cell-type diversification in the inference engine, parameterized models initially designed for multistability were modified to incorporate cell-cell communication, and the modified model was examined for the movement specification on the epigenetic

landscape. A more detailed description of this flow consists of four steps. [Step 1] Parameterized models for bistability are designed using a part of our previous reprogramming code. For examinations of bistability, the Prolog code calls a numerical calculation module. [Step 2] To implement the bifurcation dependent on the concentration of the communication molecule, a production term in an equation in the model is modified (Fig 2B and 2C). In the original toggle switch, the production is negatively regulated by the concentration of a repressor expressed from the other side of the toggle switch. In addition to the negative regulation, the modified production term is also regulated by the concentration of the communication molecule simultaneously. Additionally, the communication-molecule-dependent bifurcation is examined by the numerical calculation module. [Step 3] A regulation mechanism for increasing the communication molecule is selected from the rules in the code. Although the increase in our original paper depends on an enzyme expression regulated by one of the repressors of the toggle switch, constant expression of the enzyme is an alternative method for the increase. Another simple method is constantly adding the molecule to the test tube. [Step 4] To examine the specification, the engine calculates the time developments in gene-expression states of cells and the developments in the communication-molecule concentration shared among the cells. Because the molecules easily permuted the cell membrane, we assumed the same concentration in the cells and their environment. On the contrary, gene-product concentrations differ among the cell population. Before the iterations in the calculation, the initial mono-modal state of the cells is defined. After the calculation, the number of clusters of cell states is counted to examine the specification: diversification on the epigenetic landscape.

To generate genetic networks that meet a given specification, our code, written in the Prolog inference system, can combine rules that describe network motifs, collections of assumed parameters for components of the motifs, and mathematical conditional statements derived from the specification. The logical structure of rules can help us uncover alternative methods of genetic manipulations. The Prolog's flexibility allows for the addition of such alternative methods even contributed by multiple programmers, thus expanding

the scale and complexity of generated networks. Its versatility also generates texts and files describing parameters for diverse tools or databases utilized by the Prolog code. As well as the numerical calculation program we have written in C, standardized synthetic biology tools and databases will be connected in our future studies.

REFERENCES

1. Jones, T. S.; Oliveira, S. M. D.; Myers, C. J.; Voigt, C. A.; Densmore, D. Genetic circuit design automation with Cello 2.0. *Nat. Protoc.* **2022**, *17*, 1097-1113.
2. Tas, H.; Grozinger, L.; Goni-Moreno, A.; de Lorenzo, V. Automated design and implementation of a NOR gate in *Pseudomonas putida*. *Synth. Biol. (Oxf)* **2021**, *6*, ysab024.
3. Boada, Y.; Reynoso-Meza, G.; Pico, J.; Vignoni, A. Multi-objective optimization framework to obtain model-based guidelines for tuning biological synthetic devices: an adaptive network case. *BMC Syst. Biol.* **2016**, *10*, 27-0.
4. Dalchau, N.; Smith, M. J.; Martin, S.; Brown, J. R.; Emmott, S.; Phillips, A. Towards the rational design of synthetic cells with prescribed population dynamics. *J. R. Soc. Interface* **2012**, *9*, 2883-2898.
5. McLaughlin, J. A.; Myers, C. J.; Zundel, Z.; Misirli, G.; Zhang, M.; Ofiteru, I. D.; Goni-Moreno, A.; Wipat, A. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synth. Biol.* **2018**, *7*, 682-688.
6. Yanez Feliu, G.; Earle Gomez, B.; Codoceo Berrocal, V.; Munoz Silva, M.; Nunez, I. N.; Matute, T. F.; Arce Medina, A.; Vidal, G.; Vitalis, C.; Dahlin, J.; Federici, F.; Rudge, T. J. Flapjack: Data Management and Analysis for Genetic Circuit Characterization. *ACS Synth. Biol.* **2021**, *10*, 183-191.
7. Ishimatsu, K.; Hata, T.; Mochizuki, A.; Sekine, R.; Yamamura, M.; Kiga, D. General applicability of synthetic gene-overexpression for cell-type ratio control via reprogramming. *ACS Synth. Biol.* **2014**, *3*, 638-644.
8. <https://doi.org/10.5281/zenodo.8148662>.
9. Sekine, R.; Yamamura, M.; Ayukawa, S.; Ishimatsu, K.; Akama, S.; Takinoue, M.; Hagiya, M.; Kiga, D. Tunable synthetic phenotypic diversification on Waddington's landscape through autonomous signaling. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 17969-17973.

Goal: Specification

- sub-goal 1: network motif selection
- sub-goal 2: motif combination/ modification
- sub-goal 3: numerical calculation

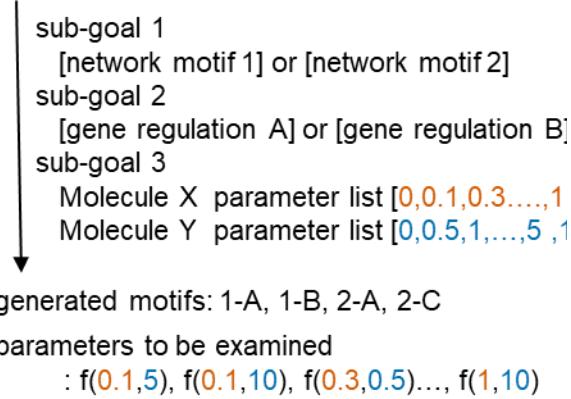


Fig 1

Diagram of design flow of Prolog inference engine. A specification of cellular behavior regulated by a genetic circuit is manually broken down into sub-goals, each of which can be a rule of the inference engine. Each sub-goal can have multiple candidates. Prolog generates parameterized networks by combination of rules and examines dynamical properties which are evaluated C++-based numerical calculation modules call by Prolog. Differences in combinations can be a source of alternative methods in genetic manipulation.

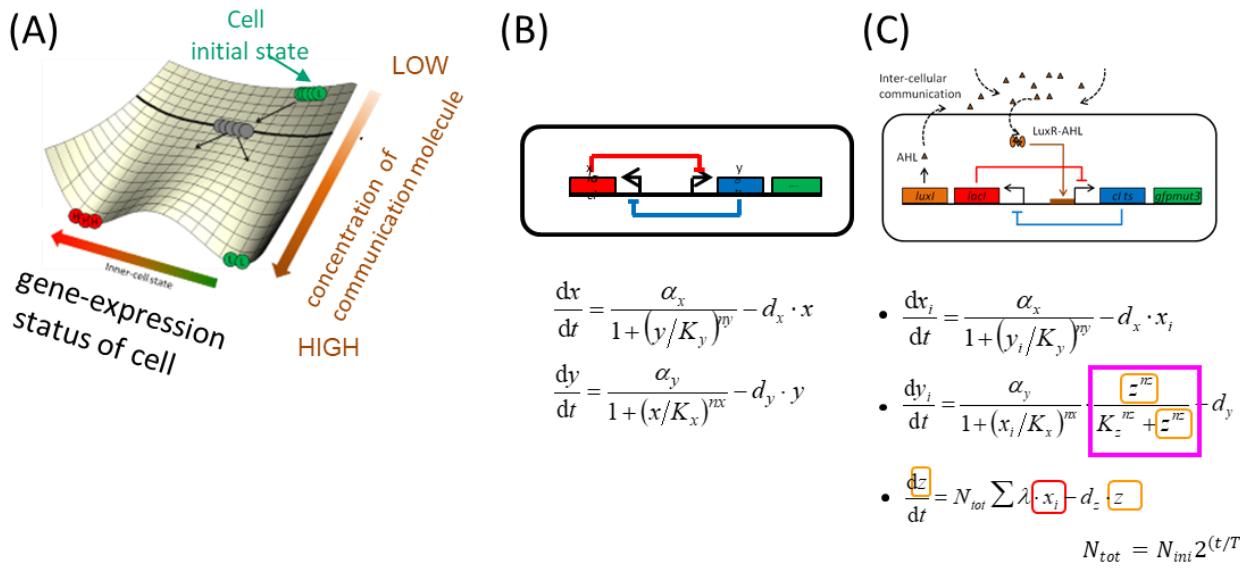


Fig 2

To explain the intercellular communication-dependent cellular behavior of the synthetic diversification system regulated by a modified toggle switch circuit, we use a landscape representation. (A) Landscape. The horizontal axis of the landscape represents the cell state, indicating the cellular concentration of LacI. The vertical axis of the landscape represents the AHL concentration. (B) Network diagram of the original toggle switch and the model for the circuit. CI_x and LacI_y mutually inhibit each other. x and y denote repressor concentrations. (C) Network diagram of the synthetic diversification circuit and the model for the circuit. Under the condition of a substantial amount of AHL, the Plux/lac promoter behaves as a Plac promoter in the original toggle switch. Note that concentrations of the repressors can differ among cells. z denotes the communication-molecule concentration shared by the cells. Due to the binding-dissociation equilibrium and high membrane permeability of AHL, we assumed equal AHL concentration among cells.

A Collection of Biological Models for the Development of Infinite-State Stochastic Model Checking Tools

Lukas Buecher¹, Payton J. Thomas⁴, Mohammad Ahmadi³, Josh Jeppson², Andrew Gerber², Eric Reiss², Chris Winstead², Hao Zheng³, Zhen Zhang², Chris J. Myers¹

¹University of Colorado Boulder, Boulder, USA, ² Utah State University, Logan, USA

³ University of South Florida, Tampa, USA, ⁴ University of Utah, Salt Lake City, USA
chris.myers@colorado.edu

1 INTRODUCTION

As synthetic biology transitions from proposed lab experiments to real-life applications, it becomes crucial for circuits to exhibit robustness in order to function effectively in changing environments outside the lab. Advancements in computational analysis offer designers the opportunity to enhance model predictability and genetic design robustness. Specifically, stochastic models prove highly effective in studying biochemical systems, encompassing chemical reaction networks and genetic circuits [9].

Stochastic models encode a system's probabilistic transitions and interactions in precise mathematical formalisms. Stochastic model checking then uses defined properties to calculate the exact probability of specific events occurring or the expected time to reach certain states. Compared to simulations that estimate the probabilities of failure, stochastic model checking does not just calculate the probability of failure, but also determines the failure's cause [2]. Nonetheless, models of biological systems often feature an infinite state space, since proteins and other chemical species can exist in any amount, rendering them intractable for analysis using most existing tools. Consequently, it becomes imperative to develop tools capable of verifying stochastic models with infinite state spaces.

To facilitate the development of improved software tools for computational analysis, this paper presents a novel collection of case studies that focuses specifically on biochemical systems with infinite state spaces. These case studies will allow for the evaluation and assessment of the performance of stochastic model-checking tools as they are being developed.

2 RESULTS

The provided case studies cover a range of models including chemical reaction systems, complex genetic circuits, and simple benchmarks. The models are accessible in the PRISM language [8] with each encompassing at least one intentionally designed artificial property specified in *Continuous Stochastic Logic* (CSL) for testing purposes. The properties for the simple models check if a given chemical species reaches its expected steady state value within 50 time units. As an example, Property 1 is used to calculate the probability of

species X staying below its steady state concentration of 6 within the first 50 time units.

$$P = ? [\text{true } U[0, 50] (X < 6)] \quad (1)$$

The properties for the more complex genetic circuits check the proper functioning of the circuit and are of real world interest. The example given in Property 2 specifies if the output of a genetic digital logic circuit stays low during an input transition from one state to another, both with low output. If the circuit turns on during the transition, it is not functioning as intended [11]. More details can be found in [3, 5].

$$P = ? [\text{true } U[0, 1000] (YFP \leq 30)] \quad (2)$$

The collection of models is briefly described below. Table 1 provides an overview of the models that are currently available. The case studies and the results of their analysis for comparison can be found on GitHub (https://github.com/fluentverification/CaseStudies_StochasticModelChecking).

Chemical reaction systems include chemical species reacting through defined reaction channels. These systems are essential for biomolecular processes and have applications in medicine, biomanufacturing, and biofuels [1]. Stochastic modeling is crucial in these systems due to their low molecule counts, which traditional chemical kinetics cannot adequately capture [13]. The collection presents five stochastic models of chemical reaction systems that focus on calculating the low probability of rare events.

Synthetic biologists design genetic circuits by assembling defined biological parts to add desired functionalities to biological systems. Automation in the design process allows scientists to model and analyze their genetic circuits *in silico* to test the system before implementation. Genetic circuit CTMC models have an infinite state space and improved stochastic model-checking tools would not only calculate the probability of failure but also its cause [2]. The collection includes four genetic circuit models shown in Table 1.

The *Systems Biology Markup Lanugage* (SBML) is an XML-based data format encoding models of biological systems. Our collection includes models from the SBML stochastic test suite [4], developed for testing the accuracy of stochastic simulators. The SBML stochastic test suite and all models in the

SBML language can be found at <https://github.com/sbmlteam/sbml-test-suite/tree/release/cases/stochastic>. Encouragingly, the SBML stochastic test suite has been used by various developers of several stochastic simulators as shown in [7]. The 30 available models in the benchmark suite include models describing birth-death processes, dimerization [15], and batch immigration-death processes [6]. The SBML stochastic test suite has been converted into the PRISM language to also allow the testing of software for stochastic model checking.

3 METHODS

An SBML-to-PRISM converter implemented in iBioSim [14], a genetic design automation tool, was used to convert the models to the PRISM language. This converter, written in Java, reads the SBML file, converts its contents into PRISM syntax, and translates any analysis constraints into mathematical formulas in PRISM format. The converter handles elements such as parameters, species, reactions, reaction rates, and constraints. Future iterations aim to include events and rules. The iBioSim software, including the SBML-to-PRISM converter, is available as open-source on GitHub at <https://github.com/MyersResearchGroup/iBioSim>.

4 DISCUSSION

Stochastic models have been crucial in the analysis of systems in diverse disciplines. With the increasing need of computational testing and verifying of designs in synthetic biology, the scope and complexity of the models increase together with the model's state space. Currently, there are still only limited support for infinite state space models in stochastic model checking tools. This work presents a novel collection of case studies of infinite state space stochastic models and properties to aid the development of such tools. Work with some of the models shown here aided in the development of the software tool STAMINA [10, 12]. By collecting and releasing these models, we hope to aid the community in developing and improving similar model checking tools. Finally, we want to invite the community to contribute to the collection of available models to help the growth of infinite state stochastic model-checking.

Acknowledgements

We thank Pedro Fontanarrosa and Curtis Madsen for providing layouts for Circuit0x8E and the Muller C-element, as well as all members of the FLUENT Verification Project (<https://fluentverification.github.io>) for their feedback. This work was supported by the National Science Foundation

under Grant No. 1856733, 1856740, and 1900542. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

- [1] AHMADI, M., THOMAS, P. J., BUECHERL, L., WINSTEAD, C., MYERS, C. J., AND ZHENG, H. A Comparison of Weighted Stochastic Simulation Methods for the Analysis of Genetic Circuits. *ACS Synthetic Biology* (Dec. 2022), acsynbio.2c00553.
- [2] AHMADI, M., ZHANG, Z., MYERS, C., WINSTEAD, C., AND ZHENG, H. Counterexample generation for infinite-state chemical reaction networks, 2022.
- [3] BUECHERL, L., ROBERTS, R., FONTANARROSA, P., THOMAS, P. J., MANTE, J., ZHANG, Z., AND MYERS, C. J. Stochastic Hazard Analysis of Genetic Circuits in iBioSim and STAMINA. *ACS Synthetic Biology* (Oct. 2021), acsynbio.1c00159.
- [4] EVANS, T. W., GILLESPIE, C. S., AND WILKINSON, D. J. The SBML discrete stochastic models test suite. *Bioinformatics* 24, 2 (Jan. 2008), 285–286.
- [5] FONTANARROSA, P., DOOSTHOSEINI, H., BORUJENI, A. E., DORFAN, Y., VOIGT, C. A., AND MYERS, C. Genetic Circuit Dynamics: Hazard and Glitch Analysis. *ACS Synthetic Biology* (2020), 15.
- [6] GILLESPIE, C. S., AND RENSHAW, E. The evolution of a batch-immigration death process subject to counts. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 461, 2057 (May 2005), 1563–1581.
- [7] GILLESPIE, C. S., WILKINSON, D. J., PROCTOR, C. J., SHANLEY, D. P., BOYS, R. J., AND KIRKWOOD, T. B. L. Tools for the SBML Community. *Bioinformatics* 22, 5 (Mar. 2006), 628–629.
- [8] KWIATKOWSKA, M., NORMAN, G., AND PARKER, D. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In *Computer Aided Verification* (Berlin, Heidelberg, 2011), G. Gopalakrishnan and S. Qadeer, Eds., Springer Berlin Heidelberg, pp. 585–591.
- [9] MADSEN, C., ZHANG, Z., ROEHNER, N., WINSTEAD, C., AND MYERS, C. Stochastic Model Checking of Genetic Circuits. *ACM Journal on Emerging Technologies in Computing Systems* 11, 3 (Dec. 2014), 1–21.
- [10] NEUPANE, T., MYERS, C. J., MADSEN, C., ZHENG, H., AND ZHANG, Z. STAMINA: Stochastic Approximate Model-Checker for INfinite-State Analysis. In *Computer Aided Verification* (Cham, 2019), Computer Aided Verification, Springer International Publishing, pp. 540–549.
- [11] 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 (Apr. 2016), aac7341–aac7341.
- [12] ROBERTS, R., NEUPANE, T., BUECHERL, L., MYERS, C. J., AND ZHANG, Z. STAMINA 2.0: Improving Scalability of Infinite-State Stochastic Model Checking. In *Verification, Model Checking, and Abstract Interpretation*, B. Finkbeiner and T. Wies, Eds., vol. 13182. Springer International Publishing, Cham, 2022, pp. 319–331.
- [13] SAMOILOV, M. S., AND ARKIN, A. P. Deviant effects in molecular reaction pathways. *Nature Biotechnology* 24, 10 (Oct. 2006), 1235–1240.
- [14] 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* 8, 7 (July 2019), 1560–1563.
- [15] WILKINSON, D. J. *Stochastic Modelling for Systems Biology, Third Edition*, zeroth ed. Chapman and Hall/CRC, Dec. 2018.

Table 1: Overview of the synthetic biology-inspired models with infinite state space. The model type characterizes the category and name of the model. The available number of differently abstracted variations of the given model is indicated in the second column. The last two columns indicate the mean number of species and reactions (rounded to the nearest integer) across the specific model. For example, Circuit0x8E is part of the genetic circuits category. Due to different abstraction and implementation methods, there are 48 models of the circuit available with a mean of 18 species and 16 reactions.

Model type	# of Models	Mean # of Species	Mean # of Reactions
<i>Chemical Reaction Systems</i>			
Reversible Isomerization	1	2	2
Single Species Production-Degradation	1	2	2
Enzymatic Futile Cycle	1	6	6
Modified Yeast Polarization	1	7	8
Simplified Motility Regulation	1	9	12
<i>Genetic Circuits</i>			
Circuit0x8E	48	18	15
Muller C-element	3	18	16
Repressilator	1	6	6
Dual Feedback Oscillator	1	4	4
<i>Benchmarks</i>			
SBML Test Suite	30	1	2
<i>Total</i>	88		

APPENDIX

This section presents the results of the analysis of the models and their properties for comparison for developers of novel software shown in Tables 2, 3, 4, 5, and 6. The models, along with their properties, were analyzed using iBioSim, PRISM, and STAMINA for result validation. iBioSim and PRISM employed Gillespie simulation and statistical model-checking,

respectively, with a using 1000 runs. STAMINA was used for stochastic model-checking, whenever possible. However, due to limitations such as excessive memory usage or time constraints, STAMINA was not always able to provide results within the given parameters. The following tables show the expected probabilities of all included models.

Table 2: The chemical reaction models with their given properties were simulated using software tools iBioSim [14], PRISM, and STAMINA. In iBioSim, the models were simulated using Gillespie with 1000 runs and a time limit of 50. PRISM was used to run statistical model-checking, again using a 1000 runs and a time limit of 50. STAMINA was used to run stochastic model-checking. Timeout indicates that STAMINA was not able to produce results within 5 minutes.

Chemical Reaction Networks	iBioSim	PRISM	STAMINA
Reversible Isomerization	0.0	0.0	0.0
Single Species Production-Degradation	0.0	0.0	0 - 2.05E-4
Enzymatic Futile Cycle	0.0	0.0	0 - 7.42E-4
Modified Yeast Polarization	0.0	0.0	Timeout
Simplified Motility Regulation	0.0	0.0	Timeout

Table 3: The results of the circuit0x8E simulation. The models with their given properties were analyzed using the software tools iBioSim and PRISM. In iBioSim, the models were simulated using Gillespie with 1000 runs and a time limit of 50. PRISM was used to run statistical model-checking, again using a 1000 runs and a time limit of 50. No results were obtained with STAMINA since the abstracted, finite representation of the infinite state space of each model was too large for analysis.

Circuit0x8E	iBioSim	PRISM
Circuit0x8E-000to011-G1	0.81	0.83
Circuit0x8E-000to011-G1-10-10	0.81	0.84
Circuit0x8E-000to101-G1	0.99	0.95
Circuit0x8E-000to101-G1-10-10E	0.99	0.96
Circuit0x8E-010to100-G0	0.70	0.73
Circuit0x8E-010to100-G0-10-10	0.70	0.70
Circuit0x8E-010to111-G0	0.32	0.30
Circuit0x8E-010to111-G0-10-10	0.32	0.30
Circuit0x8E-011to000-G1	0.83	0.86
Circuit0x8E-011to000-G1-10-10	0.83	0.86
Circuit0x8E-011to101-G1	0.98	0.99
Circuit0x8E-011to101-G1-10-10	0.98	0.99
Circuit0x8E-100to010-G0	0.79	0.81
Circuit0x8E-100to010-G0-10-10	0.79	0.76
Circuit0x8E-100to111-G0	0.33	0.29
Circuit0x8E-100to111-G0-10-10	0.33	0.32
Circuit0x8E-101to000-G1	0.87	0.85
Circuit0x8E-101to000-G1-10-10	0.87	0.85
Circuit0x8E-101to011-G1	0.97	0.98
Circuit0x8E-101to011-G1-10-10	0.97	0.98
Circuit0x8E-111to010-G0	0.90	0.90
Circuit0x8E-111to010-G0-10-10	0.90	0.91
Circuit0x8E-111to100-G0	0.93	0.93
Circuit0x8E-111to100-G0-10-10	0.93	0.93

Table 4: The results of the circuit0x8E LHF simulation. The models with their given properties were analyzed using the software tools iBioSim and PRISM. In iBioSim, the models were simulated using Gillespie with 1000 runs and a time limit of 50. PRISM was used to run statistical model-checking, again using a 1000 runs and a time limit of 50. No results were obtained with STAMINA since the abstracted, finite representation of the infinite state space of each model was too large for analysis.

Circuit0x8E	iBioSim	PRISM
Circuit0x8E-LHF-000to011-G1	0.82	0.82
Circuit0x8E-LHF-000to011-G1-10-10	0.82	0.83
Circuit0x8E-LHF-000to101-G1	0.99	0.99
Circuit0x8E-LHF-000to101-G1-10-10	0.99	0.99
Circuit0x8E-LHF-010to100-G0	0.24	0.22
Circuit0x8E-LHF-010to100-G0-10-10	0.24	0.27
Circuit0x8E-LHF-010to111-G0	0.29	0.28
Circuit0x8E-LHF-010to111-G0-10-10	0.29	0.30
Circuit0x8E-LHF-011to000-G1	0.80	0.76
Circuit0x8E-LHF-011to000-G1-10-10	0.80	0.76
Circuit0x8E-LHF-011to101-G1	0.99	0.99
Circuit0x8E-LHF-011to101-G1-10-10	0.99	0.99
Circuit0x8E-LHF-100to010-G0	0.72	0.75
Circuit0x8E-LHF-100to010-G0-10-10	0.72	0.74
Circuit0x8E-LHF-100to111-G0	0.33	0.32
Circuit0x8E-LHF-100to111-G0-10-10	0.31	0.33
Circuit0x8E-LHF-101to000-G1	0.73	0.72
Circuit0x8E-LHF-101to000-G1-10-10	0.73	0.74
Circuit0x8E-LHF-101to011-G1	0.98	0.99
Circuit0x8E-LHF-101to011-G1-10-10	0.98	0.99
Circuit0x8E-LHF-111to010-G0	0.93	0.89
Circuit0x8E-LHF-111to010-G0-10-10	0.92	0.9
Circuit0x8E-LHF-111to100-G0	0.92	0.90
Circuit0x8E-LHF-111to100-G0-10-10	0.92	0.92

Table 5: The results of the analysis of the genetic circuits. The models were analyzed with their given properties using the software tools iBioSim, PRISM, and STAMINA. In iBioSim, the models were simulated using Gillespie with 1000 runs and a time limit of 50. PRISM was used to run statistical model-checking, again using a 1000 runs and a time limit of 50. STAMINA was used to run stochastic model-checking. Timeout indicates that STAMINA was not able to produce results within 5 minutes.

Model	iBioSim	PRISM	STAMINA
Repressilator	0.38	0.41	0.38 - 0.38
Dual Feedback Oscillator	0.07	0.06	0.07 - 0.07
Muller C-element			
Majority-10-10	0.70	0.68	0.70 - 0.70
Speed-Independent-10-10	0.71	0.71	Timeout
Toggle-10-10	0.38	0.38	Timeout

Table 6: The results of the analysis of the benchmarks. The models were analyzed with their given properties using the software tools iBioSim, PRISM, and STAMINA. In iBioSim, the models were simulated using Gillespie with 1000 runs and a time limit of 50. PRISM was used to run statistical model-checking, again using a 1000 runs and a time limit of 50. STAMINA was used to run stochastic model-checking.

Benchmarks	iBioSim	PRISM	STAMINA
00001	0.66	0.68	0.67 - 0.67
00002	0.66	0.68	0.67 - 0.67
00003	0.94	0.94	0.94 - 0.94
00004	0.86	0.85	0.85 - 0.85
00006	0.66	0.66	0.67 - 0.67
00007	0.66	0.67	0.67 - 0.67
00008	0.66	0.68	0.67 - 0.67
00009	0.66	0.67	0.67 - 0.67
00010	0.66	0.72	0.67 - 0.67
00011	0.02	0.02	0.02 - 0.02
00012	0.66	0.68	0.67 - 0.67
00013	0.66	0.67	0.67 - 0.67
00014	0.66	0.68	0.67 - 0.67
00015	0.66	0.65	0.67 - 0.67
00016	0.66	0.66	0.67 - 0.67
00017	0.66	0.65	0.67 - 0.67
00018	0.66	0.66	0.68 - 0.68
00019	0.66	0.66	0.67 - 0.67
00020	0.98	0.98	0.98 - 0.98
00021	0.96	0.95	0.94 - 0.94
00022	0.96	0.96	0.96 - 0.96
00027	0.98	0.98	0.98 - 0.98
00028	1.00	1.00	1.00 - 1.00
00029	1.00	1.00	1.00 - 1.00
00034	0.14	0.16	0.14 - 0.14
00035	0.83	0.85	0.84 - 0.84
00036	0.14	0.14	0.14 - 0.14
00037	0.75	0.76	0.74 - 0.74
00038	0.65	0.69	0.66 - 0.66
00039	0.06	0.06	0.07 - 0.07

Guided Design of Genetic Circuits Exploiting Stochastic Model Verification

Lukas Buecher^{1,*}, Mohammad Ahmadi^{2,*}, Hao Zheng², Chris J. Myers¹

¹University of Colorado Boulder, ²University of South Florida,

* Both authors contributed equally
chris.myers@colorado.edu

1 INTRODUCTION

Synthetic Biology has established itself as a scientific discipline and is now making strides toward real-world applications [1]. However, as we venture beyond laboratory conditions and into systems operating outside controlled environments, ensuring the reliable functionality of genetic circuits within cells becomes paramount. As systems grow larger and more complex, the utilization of *genetic design automation* (GDA) tools becomes indispensable [2]. Regrettably, to the best of our knowledge, most currently available GDA tools lack support of circuit redesign as part of their generation capabilities.

In contrast to electronic circuits, genetic circuits exhibit a significantly higher level of unpredictability due to their inherent noisy behavior arising from low molecular counts [4, 10]. This noise introduces robustness challenges, compounded by the fact that existing tools like Cello [6, 9] do not adequately account for temporal dynamics. Genetic circuits respond to input molecules by orchestrating repression and activation events across different regions of the DNA. However, each transcriptional unit, which comprises the promoter, ribosome binding site, coding sequence, and terminator, exhibits distinct response times. Additionally, when input signals affect changes across multiple paths in the genetic circuit, variations in delay can arise, leading to undesirable switching behaviors commonly known as glitches [3, 5]. These issues pose significant challenges in real-world applications, as genetic circuits must function reliably and consistently.

In [5], the authors demonstrated the computational reproducibility of the glitching behavior observed in circuit 0x8E originally published by Nielsen et al [9]. This paper also proposed alternative designs of the same logic function with the aim of mitigating the glitching phenomenon during hazardous transitions [5]. Buecherl et al. [3] utilized the STAMINA model checking tool [8] to perform stochastic model verification on *continuous-time Markov chain* (CTMC) models to predict the likelihood of glitches to compare these various design choices. This approach enabled the evaluation of different design alternatives based on their predicted glitch occurrence probabilities.

This abstract explores the possibility of devising an automated analysis method on the CTMC models of genetic circuits to identify the specific pathways in the circuit that contribute to the high probability of a glitch. This is achieved by searching for high-probability *traces* in the model's state-space. A trace is a time-abstracted series of transitions showing a possible execution scenario of the model. comparing traces in which a glitch happens in the output to traces in which the output remains stable can highlight the execution scenarios and pathways that lead to the glitching behavior of the circuit.

Identifying the paths responsible for glitching behavior enables users to fine-tune circuit designs and address faulty transitions that the user deems critical for the system. If the analysis results demonstrate the need to either slow down or speed up specific paths through the logic, the user can select different genetic parts or modify the logic itself (e.g., by adding delay elements). *In silico* identification of the failure cause conserves valuable lab resources and leads to more robust designs, facilitating the transition to real-world applications.

2 RESULTS

This work analyzes one of the Cello circuits, which was originally designed by Nielsen et al. [9]. The selection of this circuit is motivated by the well-characterized nature of the Cello parts used. The logic of the chosen circuit, *Circuit 0xF6*, is illustrated in Figure 1. The circuit has three inputs, *Arabinose* (Ara, ChEBI = 17535), *Isopropyl-beta-D-thiogalactopyranoside* (IPTG, ChEBI=61448), *Acetylcholine* (aTc, ChEBI=15355), and *yellow fluorescent protein* (YFP) as an output reporter.

During an experiment, this circuit was transitioned from $(Ara, aTc, IPTG) = (0, 0, 0)$ to $(1, 1, 0)$, both resulting in an expected high YFP output. However, the final state $(1, 1, 0)$ exhibited lower fluorescence compared to the other output states, leading to the selection of this circuit for further examination. This specific transition is illustrated in the Karnaugh map shown in Figure 2. A *low* input, indicating the absence of the inducer in the cell, is denoted by a 0, and vice versa. When the inputs of the circuit transition from $(0, 0, 0)$ to $(1, 1, 0)$, two possible transition paths can occur: $(1, 0, 0)$ or $(0, 1, 0)$. If the transition proceeds via $(0, 1, 0)$ following the

blue arrow, no glitching behavior is observed. However, if the transition follows the red arrow via the state $(1, 0, 0)$, the circuit briefly decreases *YFP* production. Transitions exhibiting the described behavior are known as transitions with a *static function hazard* [7], an unwanted switching behavior resulting from the logic circuit's function and thus unavoidable. The function hazard described here could potentially explain the dimmer final state $(1, 1, 0)$ observed in the lab.

Figure 3 compares two execution traces retrieved from the circuit's CTMC model. These traces are found by bounding the model's original infinite state-space and then running *Dijkstra's* algorithm on this finite state-space to return the trace with the highest probability. Trace A is found by searching for the highest probability trace that ends in the steady-state of the model and is forced to pass through a state where the output of the circuit glitches. Trace B is found by searching for the highest probability trace that ends in the steady-state of the model and is enforced to only pass through states where the production rate of the output reporter remains higher than its degradation rate, ensuring that the output remains stable during execution.

When the circuit's input changes from $(0, 0, 0)$ to $(1, 1, 0)$, the output of the purple NOT gate (*pAmtR*) and the blue NOR gate (*pSrpR*) changes from high to low and the output of the green NOR gate (*pPhIF*) changes from low to high. In trace A, the circuit first senses the change in the output of the purple NOT gate without yet sensing the change in the output of the green NOR gate. This results in the circuit's output to switch to low before the changes in the blue and green NOR gates are sensed and the circuit's output switches back to high again. Trace B shows an execution scenario where the circuit senses the change in the outputs of the blue and green NOR gates before it senses the change in the output of the purple NOT gate, and the circuit's output remains stable during this scenario. This comparison suggests that the circuit's glitching behavior is likely due to the order of the transitions observed in trace A, namely the circuit sensing the change in the output of the purple NOT gate before it senses the change in the green NOR gate. A potential model refinement that reduces the probability of this transition ordering is to slow down the path going through the purple NOT gate, e.g. by introducing buffers along this path.

This analysis is relevant to synthetic biology as it offers valuable insights for the redesign process of genetic circuits, especially in scenarios where combinational genetic circuits exhibit undesired switching variations (glitches). By utilizing this analysis, users can effectively modify the circuit layout or part selection to fine-tune the circuit and achieve robust behavior.

3 DISCUSSION

This abstract demonstrates how computational analysis can aid the design process of genetic circuits, enhancing their robustness and facilitating their application beyond the laboratory setting. By pinpointing the specific pathway within the circuit that leads to an output glitch for a particular input transition, the user can refine the circuit, thereby reducing the probability of observing glitches in the system's output. Looking ahead, these analysis results can be seamlessly integrated into a pipeline that automates the redesign process and part selection, providing users with detailed instructions and protocols for the build steps in the laboratory. Furthermore, this advancement can potentially extend to protocols for liquid handling robots, further streamlining and automating the synthetic biology workflow.

Acknowledgements

We thank all members of the FLUENT Verification Project (<https://fluentverification.github.io>) for their feedback. This work was supported by the National Science Foundation under Grants No. 1856740 and 1900542. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

- [1] BROOKS, S. M., AND ALPER, H. S. Applications, challenges, and needs for employing synthetic biology beyond the lab. *Nature Communications* 12, 1 (Mar. 2021), 1390.
- [2] BUECHERL, L., AND MYERS, C. J. Engineering Genetic Circuits: Advancements in Genetic Design Automation Tools and Standards for Synthetic Biology. *Current Opinion in Microbiology* (2022).
- [3] BUECHERL, L., ROBERTS, R., FONTANARROSA, P., THOMAS, P. J., MANTE, J., ZHANG, Z., AND MYERS, C. J. Stochastic Hazard Analysis of Genetic Circuits in iBioSim and STAMINA. *ACS Synthetic Biology* (Oct. 2021), aacsynbio.1c00159.
- [4] 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.
- [5] FONTANARROSA, P., DOOSTHOSSEINI, H., BORUJENI, A. E., DORFAN, Y., VOIGT, C. A., AND MYERS, C. Genetic Circuit Dynamics: Hazard and Glitch Analysis. *ACS Synthetic Biology* (2020), 15.
- [6] JONES, T. S., OLIVEIRA, S. M. D., MYERS, C. J., VOIGT, C. A., AND DENSMORE, D. Genetic circuit design automation with Cello 2.0. *Nature Protocols* (Feb. 2022).
- [7] MYERS, C. J. *Asynchronous circuit design*. John Wiley & Sons, 2001.
- [8] NEUPANE, T., MYERS, C. J., MADSEN, C., ZHENG, H., AND ZHANG, Z. STAMINA: Stochastic Approximate Model-Checker for INfinite-State Analysis. Computer Aided Verification, Springer International Publishing, pp. 540–549.
- [9] 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 (Apr. 2016), aac7341–aac7341.
- [10] SANCHEZ, A., CHOUBEY, S., AND KONDEV, J. Stochastic models of transcription: From single molecules to single cells. *Methods* 62, 1 (July 2013), 13–25.

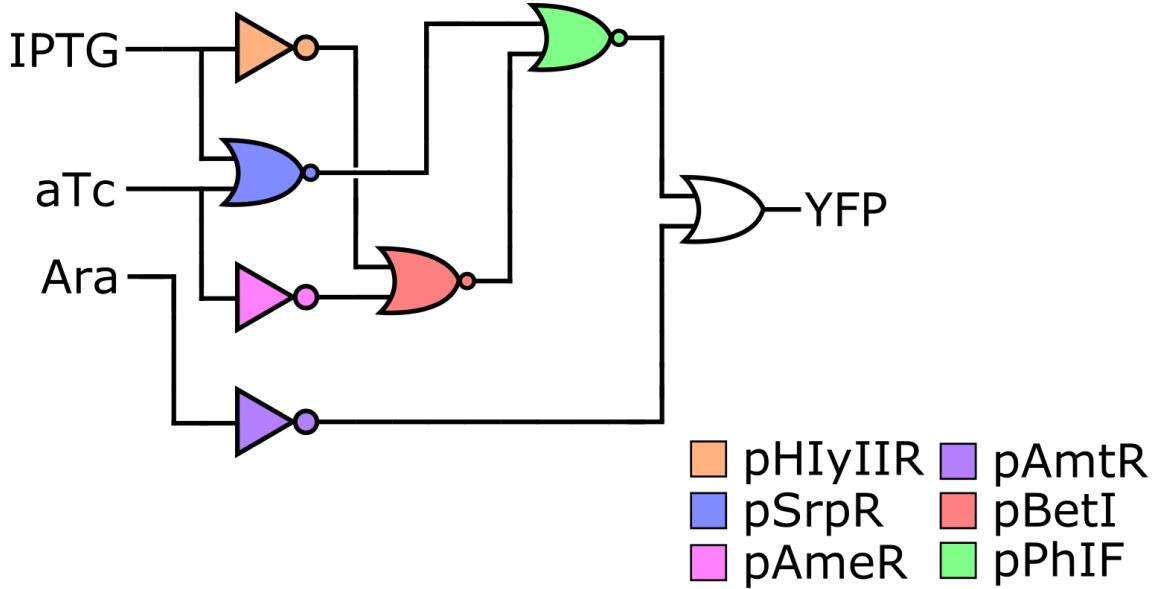


Figure 1: "Circuit 0xF6," a genetic circuit originally published by Nielsen et al. [9], comprises three NOT gates, three NOR gates, and one OR gate. The color coding connects the gates to their respective protein encoding. The three input molecules are IPTG, aTc, and Ara. The output is the fluorescent reporter YFP.

		aTc	IPTG	0 0	1 0	1 1	0 1
		Ara		0	1	1	1
Ara	0	0		1	1	1	1
		1		0	1	0	1

Figure 2: Karnaugh map of circuit 0xF6. The columns represent the absence or presence of Ara, while the rows represent the absence or presence of IPTG and aTc. Each cell indicates the presence or absence of YFP, with a *high* concentration denoted as 1 and a *low* concentration as 0. When transitioning from the initial state (0,0,0), the circuit can detect either the change in Ara or aTc first. If the change in Ara is detected first, the circuit transitions from (0,0,0) to (1,0,0) to (1,1,0) following the red arrow, exhibiting an unexpected decrease in YFP production. However, if aTc is detected first, the circuit transitions from (0,0,0) to (0,1,0) and then (1,1,0). According to the map, the state (0,10) also indicates YFP production. This is represented by the red arrow, showing the expected behavior.

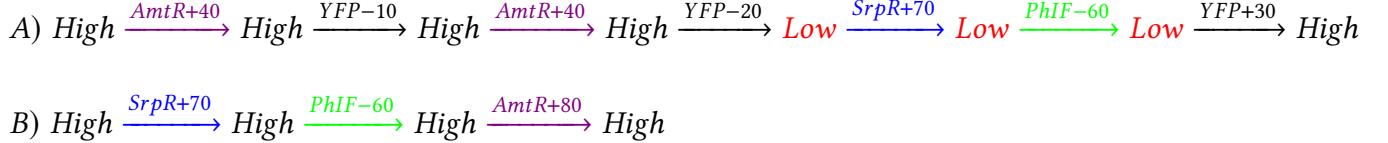


Figure 3: Comparing two possible execution scenarios for circuit 0xF6. Both traces are found by first bounding the circuit's infinite-state CTMC to a finite state-space and then running the Dijkstra's algorithm on this finite state-space. Each trace illustrates a series of transition firings along with the state of the circuit's output reporter (YFP) after each transition. In the original model, species' population change with a step of 10. To save space, consecutive identical transitions are grouped together. The transitions changing the population of HIyIIR, AmeR and BetI are not shown since they do not affect the output of the circuit. Trace A is found by identifying the shortest (highest probability) path to the steady-state of the model that passes through a state where degradation rate of the output reporter is greater than its production rate and the output glitch happens. Trace B is found by identifying the shortest (highest probability) path to the steady-state of the model and is enforced to only pass through states where the production rate of the output reporter remains higher than its degradation rate, ensuring with a high probability that the circuit's output remains stable during the input transition.

SeqImprove: Machine Learning Assisted Curation of Genetic Circuit Sequence Information

Jeanet Mante¹, Zach Sents¹, Duncan Britt¹, William Mo¹, Chris J. Myers¹

¹University of Colorado Boulder, Boulder, USA

chris.myers@colorado.edu

1 INTRODUCTION

Synthetic biology has vast potential applications in numerous fields. However, the progress and utility of synthetic biology are currently hindered by the lengthy process of studying literature and replicating poorly documented work. The reuse of genetic components is currently low [17]. More complete data records makes the data more reusable and the database to which they are submitted more valuable [7].

Much of the data that is submitted does not contain sufficient information for data reuse. The *Synthetic Biology Knowledge System* (SBKS) attempted to address this problem by creating an integrated knowledge system built using data generated with post-hoc curation [6]. The curation consisted of two parts: (1) text mining to perform automatic annotation of the articles using *natural language processing* (NLP) to identify salient content such as key terms, relationships between terms, and main topics [9]; and (2) a data mining pipeline that performs automatic annotation of the sequences extracted from the supplemental documents with the genetic parts used in them [8]. The curation allows the linkage of knowledge, genetic parts, and the context in which they are used to the papers describing their usage. In order to process vast amounts of data, automated tools are employed to analyze unstructured text and identify relevant keywords, while attempting to derive their intended meaning from the surrounding context. This tests the limits of NLP methods, such as *named entity recognition* (NER) and *entity classification* [3]. Furthermore, sequences provided as supplemental information in publications are typically poorly annotated, incomplete, and provided in non-machine accessible formats (e.g. PDFs). The SBKS project demonstrated that reconstruction of important design information through post-hoc curation is extremely noisy and error prone [6, 8].

The idea of author based curation (having the submitters curate their own data) is becoming increasingly popular [18], and it would help address the issues encountered by the SBKS project. Author curation requires intuitive interfaces to ensure standardization and completeness in their metadata. We developed the SeqImprove curation interface to enable authors to curate machine generated metadata and annotations and save this in a machine accessible format. This paper presents the capabilities and underlying architecture of SeqImprove.

2 RESULTS

SeqImprove is designed to aid authors in creating machine accessible sequence data with complete metadata. It consists of a user-interface that was built using modular code. It can be reused by others to work as the front-end for their curation software. Additionally, the back-end consists of a series of tools that automate NER, *named entity normalization* (NEN), sequence annotation, and protein prediction. The functions are accessed by users via the front end. The backend has two main machine aided curation functions:

- (1) **Annotate Sequence:** This is the method used to suggest sequence annotations. It is based on SYNBICT [14]. It uses the feature libraries found in our Github Repository. These include libraries from parts-rich papers [1, 5, 11, 12].
- (2) **Annotate Text:** This method is used to suggest keyword annotations. It uses BERN2 for NER and NEN [16]. Additional fuzzy matches are carried out to catch potential misspellings using the fast-fuzzy package.

The first step is sequence data input using an existing sequence file in the *Synthetic Biology Open Language* (SBOL) [13] format, a link to a sequence stored in SynBioHub [10], or a FASTA file. It is also capable of providing an empty template for the user to manually copy-and-paste a DNA sequence of interest. Next, it takes authors through four sections of metadata.

The first section, as shown in Figure 1, provides the description of the part with hyperlinks for recognized terms, allows users to select the role or function of the sequence via a drop down menu of *sequence ontology* (SO) terms [4], designate any target organisms of sequence insertion in machine accessible formatting based on the *National Center for Biotechnology Information* (NCBI) Taxonomy [15], and link relevant papers or pre-prints using a DOI.

The second section, as shown in Figure 2, displays the sequence annotated with sub-components. The “Analyze Sequence” button can be used to generate suggestions of sub-components based on a SeqImprove’s library of frequently used components. The suggestions may be accepted by selecting the checkbox next to the sub-component’s name. Alternatively, the user can manually add and label their own annotations.

The third section, as shown in Figure 3, is where the description can be edited and machine accessible keywords are selected within the description. The “Analyze Text” button uses machine learning to suggest keywords, group similar ones together, and suggest a machine accessible ontology term for the keyword. Like in the previous section, users can approve a keyword with checkboxes, or manually add annotations using the “Create Text Annotation” button.

In the final section, as shown in Figure 4, proteins produced by the sequence are added to the metadata. There is a suggestion box where proteins frequently associated with keywords or sequence annotations are provided. For example, *E. coli* in the description field leads to the suggestion of common *E. coli* proteins. The user can also add further proteins from the UniProt database [2].

3 DISCUSSION

We have presented SeqImprove, a platform for machine-assisted author curation of genetic sequences. SeqImprove helps authors submit sequence data and associated metadata in machine accessible formats. It prompts authors to consider metadata such as role, target organism, reference papers, subsequences, protein production, and keywords. It makes the information machine accessible by using existing ontologies to structure the metadata. Authors are helped by suggestions of keywords, proteins, and sequence annotations. They can review and edit the suggestions in a user friendly interface. The interface was also designed to be modular so it could be reused for similar curation in other contexts.

While SeqImprove offers many benefits, there are still limitations to the system for future work to address. The most important one is author participation. SeqImprove only works if authors use it, and it can be difficult to incentivize researchers to participate. Since the top benefit for researchers is faster searching for sequences that others have curated and additional citations for their own, the results of additional effort in curation are indirect. This is particularly the case initially, as there will be little well curated output data to be utilized. This reduces the incentives for researchers to adopt the system, and without adoption, the available data remains scarce. Breaking the initial consensus threshold will be difficult, but would be aided by journal incentives.

Acknowledgements

We thank all members of the Genetic Logic Lab at CU Boulder. This work was supported by the National Science Foundation under Grant No. 1939892 and 2231864 and the CU Summer Program for Undergraduate Research. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

- [1] Addgene: CIDAR MoClo Extension, Volume I.
- [2] CONSORTIUM, T. U. Uniprot: a worldwide hub of protein knowledge. *Nucleic Acids Research* 47, D1 (Jan 2019), D506–D515.
- [3] CRICHTON, G., PYYSALO, S., CHIU, B., AND KORHONEN, A. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics* 18, 368 (2017).
- [4] EILBECK, K., LEWIS, S. E., MUNGALL, C. J., YANDELL, M., STEIN, L., DURBIN, R., AND ASHBURNER, M. The sequence ontology: a tool for the unification of genome annotations. *Genome biology* 6, 5 (2005), R44.
- [5] LEE, M. E., DELOACHE, W. C., CERVANTES, B., AND DUEBER, J. E. A Highly Characterized Yeast Toolkit for Modular, Multipart Assembly. *ACS Synthetic Biology* 4, 9 (Sept. 2015), 975–986. Publisher: American Chemical Society.
- [6] MANTE, J., HAO, Y., JETT, J., JOSHI, U., KEATING, K., LU, X., NAKUM, G., RODRIGUEZ, N. E., TANG, J., TERRY, L., WU, X., YU, E., DOWNIE, J. S., MCINNES, B. T., NGUYEN, M. H., SEPULVADO, B., YOUNG, E. M., AND MYERS, C. J. Synthetic biology knowledge system. *ACS Synthetic Biology* (Aug 2021).
- [7] MANTE, J., AND MYERS, C. J. Advancing reuse of genetic parts: progress and remaining challenges. *nature communications* 14, 1 (2023), 2953.
- [8] MANTE, J. V. *Promotion of Data Reuse in Synthetic Biology*. PhD thesis, CU Boulder, Boulder Colorado, 2022.
- [9] MCINNES, B. T., DOWNIE, J. S., HAO, Y., JETT, J., KEATING, K., NAKUM, G., RANJAN, S., RODRIGUEZ, N. E., TANG, J., XIANG, D., YOUNG, E. M., AND NGUYEN, M. H. Discovering content through text mining for a synthetic biology knowledge system. *ACS Synthetic Biology* 11, 6 (2022), 2043–2054. PMID: 35671034.
- [10] 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.
- [11] MISIRLI, G., HALLINAN, J., POCOCK, M., LORD, P., McLAUGHLIN, J. A., SAURO, H., AND WIPAT, A. Data integration and mining for synthetic biology design. *ACS synthetic biology* 5, 10 (Oct 2016), 1086–1097.
- [12] OBST, U., LU, T. K., AND SIEBER, V. A Modular Toolkit for Generating *Pichia pastoris* Secretion Libraries. *ACS Synthetic Biology* 6, 6 (June 2017), 1016–1025.
- [13] ROEHRER, N., BEAL, J., CLANCY, K., BARTLEY, B., MISIRLI, G., GRÜNBERG, R., OBERORTNER, E., POCOCK, M., BISSELL, M., MADSEN, C., NGUYEN, T., ZHANG, M., ZHANG, Z., ZUNDEL, Z., DENSMORE, D., GENNARI, J. H., WIPAT, A., SAURO, H. M., AND MYERS, C. J. Sharing structure and function in biological design with sbol 2.0. *ACS Synthetic Biology* 5, 6 (Jun 2016), 498–506.
- [14] ROEHRER, N., MANTE, J., MYERS, C. J., AND BEAL, J. Synthetic biology curation tools (SYNBICT). *ACS Synthetic Biology*.
- [15] SCHOCH, C. L., CIUFO, S., DOMRACHEV, M., HOTTON, C. L., KANNAN, S., KHOVANSKAYA, R., LEIPE, D., McVEIGH, R., O’NEILL, K., ROBBERTSE, B., AND ET AL. Ncbi taxonomy: a comprehensive update on curation, resources and tools. *Database: The Journal of Biological Databases and Curation* 2020 (Jan 2020), baaa062.
- [16] SUNG, M., JEONG, M., CHOI, Y., KIM, D., LEE, J., AND KANG, J. Bern2: an advanced neural biomedical named entity recognition and normalization tool. *Bioinformatics* 38, 20 (Oct 2022), 4837–4839.
- [17] VILANOVA, C., AND PORCAR, M. igem 2.0—refoundations for engineering biology. *Nature Biotechnology* 32, 5 (May 2014), 420–424.
- [18] ZAVERI, A., HU, W., AND DUMONTIER, M. Metacrowd: Crowdsourcing biomedical metadata quality assessment. *Human Computation* 6 (Sep 2019), 98–112.

Test_Part 

OVERVIEW **SEQUENCE** **TEXT** **PROTEINS**

Description

This part makes [GFP](#). It was tested in [E. coli](#) and is thought to work in [B. subtilis](#). The circuit functions better when background levels of [lactose](#) are low. [E. coli](#) is the same as [Escherichia coli](#). Escichia coi.

Roles

SO:0000804

Role	Engineered Region	
------	-------------------	-------------------------------------------------------------------------------------

Add Role

Target Organisms

Escherichia coli 

Search Taxonomy... 

Prioritize parents in search

References

Enter a DOI 

Figure 1: Overview Tab. Shows description with hyperlinks, role selection menu, target organisms with search, and references.

The Sequence Tab displays a DNA sequence in a text editor. To the right, a panel titled "Sequence Annotations" lists three entries: "AmpR terminator" (pink square), "pRPL18B" (yellow square), and a button labeled "Analyze Sequence".

```
actttca tactccg ccacaggt ggcacttt tcggggaa atgtgcgc ggaacccc  
tattttttt tatttttc taaataca ttcaaata tgtatccg ctcatgag acaataac  
cctgataa atgcttca ataataattt gaaaaagg aagagttt cagagaag aaaccaat  
tgtccata ttgcatca gacattgc cgtaactg cgctttt actggctc ttctcgct  
aaccacac cggtaacc ccgttat taaaagca ttctgtaa caaagcgg gaccaag
```

Figure 2: Sequence Tab. Shows file sequence and annotations of parts within sequence.

The Text Tab contains a "Description" section with a text editor. It describes a part that makes GFP, was tested in E. coli, and is thought to work in B. subtilis. The circuit functions better when lactose levels are low. A note states that E. coli is the same as Escherichia coli. To the right, a panel titled "Recognized Terms" lists four entries with checkboxes: GFP (NCBIGene:8790), E. coli (NCBITaxon:562), B. subtilis (NCBITaxon:518697), and lactose (mesh:D007785). Buttons for "Create Text Annotation" and "Analyze Text" are also present.

This part makes GFP. It was tested in E. coli and is thought to work in B. subtilis. The circuit functions better when background levels of lactose are low. E. coli is the same as Escherichia coli. Escichia coi.

Figure 3: Text Tab. Shows description and various keyword annotations.

The Proteins Tab shows a list of associated proteins: Green fluorescent protein (Aequorea victoria) with a delete icon, and a search bar for UniProt. To the right, a panel titled "Suggested Proteins" lists several entries with "+" icons and "KEYWORD" buttons: Pro-opiomelanocortin (Homo sapiens), Type I restriction enzyme EcoR124II endonuclease subunit (Escherichia coli), SpolVB peptidase (Bacillus subtilis (strain 168)), Lactose permease (Escherichia coli (strain K12)), and another entry for Pro-opiomelanocortin (Homo sapiens).

Figure 4: Proteins Tab. Shows associated proteins and suggested additional ones.

A Report on SynBio Data Management Practices

Carolus Vitalis*

Sai P. Samineni*

University of Colorado Boulder
Boulder, United States
carolus.vitalis@colorado.edu
sasa6749@colorado.edu

Chris J. Myers

University of Colorado Boulder
Boulder, United States
chris.myers@colorado.edu

Pedro Fontanarrosa

University College London
London, United Kingdom
pfontanarrosa@gmail.com

1 INTRODUCTION

Robust data management practices are crucial to promote a reproducible *design-build-test-learn* (DBTL) cycle for synthetic biology (SynBio) applications [4]. These data management practices are built upon a set of software tools to capture information, data standards to encode the information in machine-readable formats, and digital repositories to support data sharing.

This abstract presents the analysis of data management practices based on interviews conducted with 25 lab collaborators from the Army Center for Synthetic Biology project. In particular, this abstract presents an overview of the data types, the software tools, the data storage solutions, and the major insights and challenges the labs face in the absence of a uniform data management practices. Additionally, this abstract offers suggestions for how the uniform data management practices may streamline data storage, sharing, and analysis processes to enhance research efficiency and collaboration. Finally, this abstract aims to identify common trends and challenges in data management within research environments and provide recommendations for improving data management practices in future efforts to build a uniform data management plans for multi-lab collaborations to advance SynBio applications.

2 METHODS

This research involved interviews with multiple research labs to gather information about their data management practices. The interviews were conducted via Zoom and lasted approximately one hour. The interviewees were given a document with the questions in advance; the questions were the following:

- (i) **Data Inputs**—(1) What kind of data do you consume/ require for your research? Is it sequence, -omics, circuit design, or other? (2) Where does that data come from? Is it from a database, shared Excel files, or publications? (3) What data types do you wish were more readily accessible? Are there particular formats for these data types that would greatly help your research? (4) What problems have you faced acquiring data? (i.e., Maybe

it was not in the format you wanted it to be, or it was difficult to access/edit) and (5) Do you obtain most of your data in one setting or condition (i.e., lab), or are there multiple settings as the source of your data (i.e., field, lab, other out of the lab)?

- (ii) **Data Processing**—(6) What kind of data processing occurs in your workflow? What kind of metadata do you produce/record? (7) What software tools do you use (if any)?
- (iii) **Data Output/Publication**—(8) In what format do you usually publish your results? Strings (text), Excel files, SBOL, or other data standards. (9) Where do you publish your data? For example, in a database using data standards, in an Excel file, in PDF, in publications, etc.

These questions allowed for discussions covering topics such as data types, software tools, data storage solutions, and challenges faced by the labs.

3 RESULTS

Data Types—The labs deal with diverse data types, including genetic designs, images, fluorescence measurements, molecular characterization data, assembly plans, and other experimental protocols. Common file formats include FASTA for genetic sequences, GenBank for genome information, TIFF for images, and Excel for fluorescence measurements. A complete list of the data types is shown in Table 1.

Software Tools—Excel, GitHub, Zenodo, and Jupyter Notebooks were commonly used for data analysis, storage, and management. Specialized tools like USearch, SPADES, anti-SMASH, Bowtie2, and antifungal were employed for specific data analysis tasks. MATLAB, Geneious, and Python scripts were also used for image processing, sequence analysis, and data manipulation.

Data Storage—The labs utilized various tools for data storage, including Microsoft Teams, OneDrive, Dropbox, Zenodo, and internal cloud storage. NCBI and the Protein Data Bank were used for data deposition and sharing. Some labs employed specific platforms like the CAMII Biobank for microbial strain collections.

Challenges—Common challenges faced by the labs included incomplete or missing metadata, non-standardized data presentation, and the need for better data sharing and

*Both authors contributed equally to this research.

visualization platforms. The volume of data varied across labs, with one lab reporting approximately 10 TB per experiment leading to many petabytes of data accumulated over 15 years.

Based on our findings, the following recommendations are suggested to enhance data management practices:

1. Encourage using standardized data formats: Standardized formats such as the *Synthetic Biology Open Language* (SBOL) [3], SBOL Visual [8], LabOp [1], the *Systems Biology Markup Language* (SBML) [5], etc., should be promoted to ensure data compatibility and interoperability across different labs and tools.

2. Establish a centralized data repository: Platforms like SynBioHub [7] and Flapjack [2] can create a unified location for storing, organizing, and sharing data within and across labs, promoting easy access and collaboration.

3. Provide training sessions on data management practices: Training sessions and resources should be offered to lab members to enhance their data management, sharing, and analysis skills. This will promote the adoption of best practices and ensure that researchers have the necessary skills to manage research data effectively.

Our proposed data management workflow is shown in more detail in Figure 1. SynBioHub serves as a central platform for connecting computational tools, experimental planning, and genetic repositories. The cycle starts with the *store and share stage*, where the parts relevant to a given project are characterized, and their information and sequences are stored in an Excel sheet, which is then converted to SBOL format and uploaded to SynBioHub using the Excel-to-SBOL Converter [6]. Other types of data can also be uploaded to platforms such as NCBI, PDB, and Genbank, enabling sharing and collaboration. Next, in the *design stage*, SynBioHub is used to access the information and make informed decisions about the genetic designs, assembly plans, and sequence verification, which are then uploaded back to SynBioHub. Then, in the *build stage*, SynBioHub aids the generation of protocols, automated models, and robotics scripts. After the genetic devices are assembled, the *test stage* starts, where metadata, measurements, and images are captured; for experimental data and metadata captured in Excel sheets, the Experimental Data Connector (XDC) [9] is used to upload them to Flapjack and SynBioHub, respectively, employing a template that simplifies this process for researchers. Finally, in the *learn stage*, modeling and statistical analysis are performed, which retrieve the data from SynBioHub and Flapjack to provide feedback and allow modifications and improvements to the devices.

To make this possible, the standard formats shown in Table 1 must be adopted. This table indicates the proposed standards to be used for the different stages mentioned above. In the case of the *part libraries*, we have the characterized

parts, which would be stored in SynBioHub in SBOL format. For the *design stage*, we have the genetic designs and the assembly plans, where both would be stored in SynBioHub, the former in SBOL or SBOL Visual format, depending on the information to be represented, and the latter in LabOp format. Moving on to the *build stage*, we have the experimental protocols, which would also be stored in SynBioHub in LabOp format, as well as the design automated models, which correspond to robotics scripts and would be stored in GitHub. For the *test stage*, we have the measurements and metadata that would be in Excel template format and are stored in Flapjack and SynBioHub, respectively. Finally, in the *learn stage*, we have the scripts that are proposed to be stored as Docker Images stored in GitHub and Docker Hub. Then, the statistical analyses would be saved in SBOL format if they have as output a new characterization of a part or in an Excel template if they correspond to an analysis of the data, and would be saved in SynBioHub or Flapjack, respectively. Finally, we have the modeling that can be stored in SynBioHub in SBML format.

4 DISCUSSION

Several steps are necessary to further improve and refine this data management process. First, more detailed evaluations of the software tools employed by the laboratories should be conducted to determine a smooth transition to this workflow and improve adoption. Second, based on the insights from this survey, a data management guide should be created to provide possible recommendations on best data capture, storage, and analysis practices and help promote effective data management practices. Third, collaborating with more institutions will provide further insights into data management practices and help identify common trends and challenges in a broader range of research environments. Fourth, more research is needed to understand existing data management challenges and develop targeted solutions to address them.

In summary, the recommendations presented here will enable data compatibility and facilitate data access, which are essential for effective collaboration. Taken together, these recommendations will result in a data management workflow that promotes collaborative and reproducible research.

Acknowledgments

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-22-2-0210. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

REFERENCES

- [1] BARTLEY, B., ET AL. Building an open representation for biological protocols. *J. Emerg. Technol. Comput. Syst.* 19, 3 (jun 2023).
- [2] FELIÚ, G. Y., ET AL. Flapjack: Data management and analysis for genetic circuit characterization. *ACS Syn Bio* (2020).
- [3] 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.
- [4] JESSOP-FABRE, M. M., AND SONNENSCHEIN, N. Improving Reproducibility in Synthetic Biology. *Frontiers in Bioeng. and Biotech.* 7 (2019).
- [5] KEATING, S. M., ET AL. SBML level 3: an extensible format for the exchange and reuse of biological models. *Molecular Systems Biology* 16, 8 (2020-08), e9110. Publisher: John Wiley & Sons, Ltd.
- [6] MANTE, J., ET AL. Excel-sbol converter: Creating sbol from excel templates and vice versa. *ACS Syn Bio* 12, 1 (2023), 340–346. PMID: 36595709.
- [7] MC LAUGHLIN, J. A., ET AL. SynBioHub: A standards-enabled design repository for synthetic biology. *ACS Syn Bio* 7, 2 (2018), 682–688.
- [8] QUINN, J., ET AL. SBOL visual: A graphical language for genetic designs. *PLOS Biology* 13, 12 (12 2015), 1–9.
- [9] SAMINENI, S. P., VIDAL, G., ET AL. Experimental data connector (XDC): Integrating the capture of experimental data and metadata using standard formats and digital repositories. *bioRxiv* (2022).

Table 1: Summary of the data types and formats involved in different stages of the synthetic biology DBTL cycle. The stages are Part Libraries, Design, Build, Test, and Learn, as shown in the first column. The second column lists the data types that need to be standardized for each stage. The third and fourth columns show the current and proposed standard formats for the data types, respectively. The last column indicates the machine-readable data storage platform that can be used to store the data in the standard format. By adopting these standards, laboratories can enhance their research efficiency, quality, and collaboration.

Stage	Data Types	Current Formats Used	Standard Formats	Machine-readable Data Storage
Part Libraries	Characterized Parts	Excel Sheets	SBOL	SynBioHub
	Genetic Designs (Sequences, Proteins, etc.)	FASTA GenBank	SBOL SBOL Visual	SynBioHub
Design	Assembly Plans	Plain Text	LabOp	SynBioHub
	Experimental Protocols	Plain Text	LabOp	SynBioHub
Build	Design Automated Models	Python	Robotics Scripts	GitHub
	Measurements	Excel Sheets	Excel Template	Flapjack
Test	Metadata	Excel Sheets	Excel Template	SynBioHub
	Scripts	Commented Scripts	Docker Images	GitHub Docker Hub
Learn	Statistical Analysis	Excel Sheets	SBOL Excel Template	SynBioHub Flapjack
	Modeling	Python Notebooks MATLAB	SBML	SynBioHub

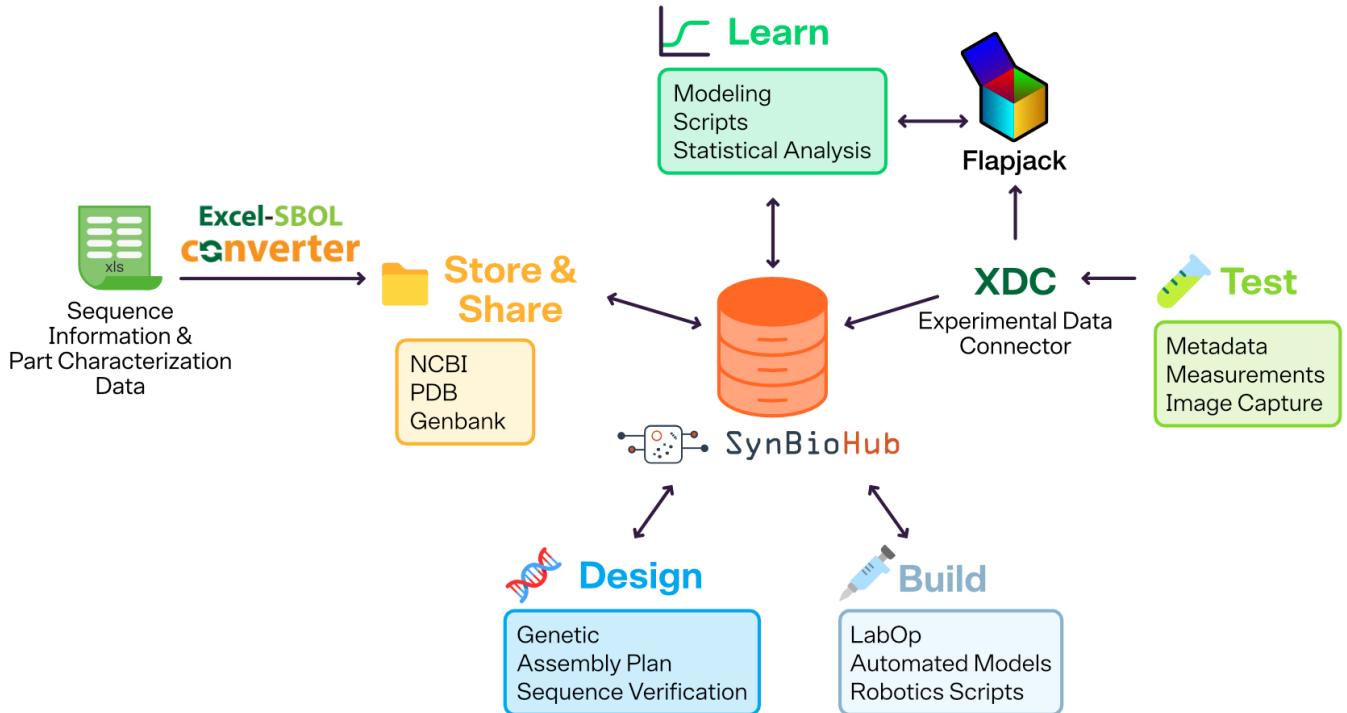


Figure 1: Synthetic Biology workflow enabled by SynBioHub. In this workflow, parts are characterized, and information is collected in a spreadsheet file, which can be converted to SBOL format and uploaded to SynBioHub. In the *design step*, various software tools that can interact with SynBioHub are used to create and modify designs. These tools can access data from SynBioHub and upload new designs to it. The same applies to the *build step*, which can be done manually or automatically using different assembly methods. In the *test step*, the experimental data and metadata are uploaded to Flapjack and SynBioHub, respectively, using the XDC. The *learn step* involves modeling and statistical analysis that can use data from SynBioHub and provide feedback to improve design decisions. All the data uploaded to SynBioHub can be accessed via a static URL. This workflow demonstrates the collaborative aspect of SynBioHub. Data transfers via API are also possible and facilitate each step of the design process, enabling automated data upload and download.

Software for Synthetic Biology Workflows: How to Improve Your Productivity and Impact

Chris J. Myers¹, Lukas Buecherl¹, Daniel Fang¹, Pedro Fontanarrosa¹, Jeanet Mante¹, William Mo¹,
Sai P. Samineni¹, Gonzalo Vidal², Carolus Vitalis¹, Guillermo Yanez-Feliu², Timothy J. Rudge²

¹University of Colorado Boulder, Boulder, USA

² Newcastle University, Newcastle upon Tyne, United Kingdom

chris.myers@colorado.edu

1 INTRODUCTION

In 2011, the authors of [7] highlighted the fact that reproducibility in the field of synthetic biology was being hampered by the lack of data sharing, particularly of the DNA sequences used in genetic designs. This observation led to the formation of efforts to develop the *Synthetic Biology Open Language* (SBOL) to facilitate data sharing [1], and *SBOL Visual* to provide consistency in genetic circuit diagrams [8]. The standards along with the requirements of data management plans by funding agencies and sequence submission requirements by some journals has led to some progress. Unfortunately, uncertainty about how policies should be implemented and how they should be incentivized and enforced has seen limited progress [4]. As a result, synthetic biology data is still not *FAIR* (findable, accessible, interoperable, and reusable) impeding progress in synthetic biology [4].

In order to overcome this challenge, complete synthetic biology workflows must be developed that provide solutions that easily integrate into existing processes for each step of the *Design-Build-Test-Learn* (DBTL) cycle. This abstract presents one such workflow, and we hope to inspire the IWBDA community of developers to create additional software solutions that can plug into this workflow to further improve productivity and impact of synthetic biologists.

2 RESULTS

The Synthetic Biology: Engineering, Evolution & Design (SEED) Conference each year has hosted workshops on software to support synthetic biology workflows. The last two years, we have presented a complete example synthetic biology workflow that includes software solutions for each step of the DBTL cycle using tools developed by the Genetic Logic Lab led by Myers at CU Boulder and Rudge Lab at Newcastle University. The workflows that we have presented are not meant to be restricted to only this set of tools, but instead they demonstrate that when standards are utilized that a seamless DBTL workflow can be constructed. If other tools become available that support the standards being employed here, these tools can be integrated into this workflow. The proposed workflow follows the DBTL cycle, as depicted in Figure 1.

The first step of the workflow is to select the genetic parts needed for a given genetic design from databases like SynBioHub [5]. SynBioHub utilizes the SBOL data standard to facilitate the storing and sharing of genetic design information. SynBioHub is utilized at each step of this workflow to share the results from each step between the software tools. If the required parts are not already found in SynBioHub, they can be easily uploaded using a variety of formats (SBOL, GenBank, FASTA, GFF3, Snapgene, etc.), and they can be also uploaded using the Excel-SBOL Converter tools [3].

The design and modeling step is facilitated by *genetic design automation* (GDA) tools [6], like Cello [2], iBioSim [13], LOICA [12], SBOLCanvas[11], and SynBioSuite [10]. SynBioSuite, specifically, is a cloud-based tool that enables users to design the layout of their genetic circuit, import relevant part information from SynBioHub collections, and simulate the behavior of the design to ensure its proper functioning.

In the Build step, tools like SBOLDesigner [15] can be utilized to connect a backbone vector to transcriptional units, preparing them for ordering from DNA synthesis companies. Moreover, this tool can specify combinatorial designs, allowing users to create many design variations. Subsequently, if there is a need to assemble multiple transcriptional units into a single plasmid, the Python package PUDU can be employed to generate protocols for liquid handling robots, leveraging the SBOL files of the genetic designs. PUDU supports test plate setup, calibration, and common cloning protocols, including DNA assembly. These protocols can be conveniently customized and tailored to accommodate diverse laboratory requirements, effectively lowering the entry barriers for novice students and researchers in utilizing the OpenTrons OT-2 platform.

The test step is supported by the *Experimental Data Connector* (XDC) [9]. The input to this tool is an Excel spreadsheet template that enables the capture of experimental metadata and measurements. XDC transforms the experimental metadata into the SBOL standard and uploads it to SynBioHub. The measurement data is uploaded to the Flapjack data repository [14] for further processing during the learn step.

During the learn step, the measurement data is examined using the software tool Flapjack [14], a platform that

provides a comprehensive suite of features for managing, analyzing, and visualizing experimental results. By utilizing the Flapjack platform, researchers can effectively integrate the test phase with the build and learn phases, facilitating the characterization and optimization of genetic circuits.

At the conclusion of each step, the gathered data is stored in a standardized format and uploaded to SynBioHub. As a result, at the end of each cycle, a comprehensive collection of data is accumulated, including metadata, designs, mathematical models, simulation results, and experimental findings. This facilitates seamless sharing within the community, fostering adherence to the FAIR principles.

3 DISCUSSION

The synthetic biology workflow presented here provides an example partial solution to enable researchers to capture and store essential information in a standardized and structured manner. This workflow ensures consistency and coherence in the documentation of experimental protocols, genetic sequences, modeling parameters, and analysis results. It is a partial solution, since sharing the data alone does not address all problems. Namely, best practices for calibration of the data along with complete descriptions of the experimental methods must also be developed. Continued development of these best practices along with additional tools that can integrate into this workflow will enable researchers to share their data in a FAIR manner accelerating the progress of the synthetic biology field.

Acknowledgements

We would like to thank all members (past and present) of the Genetic Logic Lab at CU Boulder and Rudge's Lab at Newcastle University that contributed to the construction, critique, and refinement of the tools presented here. We would also like to thank the attendees of our tutorial workshops at SEED over the years for their extremely valuable feedback.

This work was supported by numerous grants from the National Science Foundation, DARPA, NIST, and the US Army. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

- [1] GALDZICKI, M., CLANCY, K. P., OBERORTNER, E., POCOCK, M., QUINN, J. Y., RODRIGUEZ, C. A., ROEHNER, N., WILSON, M. L., ADAM, L., ANDERSON, J. C., BARTLEY, B. A., BEAL, J., CHANDRAN, D., CHEN, J., DENSMORE, D., ENDY, D., GRUNBERG, R., HALLINAN, J., HILLSON, N. J., JOHNSON, J. D., KUCHINSKY, A., LUX, M., MISIRLI, G., PECCOUD, J., PLAHAR, H. A., SIRIN, E., STAN, G.-B., VILLALOBOS, A., WIPAT, A., GENNARI, J. H., MYERS, C. J., AND SAURO, H. M. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology* 32, 6 (2014), 545–550.
- [2] JONES, T. S., OLIVEIRA, S. M. D., MYERS, C. J., VOIGT, C. A., AND DENSMORE, D. Genetic circuit design automation with Cello 2.0. *Nature Protocols* (Feb. 2022).
- [3] MANTE, J., ABAM, J., SAMINENI, S. P., POTZSCH, I. M., BEAL, J., AND MYERS, C. J. Excel-SBOL converter: Creating SBOL from Excel templates and vice versa. *ACS Synthetic Biology* 12, 1 (2023), 340–346. PMID: 36595709.
- [4] MANTE, J., AND MYERS, C. J. Advancing reuse of genetic parts: Progress and remaining challenges. *Nature Communications* 14, 1 (May 2023), 2953.
- [5] 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.
- [6] MYERS, C. J., BARKER, N., KUWAHARA, H., JONES, K., MADSEN, C., AND NGUYEN, N.-P. D. Genetic design automation. In *Proceedings of the 2009 International Conference on Computer-Aided Design* (New York, NY, USA, 2009), ICCAD '09, Association for Computing Machinery, p. 713. doi:10.1145/1542256.1542316
- [7] PECCOUD, J., ANDERSON, J. C., CHANDRAN, D., DENSMORE, D., GALDZICKI, M., LUX, M. W., RODRIGUEZ, C. A., STAN, G.-B., AND SAURO, H. M. Essential information for synthetic DNA sequences. *Nature Biotechnology* 29, 1 (2011), 22–22.
- [8] QUINN, J. Y., COX, III, R. S., ADLER, A., BEAL, J., BHATIA, S., CAI, Y., CHEN, J., CLANCY, K., GALDZICKI, M., HILLSON, N. J., LE NOVERE, N., MAHESHWARI, A. J., McLAUGHLIN, J. A., MYERS, C. J., P, U., POCOCK, M., RODRIGUEZ, C., SOLDATOVA, L., STAN, G.-B. V., SWAINSTON, N., WIPAT, A., AND SAURO, H. M. SBOL Visual: A graphical language for genetic designs. *PLOS Biology* 13, 12 (2015), 1–9.
- [9] SAMINENI, S. P., VIDAL, G., VITALIS, C., FELIU, G. Y., RUDGE, T. J., MYERS, C. J., AND MANTE, J. Experimental Data Connector (XDC): Integrating the Capture of Experimental Data and Metadata Using Standard Formats and Digital Repositories. *ACS Synthetic Biology* 12, 4 (Apr. 2023), 1364–1370.
- [10] SENTS, Z., STOUGHTON, T. E., BUECHERL, L., THOMAS, P. J., FONTANARROSA, P., AND MYERS, C. J. SynBioSuite: A Tool for Improving the Workflow for Genetic Design and Modeling. *ACS Synthetic Biology* 12, 3 (Mar. 2023), 892–897.
- [11] TERRY, L., EARL, J., THAYER, S., BRIDGE, S., AND MYERS, C. J. SBOLCanvas: a visual editor for genetic designs. *ACS Synthetic Biology* 10, 7 (2021), 1792–1796.
- [12] VIDAL, G., VITALIS, C., AND RUDGE, T. J. Loica: Integrating models with data for genetic network design automation. *ACS Synthetic Biology* 11, 5 (2022), 1984–1990.
- [13] 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* 8, 7 (July 2019), 1560–1563.
- [14] YÁÑEZ FELIU, G., EARLE GÓMEZ, B., CODOCEO BERROCAL, V., MUÑOZ SILVA, M., NUÑEZ, I. N., MATUTE, T. F., ARCE MEDINA, A., VIDAL, G., VITALIS, 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. PMID: 33382586.
- [15] ZHANG, M., McLAUGHLIN, J. A., WIPAT, A., AND MYERS, C. J. SBOLDesigner 2: An Intuitive Tool for Structural Genetic Design. *ACS Synthetic Biology* 6, 7 (July 2017), 1150–1160.

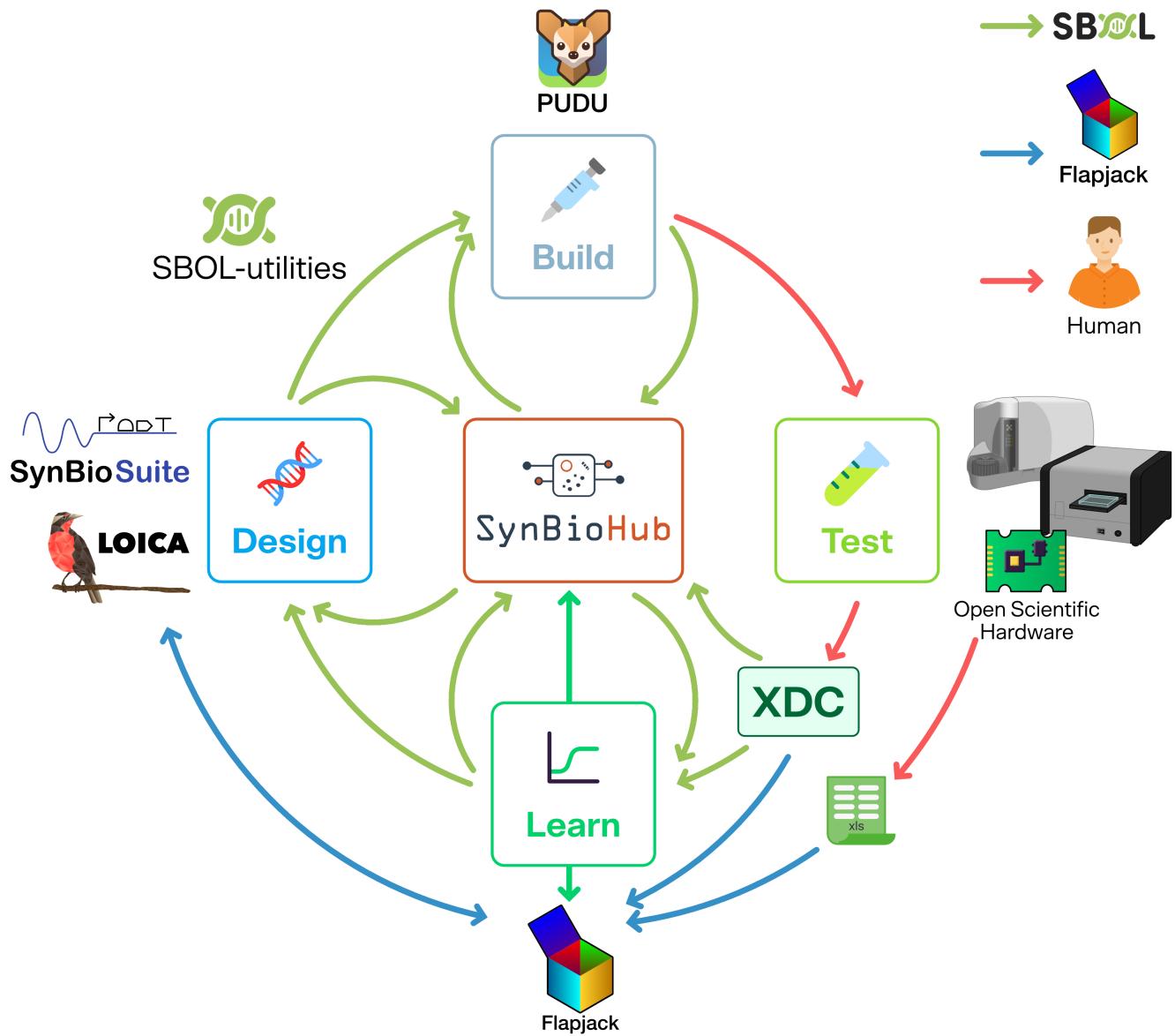


Figure 1: The proposed synthetic biology workflow. Data repositories like SynBioHub are used by the user to find information and select genetic parts for their project. Design and modeling tools like SynBioSuite and LOICA are used to design and simulate the behavior of the desired genetic design. During the build step, tools like SBOLDDesigner and PUDU can be used to ready the design for assembly by adding a backbone and to automatically create an assembly protocol for a liquid handling robot, respectively. During testing, XDC captures all experimental data in a machine readable format to foster accessible sharing and reuse amongst researchers. Learning is supported by Flapjack, providing tools for experimental data storage and analysis. The seamless integration of the tools is possible due to the incorporation of standards like SBOL. All tools shown are open source, and their links can be found in the additional information.

DNA Editing Game (DEGA) Theory

Nicholas Roehner¹

¹Raytheon BBN

nicholas.roehner@rtx.com

1 INTRODUCTION

DNA editing technologies such as CRISPR/Cas-based systems hold great promise for treating diseases, enhancing agricultural crops, and enabling biological information storage, but this promise is tempered by the potential for their unchecked misuse. Thus far, biodefense and biosecurity concerns for these technologies have largely focused on containment of their resulting modifications or safeguarding their implementation as autonomous gene drives [3]. As DNA editing becomes more effective and more widely deployed, however, it will become necessary to consider the security of DNA sequences being edited by multiple organizations with conflicting goals. While there currently exist computational methods and tools to assess potential sites for DNA editing using different systems, few if any have been developed to evaluate these systems in a head-to-head contest.

To address this gap in biodefense and biosecurity capabilities, we have applied combinatorial game theory [2] to estimate the edit-based advantage that a DNA editing system or set of systems has relative to an opponent's. Unlike existing tools for site identification, our approach takes into account that some edits can prevent others from occurring, for instance by disrupting the PAM site required by a CRISPR-Cas based system. Here we investigate one-versus-one match-ups between 18 CRISPR base editors from a recent review [5] when they are applied to commonly edited genes in rice (*Oryza sativa*) [6] and bread wheat (*Triticum aestivum*) [4].

2 RESULTS

Figure 1 shows the results of analyzing two DNA editing games for all 153 one-versus-one match-ups between 18 CRISPR base editors. The first game is played on 28 rice genes (~ 130K bp), while the second is played on 15 bread wheat genes (~ 93K bp). For a complete list of these genes, see Table 1. The winners of these DNA editing games are the editors that can make the most uncontested edits (i.e., that have the largest edit advantage). Interestingly, the same five CRISPR base editors (SpCas9-NG(n), Sp-xCas9(n):ABE, ScCas9(n):ABE, ScCas9(n):CBE, and Sp-xCas9(n):CBE) dominate both of the studied games regardless of host organism, and one editor named SpCas9-NG(n) has a winning record (positive edit advantages as Editor/Player 1) against all other editors. In addition, it appears that the editors break down into five tiers based on their overall records (total edit advantage across all opponents). Furthermore, these tiers

are largely preserved across both games, with only some base editors changing their rank within a tier. For example, SpCas9-VRER(n):ABE changes from the worst editor in the lowest tier of the bread wheat game to the second best editor in the lowest tier of the rice game.

What factors, then, give some editors an advantage over others? Compared to factors such as edit window length and choice of deaminase (type of edit), we have found that the difference in PAM site variability between editors has the largest effect on a winner's edit advantage. Figure 2 shows the results of a linear regression modeling the relationship between (1) the difference in the number of PAM site sequence variants between editors and (2) the winning editor's edit advantage. Based on the regression's R^2 value, we estimate that ~ 78% of the variation in a winner's edit advantage can be explained by the variation in differences between editor PAM site variabilities alone. This makes sense since in general we would expect an editor with a more variable/less specific PAM site to be capable of making more edits and be less vulnerable to having its PAM site disrupted. While other factors such as edit window length and choice of deaminase appear to have negligible effects on a winner's edit advantage when considered individually, other more complex relationships between these factors and a winner's edit advantage may remain to be discovered. These relationships could potentially help to explain why a handful of editors were able to beat their opponents despite having less variable PAM sites (albeit by a smaller edit advantage than the cases in which the differences in PAM site variability between editors are large - see the cluster of points in Figure 2 having a negative PAM variant count delta).

3 DISCUSSION

Through the application of combinatorial game theory, we have established that CRISPR base pair editors with highly variable PAM sites have a significant quantitative advantage in terms of the number of uncontested edits that they can make relative to an opposing editor. We have also shown that this advantage persists across genes from two different organisms (rice and bread wheat), but it remains to be seen whether this is true for organisms with significantly different GC content. As monocot plants, both rice and bread wheat have higher GC content on average than animals and dicot plants. Since the CRISPR base editors with the greatest PAM site variability tend to require a single G-C base pair

at a specific position, it is possible that some of their edit advantage would be lost in organisms with low GC content.

Here we discuss five areas for future research:

Besides PAM site variability, what other factors are correlated with winning DNA editors? Can we apply these factors to design new top tier DNA editors that consistently have a larger edit advantage when compared to existing DNA editors? Can we incorporate these factors into a more general framework that includes other DNA editor characteristics such as editor speed and efficiency?

Can we redesign DNA sequences to be less vulnerable to DNA editing? In other words, can we edit these sequences such that no DNA editor(s) dominate their DNA editing games? Since we will necessarily invest in some DNA editors over others, and since we cannot necessarily predict which portion of a DNA sequence will be targeted for editing or whether our editor will have the advantage, it may be in our best interest to design our engineered DNA sequences such that their DNA editing games do not heavily favor some DNA editors over others.

Can we use combinatorial game theory to design DNA sequences that are less vulnerable to natural mutations? Since there exist tools for predicting the occurrence of DNA mutations, we can potentially treat nature as a player in a DNA editing game and attempt to design a DNA sequence to make its engineered function easier to rescue from disrupting mutations via subsequent DNA editing.

Can we use combinatorial game theory to design covert DNA watermarks that can be used to attribute engineered sequences to their lab of origin? In this work, we rounded the values of DNA editing games to the nearest integer (see Methods), but we could instead leverage their more complex canonical forms to identify games that are unique among games encoded by known DNA sequences. These DNA editing game watermarks could potentially be implemented via a small number of single-bp edits that would be difficult to detect, and the same watermark could be implemented via different edits in different cells, tissue types, etc.

Can we apply advanced combinatorial game theory to more specific DNA editing games? In this work, we considered DNA editing games in which so-called “loopy” game states are ignored (see Methods). If, however, we know precisely which DNA sequence and which portions of that sequence are going to be targeted, then we could relax some of the assumptions made in this work to perform deeper analysis of more specific scenarios (e.g., co-inheritance of gene drives that have been engineered to target a particular region of the genome and each other).

4 METHODS

Base combinatorial game theory assumes that a game is of finite length and does not contain a series of moves that

can be repeated indefinitely (a.k.a. not “loopy”). To enable application of this theory to DNA editing, we assume that the same base pair cannot be edited more than once, and we assume that an edited base pair cannot be part of a PAM site for a subsequent edit. These assumptions are not altogether unreasonable from a practical standpoint since, in the case of the first assumption, correcting the same base pair over and over does not constitute an edit advantage for either editor. The second assumption, on the other hand, is justified in part by two observations: (1) edits made in series are more likely to fail and (2) alternate editing windows can often be found close by the original target that enable a single edit to obtain a result similar to that obtained via two edits.

In order to compute the numeric values of DNA editing games, we use the computer algebra system known as Combinatorial Game Suite (CGSuite) [1]. In order to adapt DNA editing games to this system and compute the total edit advantage of a DNA editor with respect to its opponents, we have written a collection of scripts in Python and CGSuite’s custom-designed scripting language, CGScript. These scripts are responsible for partitioning target DNA sequences into 30-bp DNA editing sub-games, calculating the values of these sub-games, and summing them to obtain the edit advantage of the overall game. Note that we round the values of each sub-game to the nearest integer for efficient addition, since combinatorial games include infinitesimal values that can make the sum over many such games intractable. These infinitesimal values, however, typically represent games in which moves mutually block each other (i.e., the first move wins), so they do not practically impact our estimate of edit advantage (the number of uncontested edits).

Acknowledgements

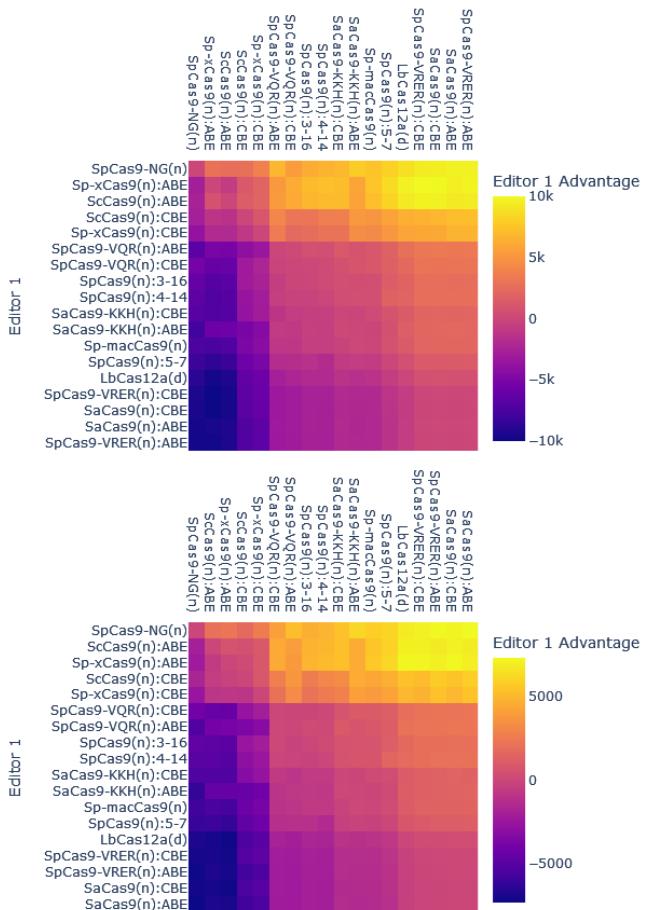
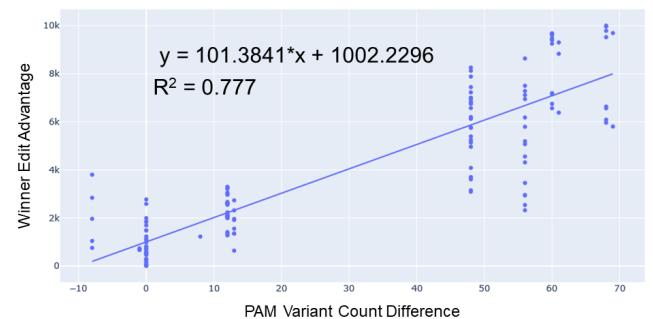
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

REFERENCES

- [1] 2022. <https://github.com/aaron-siegel/cgsuite>.
- [2] ALBERT, M. H., NOWAKOWSKI, R. J., AND WOLFE, D. *Lessons in Play: An Introduction to Combinatorial Game Theory*. Taylor & Francis Group, 2019.
- [3] ESVELT, K. M., SMIDLER, A. L., CATTERUCCIA, F., AND CHURCH, G. M. Concerning rna-guided gene drives for the alteration of wild populations. *eLife* 3 (2014), e03401.
- [4] LI, S., ZHANG, C., LI, J., YAN, L., WANG, N., AND XIA, L. Present and future prospects for wheat improvement through genome editing and advanced technologies. *Plant Commun.* 2 (2021).
- [5] MOLLA, K. A., AND YANG, Y. Crispr/cas-mediated base editing: technical considerations and practical applications. *Trends Biotechnol.* 37 (2019), 1121–1142.
- [6] ZAFAR, K., ET AL. Genome editing technologies for rice improvement: progress, prospects, and safety concerns. *Front. Genome Ed.* 2 (2020).

Table 1: Genes for DNA Editing Games in Rice and Wheat

Organism	Genes	Total bp	% GC
<i>O. sativa</i>	ROC5, SPP, YSA, OsPDS, OsBEIIb, OscDC48, OsALS, OsSPL14, Gn1a, GS3, DEP1, PYL1, PYL6, OsFAD2-1, Badh2, SBEI, OsNramp5, SAPK2, Os-SWEET11, OsSWEET13, OsSWEET14, OsERF922, BBM1, REC8, PAIR, OSD1, SF3B1, OsEPFL9	129,589	0.44
<i>T. aestivum</i>	TaMLO, TaLOX2, TaGASR7, TaEDR1, TaGW2, TaZIP4, TaHRC, TaMs1, TaSBEIIa, TaDA1, TaPDS, TaNCED1, TaCENH3alpha, TaACC, Ubiquitin	92,457	0.48

**Figure 1: Heatmaps of Editor/Player 1 advantage in all 153 one-versus-one match-ups between 18 CRISPR base editors for DNA editing games with rice genes (top) and bread wheat genes (bottom).****Figure 2: Winner edit advantage versus difference in editor PAM variant counts for DNA editing games with rice genes.**

Biological Malware Detector

Muntaha Samad

Raytheon BBN

muntaha.samad@rtx.com

Dan Wyschogrod

Raytheon BBN

dan.wyschogrod@rtx.com

Jacob Beal

Raytheon BBN

jakebeal@ieee.org

Motivation

Companies around the world have been synthesizing nucleic acids for the past several decades. With advances in recent years, vendors can produce long, low-cost, high-fidelity custom nucleic acid sequences. While this capability has helped advance biotechnology research in many positive ways, it has also created the potential for bad actors to attempt to synthesize hazardous sequences. To combat this issue, in 2009 a group of nucleic acid synthesis vendors founded the International Gene Synthesis Consortium (IGSC) to create guidelines for screening nucleic acid synthesis orders [1]. A similar set of voluntary guidelines was introduced by the US HHS in 2010 [2]. The IGSC and HHS guidelines prescribe a common protocol for screening DNA sequences and customers, while promoting the beneficial use of nucleic acid synthesis, and have been adopted by most nucleic acid synthesis providers to screen orders for the presence of sequences of concern (SoCs).

While the development of screening guidelines was an important step towards increased biosecurity, exclusively relying on nucleic acid synthesis vendors introduces several problems. First, the primary mission of nucleic acid synthesis companies is not biosecurity, even though they are serving as a primary line of defense against bioterrorism and carelessness. This can leave these organizations under-resourced and vulnerable to methods for SoC concealment such as assembly from oligos or protein editing. Additionally, sequence screening happens only after customers have placed an order for biological materials, thereby leaving threat detection to latest possible point of intervention before a physical SoC is acquired. Finally, many nucleic acid vendors are using local alignment tools, such as BLAST against public reference databases, to identify potentially dangerous orders which can be slow (minutes per kilobase), computationally expensive (terabyte-scale databases, cluster-scale computation) [3], and prone to errors [4].

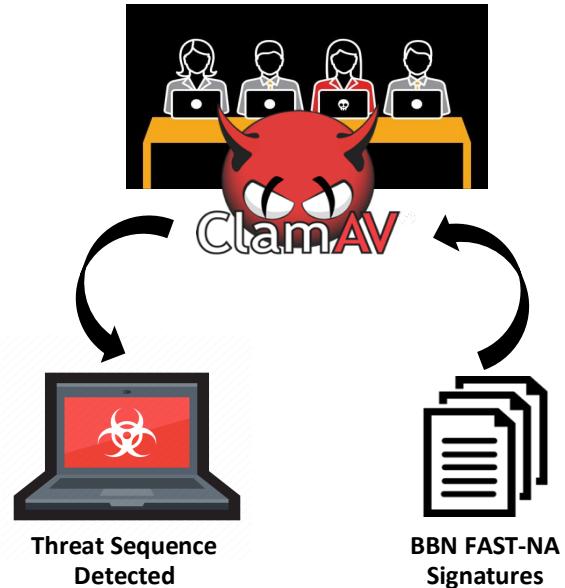


Figure 1: The Biological Malware Detector loads biological signatures for sequences of concern into standard signature-based malware detection software to search the digital activities of individuals or organizations for biological threats.

Proposed Improvement: Biological Malware Detector

To address these shortcomings, we propose the Biological Malware Detector (BMD) a software tool that scans electronic systems for the sequences of biological threats in the same way anti-virus software currently scan systems for cyber-threats. Following the architecture shown in Figure 1, BMD combines biological threat signatures from BBN's FAST-NA Scanner [5] with ClamAV, a free and open-source host intrusion detection system to directly scan the digital activities of individuals or organizations for biological threats. BMD runs on a host device in a similar manner to anti-virus software except that instead of scanning for computer viruses, BMD scans for SoC's.

BMD has several significant use cases, each with substantial implications for enhancing biosecurity. First, BMD can serve as a precautionary measure to fortify facility biosecurity. For instance, the biosafety personnel

of a laboratory could employ BMD on company laptops to monitor and detect insider threats or bio-error, by ensuring that all digital activities involving biological agents are in line with expectations. This use case would take some of the burden off nucleic acid synthesis vendors by catching bad actors before they place orders and before they have a chance to take steps towards concealment. Another use case made possible by BMD is targeted actor forensics, which would allow biosafety and intelligence organizations to make preliminary assessments of the biological threat capabilities and intentions of persons and organizations. For instance, a security organization could use BMD to investigate potential malicious actors by obtaining their computers and directly examining their digital activities for SoCs. This sort of targeted forensics was not previously possible with BLAST-based methods because they are slow and computationally intensive, making a scan through of all the digital resources of an individual or organization infeasible.

Methods

The SoC signatures that we have tested in BMD are Raytheon BBN proprietary FAST-NA Scanner signatures, though signatures developed by other means could potentially be used as well. The FAST-NA Scanner library of signatures is comprehensive, including all potentially dangerous sequences listed by the IGSC, the United States Commerce Control List, as well as the Australia Group Control List. The FAST-NA Scanner signatures have undergone stringent tuning to exclusively identify SoCs. To ensure their accuracy, these signatures are validated against curated datasets from the National Center for Biotechnology Information (NCBI), encompassing all taxa covered by the signatures, as well as closely related non-threatening sibling taxa. For instance, with this tuning process FAST-NA signatures can be used to correctly identify *Bacillus anthracis* as a significant threat while accurately classifying its close relative *Bacillus cereus* as a non-threat.

The anti-virus software utilized in BMD is ClamAV, a free and open-source signature-based detection system designed for detecting viruses and malware. ClamAV allows users to upload unique signatures and schedule scans to identify instances of the signatures within the digital resources stored on a device.

BMD combines the biological threat signatures from BBN's FAST-NA Scanner and ClamAV's signature-based detection to directly search the digital activities of individuals or organizations for biological threats.

BMD Deployment

BMD is designed to exclusively flag biological threats by leveraging finely tuned signatures. In a scan of an entire laptop (~20 GB) seeded with several files known to contain threat material, there were 0 false positives, 0 false negatives, and 10 true positives. Since nothing is reported other than signature matches, his low false positive rate can assure users that their privacy, particularly with respect to access to digital resources by lab administrators, will not be compromised by BMD flagging harmless files as potential threats. Additionally, BMD's computational load is low, only utilizing ~80% of a single CPU during scan. With this low computational load and fast scan time (Figure 2), use of BMD should not impede normal workflow for individuals or organizations.



Figure 2: Plot comparing the amount of data scanned vs the time it takes BMD to scan it.

Conclusion

The development of BMD is the beginning of a new era in digital biodefense, empowering biosafety personnel in laboratories to proactively address insider threats and enabling intelligence organizations to directly analyze digital activities for SoCs. By implementing BMD, early detection of intentions related to hazardous agents can become a reality, significantly mitigating the risks of biological accidents and bio-terrorism.

REFERENCES

1. "Home: International Gene Synthesis Consortium." *International Gene Synthesis Consortium / The Promotion of Biosecurity*, 11 July 2023, genesynthesisconsortium.org/.
2. HHS (Department of Health and Human Services). (2010) Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA. Federal Register 75, 62820. October 13.
3. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410.
4. Beal J, Clore A, Manthey J. Studying pathogens degrades BLAST-based pathogen identification. Sci Rep. 2023 Apr 3;13(1):5390. doi: 10.1038/s41598-023-32481-z. PMID: 37012314; PMCID: PMC10068195.
5. Dan Wyschogrod, et al., Adapting Malware Detection to DNA Screening, IWBD, October, 2022.

Long-term Evolution of Bacteria for Maximal Growth Rate

Antoine Vigouroux

Harvard medical school
Boston, United States
antoine_vigouroux@hms.harvard.edu

Sadık Yıldız

UT Southwestern Medical Center
Dallas, United States
msadikyildiz@gmail.com

Johan Paulsson

Harvard medical school
Boston, United States
johan_paulsson@hms.harvard.edu

1 INTRODUCTION

Long-term evolution in the laboratory enables direct observation of adaptation over time. Several experiments in the past have provided insights into the process of evolution, such as second-order selection [2, 7], the emergence of parallel subpopulations [1, 5], or universal constraints on protein evolution [3].

However, most long-term experiments so far relied on daily manual dilutions, creating a complex environment that alternates between growth and starvation. This complexity can make the results harder to interpret and limits the number of mutation/selection cycles that can be performed in a day.

Here, we introduce two new long-term evolution experiments, where *Escherichia coli* and *Vibrio natriegens* are maintained indefinitely in a state of constant exponential growth. By running these experiments for several decades, we hope to reveal the fundamental limits of growth, providing a new window into the basic requirements for life.

2 PERPETUAL EXPONENTIAL GROWTH

To create an environment where the only way to gain fitness is by replicating faster, we provide cells with a constant excess of nutrients, so they grow exponentially at the fastest possible rate. In these conditions, a mutant that grows faster than the rest of the population will increase in frequency and take over the population.

To achieve this, we designed evolution machines based on the turbidostat principle, where medium turbidity is kept at a constant level. However, in the classic turbidostat design, cells can escape dilution by sticking to the surface of the reactors. This creates a second selection pressure in favour of biofilm formation [4, 6], making it impossible to run a simple turbidostat over the timescales required to maximize growth rate.

To solve this problem, our automated machines not only dilute cultures with fresh medium but also transfer cultures between reactors. They can then autonomously sterilize and rinse the used reactors, making them available again for cell culture.

For long-term evolution, the growing populations are transferred back and forth between two reactors: first, the

population grows in one turbidostat, while the second reactor is automatically sterilized (Fig. 1A). The culture is then transferred to the second reactor. Any cell that was sticking to the reactor's walls remains in the first reactor (Fig. 1B) and is eliminated. The population then continues to grow in the second reactor, while the first reactor is sterilized (Fig. 1C). Finally, the culture is transferred back to the first reactor (Fig. 1D), and the cycle is repeated every 2.5 hours. As all these steps are fully automated, this process can continue day and night with minimal user intervention.

To control the machines, we developed a cross-platform, python-based interface designed for live diagnosis and repair without interrupting the machine's operation. All components in the hardware and software were meticulously optimized to maximize reliability, allowing the machines to be left unattended for weeks to months.

3 TWO LONG-TERM EVOLUTION EXPERIMENTS

Using these machines, we launched two long-term evolution experiments with two fast-growing bacteria: *Escherichia coli* and *Vibrio natriegens*. For each of these, we grow eight lineages in parallel.

The bacteria grow in rich medium at optimal temperatures to maximize their initial growth rate and increase the speed of mutation/selection cycles. For instance, *E. coli* divides approximately every 19 minutes, undergoing about 75 generations daily (Fig. 1E). *Vibrio natriegens*, the fastest organism ever observed, has a doubling time of 12 minutes, corresponding to 120 generations per day. We intend to run these experiments for several decades, targeting one million generations within 25-30 years.

4 45,000 GENERATIONS OF *E. COLI* EVOLUTION

At the time of writing, the *E. coli* experiment has spanned 47,000 generations (≈ 1.5 years). Competitions assays show that all populations increased in growth rate (Fig. 2). However, the increase in growth rate was remarkably slow, with a doubling time reduced by at most a minute after 30,000 generations.

We sequenced the genomes of the eight populations once every 2,500 generations (Fig. 3). Many new genetic variants

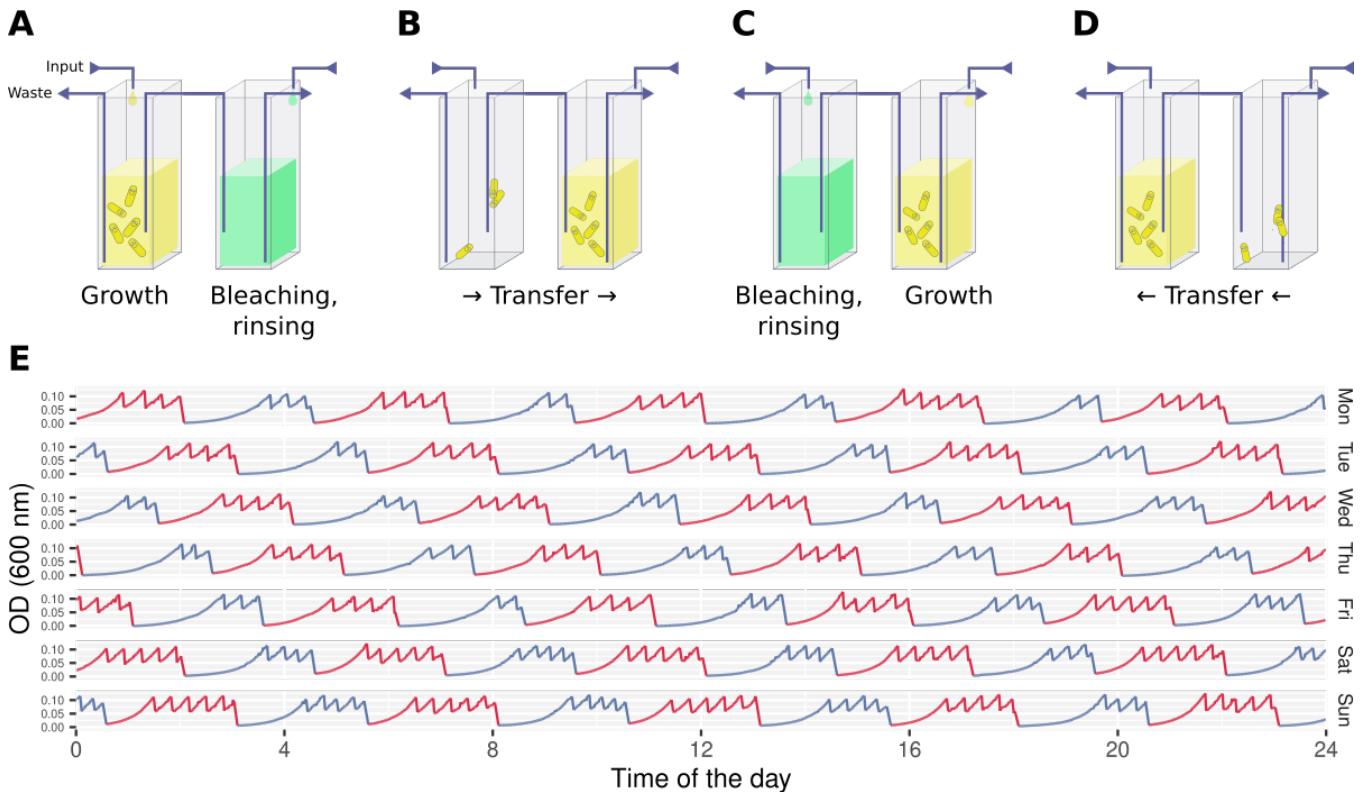


Figure 1: ABCD: Overview of a growth cycle in the self-sterilizing pairs of turbidostats. Our current machines can hold up to eight parallel populations at a time. **E:** Optical density readings for one culture during one week of exponential growth. The two colours correspond to the two alternating reactors.

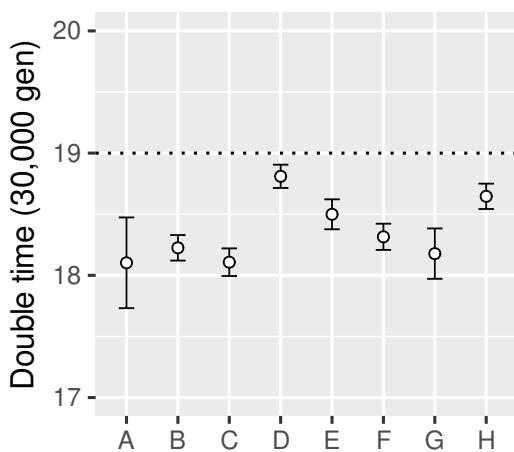


Figure 2: Preliminary estimates of the doubling times after 30,000 generations, calculated from a competition against the ancestor in the machine. The letters A to H designate the eight parallel populations. The 19 min horizontal line is the original doubling time of the ancestor.

emerged as cells evolved for higher growth rate, both in gene coding sequences and regulatory regions.

In accordance with the slow increase in growth rate, the fixation of new variants was initially very rare, and the variants took several thousands of generations to take over. This indicates that the mutations that have the potential to increase *E. coli*'s growth rate have, in general, a very small effect.

One notable feature is that several populations altered their mutation rate, usually through mutations in the mismatch repair system (*mutS*(G696E), *mutS*(M260T), indels in *mutL* and *mug*, indels in *mutL*'s promoter). Current work is ongoing to understand how these variants initially emerge and how they can reach fixation.

5 CONCLUSION

Our evolution machines provide a powerful platform for laboratory evolution in a constant, defined environment. While our first devices will be dedicated to these multi-decade evolution experiments, they are designed to be easily tweaked

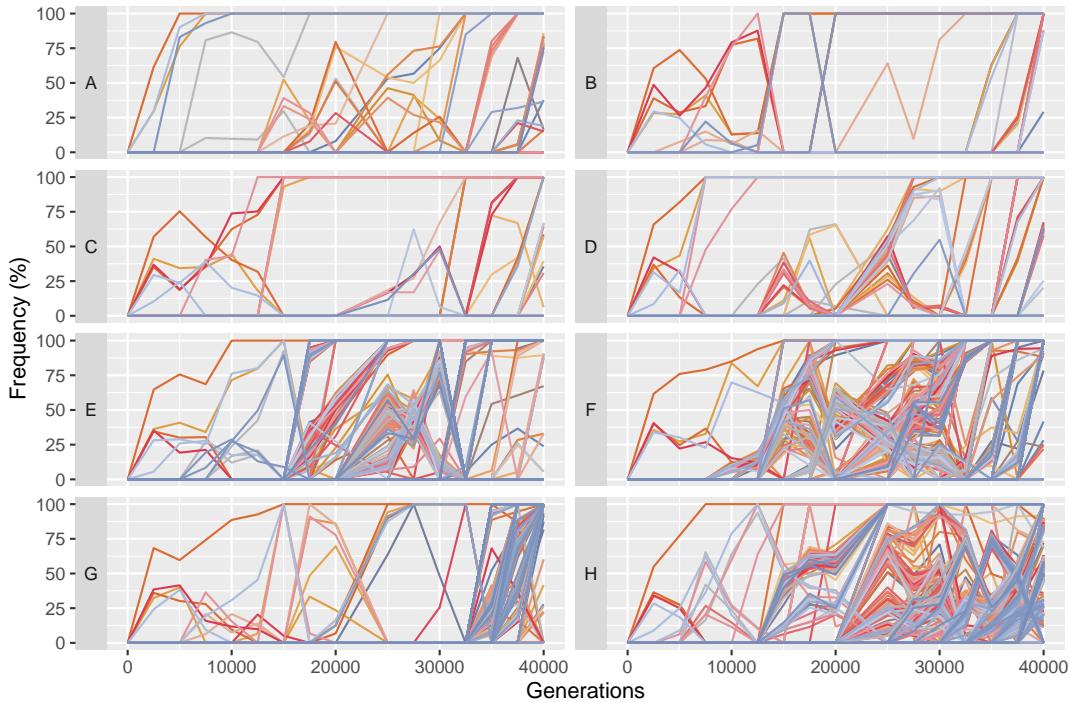


Figure 3: Evolution of the genomes of the 8 parallel populations (labeled as A to H) over 40,000 generations. Each line represents one variant (with arbitrary colors), with the y-axis showing the percentage of the population that carries the new variant. The mutation rate in lineages E, F, G and H increased ≈ 10 -fold at some point of the experiment.

and repurposed for diverse applications. The modular software makes it easy to adapt the interface to other modes of operation or different machine setups.

By making it available as an open-hardware platform in the future, we hope to accelerate progress in laboratory evolution, for both fundamental research and applied directed evolution.

REFERENCES

- [1] BEHRINGER, M. G., CHOI, B. I., MILLER, S. F., DOAK, T. G., KARTY, J. A., GUO, W., AND LYNCH, M. Escherichia coli cultures maintain stable subpopulation structure during long-term evolution. *Proceedings of the National Academy of Sciences* 115, 20 (May 2018), E4642–E4650. Publisher: Proceedings of the National Academy of Sciences.
- [2] JOHNSON, M. S., GOPALAKRISHNAN, S., GOYAL, J., DILLINGHAM, M. E., BAKERLEE, C. W., HUMPHREY, P. T., JAGDISH, T., JERISON, E. R., KOSHEL-EVA, K., LAWRENCE, K. R., MIN, J., MOULANA, A., PHILLIPS, A. M., PIPER, J. C., PURKANTI, R., REGO-COSTA, A., McDONALD, M. J., NGUYEN BA, A. N., AND DESAI, M. M. Phenotypic and molecular evolution across 10,000 generations in laboratory budding yeast populations. *eLife* 10 (Jan. 2021), e63910. Publisher: eLife Sciences Publications, Ltd.
- [3] MADDAMSETTI, R. Universal Constraints on Protein Evolution in the Long-Term Evolution Experiment with Escherichia coli. *Genome Biology and Evolution* 13, 6 (June 2021), evab070.
- [4] MILLER, C., KONG, J., TRAN, T. T., ARIAS, C. A., SAXER, G., AND SHAMOO, Y. Adaptation of Enterococcus faecalis to Daptomycin Reveals an Ordered Progression to Resistance. *Antimicrobial Agents and Chemotherapy* 57, 11 (Oct. 2013), 5373–5383. Publisher: American Society for Microbiology.
- [5] ROZEN, D. E., AND LENSKI, R. E. Long-Term Experimental Evolution in Escherichia coli. VIII. Dynamics of a Balanced Polymorphism. *The American Naturalist* 155, 1 (Jan. 2000), 24–35. Publisher: The University of Chicago Press.
- [6] TAKAHASHI, C. N. *A Platform for Microbial Evolution and Characterization*. Thesis, Dec. 2015. Accepted: 2016-03-11T22:39:27Z.
- [7] WOODS, R. J., BARRICK, J. E., COOPER, T. F., SHRESTHA, U., KAUTH, M. R., AND LENSKI, R. E. Second-Order Selection for Evolvability in a Large Escherichia coli Population. *Science* 331, 6023 (Mar. 2011), 1433–1436. Publisher: American Association for the Advancement of Science.

Splicing-based Biocontainment Devices

Allison Taggart¹, Miles Rogers¹, Jacob Beal¹

¹Raytheon BBN Technologies, Cambridge, MA

allison.j.taggart@rtx.com,miles.rogers@rtx.com,jake.beal@rtx.com

1 INTRODUCTION

Endogenous RNA splicing is catalyzed by a large ribonucleoprotein complex, known as the spliceosome, and contains snRNAs that pair directly to the pre-mRNA gene (Fig 1A, left). This pairing is critical for splice site recognition and accurate splice site pairing. Mutations to the splicing sequences involved in these interactions can disrupt recognition and result in incorrect mRNA processing. We propose to take advantage of these recognition events in RNA splicing to engineer biocontainment by making splicing-based containment devices that rely on a combination of native splicing machinery and artificial supplied splicing components.

Orthogonal splicing systems have been demonstrated in the lab in which pre-mRNAs with mutated splicing sequences can be rescued with the application of artificial snRNAs containing the compensatory mutations [1] [2]. Our design inserts introns with mutations in one or more splice site regions and matching mutations in engineered snRNAs to restore complementarity (Fig 1A, right).

There are two strategies in which these splicing devices can be used for system containment. In the first strategy, the device can be inserted into one or more essential genes (Fig 1B). Here, when artificial splicing components are not supplied to the organism, the essential genes will not be correctly processed and the organism is no longer viable. In the second strategy, the device can be inserted into an engineered gene or gene of interest, such that in non-desired conditions the organism will persist but the gene containing the device will be nonfunctional (Fig 1C). There exist different implications and evolutionary pressures for escape mechanisms for each strategy, and the most appropriate strategy may depend on application.

2 MODELING SYSTEM EFFICACY REQUIREMENTS

The desired state, in which the engineered device is spliced and processed correctly in the presence of both native and synthetic spliceosomal components, can be computed as:

$$d_{OS} = \alpha(f_{NE} + f_{EE})^k$$

where α = transcription rate, f_{NE} = fraction of engineered introns spliced by native spliceosome, f_{EE} = fraction of engineered introns spliced by synthetic spliceosome, and k = number of synthetic introns. If d_{OS} is too low, the engineered device will not be processed and the system will die.

The undesired state, in which biocontainment fails and splicing occurs in the absence of synthetic snRNAs, is:

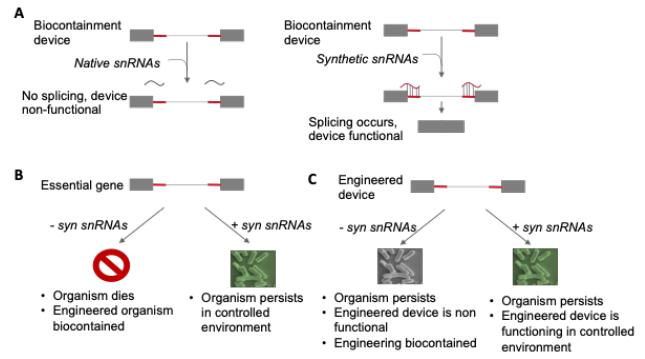


Figure 1: (A) Mutant introns block endogenous splicing (left), but splice with artificial snRNA (right). Two biocontainment strategies: (B) insert a artificial intron in an essential gene, or (C) into the engineered device.

$$n_{OS} = \alpha(f_{NE})^k$$

The efficacy e of the system is thus proportional to the ratio of desired splicing to undesired splicing:

$$e \propto \frac{(f_{NE} + f_{EE})^k}{(f_{NE})^k}$$

Modeling system efficacy shows a relationship between the required efficiencies of recognition of both native and synthetic spliceosomal components to the overall efficiency of the system (Fig 2). The highest efficacy system would be a system in which f_{EE} (splicing in the presence of supplied synthetic RNAs) is 1, and f_{NE} (splicing in the absence of supplied synthetic RNAs) is 0. Overall, our results demonstrate that there is an inflection point, such that as long as we are able to maintain a low f_{NE} , f_{EE} only needs to achieve about 0.75 efficiency to achieve maximum system efficacy. Additionally, our modeling results suggest that adding multiple introns to the system can improve system efficiency. There exist multiple design strategies for increasing the number of synthetic introns to the system.

3 DESIGN VERSUS EVOLUTIONARY ESCAPE

Splicing-based biocontainment can be implemented using a number of design strategies, outlined in Fig 3., all of which use an unspliced biocontainment intron to interrupt gene function. Inserting a biocontainment intron with a length not a multiple of three will shift codons out of frame for

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.

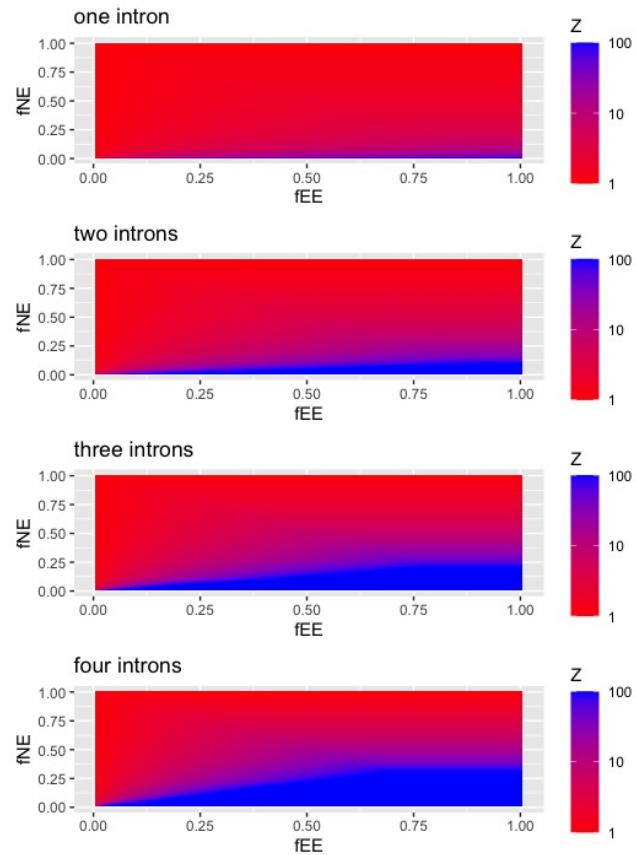


Figure 2: Modeling estimate for biocontainment efficiency from various numbers of introns and range of splicing parameters, coloring efficacy on a log scale.

the remainder of the transcript, likely destroying function (design 1). Similarly, inserting a biocontainment intron containing a stop codon can produce a non-functional truncated protein (design 2). Another design category is to insert a biocontainment intron in a known activity domain or relevant protein structure that is necessary for protein function. When unspliced, the biocontainment intron will interrupt these domains and could interfere with function (design 3).

There are additional strategies in which a biocontainment intron can be used to introduce targeting sites, such as cleavage, RNA binding protein, or siRNA targeting sites, which can all be used to block gene processing (designs 4,5). It is also possible to design a strain with an essential small RNA contained within the biocontainment, in which processing of the small RNA requires splicing and correct intron lariat formation (design 6). Lastly, biocontainment introns can be designed to contain destabilizing sequences or structures, such as hairpins, that when unspliced interfere with function (Design 7).

Several classes of evolutionary escape and biocontainment failure must be considered, including point mutations,

1	Unspliced intron puts mRNA out of frame
2	Unspliced intron introduces premature stop codon
3	Unspliced intron interrupts an activity domain or causes a misfolding change
4	Unspliced intron contains a cleavage site that can be targeted for mRNA turnover
5	Unspliced intron contains a site that recruits RBPs or siRNAs that target transcript and block processing
6	Unspliced intron contains required small RNA
7	Unspliced intron contains destabilizing mRNA or protein sequence or nonfunctional structure

Figure 3: Potential splicing-based biocontainment designs

recombination and deletion, horizontal gene transfer, and alternate pathways for essential genes. Point mutations may restore splice site sequences, affecting all designs, so it may be beneficial to have multiple mutations in the splice site sequences or spread multiple biocontainment introns across one or multiple genes. Point mutations may also restore the reading frame (Design 1) or remove a premature stop codon (Design 2), which could similarly be mitigated by insertion of multiple introns. Interruption of an activity domain (Design 3) is likely to be more resistant to point mutations, and may prove to be an attractive design choice. Recombination and deletion may remove biocontainment introns, but is predicted to be a lesser threat due to the precision required at the intron boundaries. Horizontal gene transfer may be a bigger concern, but is dependent on application, chassis, and mating patterns. Similarly the degree of concern about alternate pathways depends on organism characterization. Both horizontal gene transfer and alternate pathways can be mitigated by spreading biocontainment introns across multiple genes. Leaky gene expression is likely to be a challenge for Designs 4 and 5, which require functional external systems to be recruited to the site for containment. Finally, Design 6 may be challenging to implement, due to ambiguities with intron lariat stability and processing, but these may be overcome with well-characterized introns.

In sum, the use of an orthogonal splicing system appears to be a viable approach to biocontainment, with a number of potential designs, some of which are more resilient to escape than others. Ongoing work in our laboratory aims to demonstrate these approaches in practice.

REFERENCES

- [1] SMITH, D. J., KONARSKA, M. M., AND QUERY, C. C. Insights into branch nucleophile positioning and activation from an orthogonal pre-mRNA splicing system in yeast. *Molecular Cell* 34, 3 (May 2009), 333–343.
- [2] ZHUANG, Y., AND WEINER, A. M. A compensatory base change in human U2 snRNA can suppress a branch site mutation. *Genes & Development* 3, 10 (Oct. 1989), 1545–1552.

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.

Automated model curation using LLMs: Integration of ChatGPT with the DySE framework

Emilee Holtzapple
University of Pittsburgh
Pittsburgh, USA
erh87@pitt.edu

Tanvi Verma
University of Pittsburgh
Pittsburgh, USA
tav38@pitt.edu

Natasja Miskov-Zivanov
University of Pittsburgh
Pittsburgh, USA
nmzivanov@pitt.edu

1 1 Introduction

The advent of large language models (LLMs) has led to an increase in the potential for automation of research tasks. LLMs such as ChatGPT are trained on large knowledgebases and are capable of answering prompts about a wide variety of scientific subjects. Nevertheless, LLMs are not without limitations. They may occasionally produce inaccuracies and prove sensitive towards bias ingrained in their training data. By integrating state-of-the-art AI models with existing curation tools, we can curate models accurately and quickly. Here we showcase several uses of ChatGPT that are compatible with the Dynamic System Explanation (DySE) framework for automated model curation.

15

16 2 Background

A recent advance in the field of artificial intelligence is the development of pretrained language models such as ChatGPT [1]. During the training phase, these AI models employ contextual understanding of previous words to anticipate the next word in a sentence. This allows the model to comprehend syntax, semantics, and even engage in certain forms of reasoning. These models exhibit a broad array of capabilities that include language translation, text summarization, question-answering, chatbot interactions, and more. Their ability to generate human-like text makes them a valuable tool for a diverse array of applications – such as curating models of cell signaling and disease. ChatGPT is capable of understanding biomedical journal articles, entity recognition and event extraction, and was trained on several biomedical ontologies and databases. However, ChatGPT lacks novel information published after 2021, which limits its applicability for modeling.

Semi-automated modeling methodology is capable of incorporating up-to-date information for quick and

37 accurate model curation. The Dynamic System
38 Explanation (DySE) framework [2] is a collection of
39 techniques and tools used to extract information, curate,
40 and analyze models of cell signaling mechanisms. DySE
41 utilizes machine reading to gather interactions from
42 biomedical text and curate executable models at
43 different levels of abstraction. The framework also
44 includes methods for analyzing these models in an
45 automated manner to predict system behavior or
46 provide guidance for interventions. One such method is
47 incorporated within the Discrete Stochastic
48 Heterogeneous Simulator (DiSH), which can reproduce
49 the dynamic behavior of cell signaling [3]. DiSH takes an
50 executable model written in the [BioRECIPE](#) (Biological
51 system Representation for Evaluation, Curation,
52 Interoperability, Preserving, and Execution) format,
53 along with simulation parameters, and generates
54 trajectories that capture changes in state over time for all
55 elements within the model. This simulation can be
56 configured to mimic any scenarios observed *in vitro* or *in vivo*.

58 3 Use cases

59 **3.1 Event extraction.** We used the following prompt to
60 extract events from a literature corpus:

61 “Convert the following text: “*evidence statement*” to a
62 knowledge graph.”

63 where “*evidence statement*”, one or more sentences
64 from a paper, is provided as plain text. An example
65 prompt is shown in Figure 1(A), and the source-target
66 relation pairs are listed in Figure 1(B). We chose a
67 representative evidence statement that demonstrates
68 the utility of ChatGPT for event extraction.

69 **3.2 ID mapping.** ChatGPT was pretrained on gene and
70 protein databases and is capable of providing
71 standardized identifiers for multiple entity types. To

Source Node	Target Node	Relation
Cancer cells	Warburg effect	Uses
Warburg effect	Aerobic glycolysis	Leads to
Cancer cells	Energy demands	Meets
Aerobic glycolysis	Oxidative phosphorylation	Switches to
Oxidative phosphorylation	Aggressive phenotype	Results in
Aerobic glycolysis	Hypoxia-inducible factors (HIF)-1 α	Downregulates

Figure 1. (A) Sample prompt, (B) events extracted by ChatGPT.

72 ground names of entities in the knowledge graph
 73 generated in the previous section, the following prompt
 74 was used:

75 "Can you match pathways and processes in the
 76 knowledge graph to Gene Ontology terms, and match
 77 proteins to UniProt IDs?"

78 The results are detailed in Table 1. A human curator
 79 assessed the accuracy of each ID provided by ChatGPT.
 80 Careful construction of the prompt ensures that ChatGPT
 81 uses the correct database for each entity type. While
 82 there does exist a database for standardizing cell lines,
 83 ChatGPT was not able to match "Cancer cells" to any ID
 84 in the COSMIC database.

85 **3.3 Comparison to machine readers.** To compare the
 86 usefulness of LLMs to pre-existing methods for extracting
 87 events from text, we selected ten events extracted from
 88 text using the REACH engine [4]. Each event was judged
 89 based on its correctness using the following criteria:

- 90 1. "Regulator Name" and "Regulated Name" match
 91 the upstream and downstream elements in the
 92 evidence statement
- 93 2. The effect of the regulation ("positive" or
 94 "negative") matched the evidence statement
- 95 3. "Mechanism" matches the evidence statement

96 For the mechanism of action, occasionally the reader
 97 ignored mechanistic details present in the evidence
 98 statement. The reader also occasionally inferred
 99 additional mechanistic details not in the text. To judge
 100 whether the mechanism was correct, the guide in Table
 101 2 was used. To compare to the accuracy of using ChatGPT
 102 for event extraction, each evidence statement was used
 103 in the following prompt:

104 "From the given statement: "*evidence statement*", can
 105 you identify the molecular interaction and list the names

106 of the regulated molecule and its regulator, as well as the
 107 mechanism of interaction?"

108 The extracted event was then compared to the results
 109 from REACH (Table 3). ChatGPT was able to correctly
 110 infer slightly more events from text than the machine
 111 reader, but it did not achieve 100% accuracy.

112 4 Discussion

113 While large language models offer opportunities for
 114 leveraging vast amounts of text data to gain insights into
 115 cell signaling, they also have limitations related to
 116 domain knowledge, accuracy, and context
 117 understanding. Integrating these models with specialized
 118 biological databases and subject matter experts can lead
 119 to more robust and accurate applications. Human
 120 curators should be involved to verify the generated
 121 information through reliable sources before making
 122 scientific conclusions or decisions based solely on the
 123 outputs of language models. The results demonstrate the
 124 value of ChatGPT as input to DySE.

125 REFERENCES

- 126 [1] K. I. Roumeliotis and N. D. Tselikas, "ChatGPT and Open-AI Models:
 127 A Preliminary Review," *Future Internet*, vol. 15, no. 6, doi:
 128 10.3390/fi15060192.
- 129 [2] C. A. Telmer *et al.*, "Dynamic system explanation: DySE, a
 130 framework that evolves to reason about complex systems - lessons
 131 learned," presented at the Proceedings of the Conference on Artificial
 132 Intelligence for Data Discovery and Reuse, Pittsburgh, Pennsylvania,
 133 2019. Available: <https://doi.org/10.1145/3359115.3359123>.
- 134 [3] K. Sayed, Y. Kuo, A. Kulkarni, and N. Miskov-Zivanov, "DiSH
 135 simulator: Capturing dynamics of cellular signaling with
 136 heterogeneous knowledge," in *2017 Winter Simulation Conference*
 137 (WSC), 3-6 Dec. 2017, pp. 896-907, doi: 10.1109/WSC.2017.8247841.
- 138 [4] M. A. Valenzuela-Escarcega, G. Hahn-Powell, M. Surdeanu, and T.
 139 Hicks, "A Domain-independent Rule-based Framework for Event
 140 Extraction," in *ACL*, 2015.

Table 1. Entity IDs generated by ChatGPT.

Entity	Type	Gene Ontology ID	UniProt ID
Cancer cells	Cell Type		
Warburg effect	Metabolic Process	GO:0006096	
Aerobic glycolysis	Metabolic Pathway	GO:0006097	
Energy demands	Biological Process	GO:0006118	
Oxidative phosphorylation	Metabolic Pathway	GO:0006119	
Hypoxia-inducible factors (HIF)-1α	Protein		P12345
Aggressive phenotype	Cell Behavior		

Table 2. Guide for manual judgement of automatically extracted events.

Tool, reader, or LLM			
Evidence statement	Explicit mechanistic detail	Correct	Partially correct*
	Insufficient mechanistic detail	Incorrect	Correct

Table 3. Comparison of events extracted using machine reading to those extracted using ChatGPT. The downstream element identified using the reader is given in “Element Name”, and its standardized ID is listed in “Element ID” (if available). The name of the regulator, either positive or negative, is given in “Pos Reg Name” or “Neg Reg Name”. The “Connection Type” column describes whether the regulation is a physical or functional interaction. For direct regulations, the “Mechanism” column has additional detail about the regulation. The “Paper ID” identifies the paper where “Evidence Statement” is located in. “Reader Correctness” and “LLM Correctness” detail whether REACH and ChatGPT results agree with the “Evidence Statement”, as judged by a human curator.

Element Name	Element ID	Positive Reg Name	Negative Reg Name	Connection Type	Mechanism	Paper ID	Evidence Statement	Reader Correctness	LLM Correctness
AKT	AKT	PI3K		Indirect	NONE	PMC149420	Subsequently, PI3K activates... ¹	Correct	Incorrect
Akt	AKT	mTORC2		Direct	Phosphorylation	PMC2075366	mTORC2 phosphorylates and... ²	Correct	Correct
PRAS40	Q96B36	mTORC2		Direct	Phosphorylation	PMC2075366	PRAS40 was phosphorylated... ³	Incorrect	Correct
TSC2	P49815		Akt	Indirect	NONE	PMC2075366	Akt (also known as PKB)... ⁴	Partially correct	Incorrect
Caspase	Caspase		Akt	Indirect	NONE	PMC2185587	Akt, but Not Bcl-2, Inhibits... ⁵	Correct	Correct
p110gamma	P48736	Ras		Indirect	NONE	PMC2652403	Ras activates p110gamma... ⁶	Correct	Correct
Rho	P08100	Ras		Indirect	NONE	PMC2652403	It was demonstrated that Ras... ⁷	Correct	Correct
Akt	AKT	PIK3CA		Indirect	NONE	PMC2652403	All the mutants strongly... ⁸	Correct	Correct
tuberin	P49815		Akt	Indirect	NONE	PMC2652403	Akt also inhibited tuberin... ⁹	Incorrect	Correct
ERK	ERK	PI3K		Indirect	NONE	PMC2683723	PI3K inhibition suppresses... ¹⁰	Incorrect	Partially correct

¹Subsequently, PI3K activates AKT and PKB that interferes with the apoptotic machinery.

²mTORC2 phosphorylates and activates Akt which then phosphorylates and inactivates the pro apoptotic factors BAD and FOXO1/3a.

³PRAS40 was phosphorylated weakly by both mTORC1 and mTORC2 .

⁴Akt (also known as PKB) phosphorylates and inactivates TSC2 in response to growth factors, whereas AMPK phosphorylates and activates TSC2 in response to low energy (high AMP).

⁵Akt, but Not Bcl-2, Inhibits Caspase Activation Induced by Cytochrome c in a Cell-free System.

⁶Ras activates p110gamma at the level of the membrane , by allosteric modulation and/or reorientation of the p110gamma, implying that Ras can activate p110gamma without its membrane translocation. Ras activates p110gamma at the level of the membrane, by allosteric modulation and/or reorientation of the p110gamma , implying that Ras can activate p110gamma without its membrane translocation.

⁷It was demonstrated that Ras induces the sequential activation of PI3K, Rho, and ROCK, leading to activation of Myc through phosphorylation.

⁸All the mutants strongly activated Akt and p70 (S6K) and also induced morphologic changes , loss of contact inhibition , and anchorage independent growth of NIH3T3 cells.

⁹Akt also inhibited tuberin mediated degradation of p27 (KIP1) , thereby promoting CDK2 activity and cellular proliferation.

¹⁰PI3K inhibition suppresses insulin-EGF synergy in Ras and ERK responses.

Knowledge-Based Pathway Extraction and Validation

Gaoxiang Zhou
University of Pittsburgh
gaz11@pitt.edu

Natasa Miskov-Zivanov
University of Pittsburgh
nmzivanov@pitt.edu

1 INTRODUCTION

In the realm of synthetic biology, a crucial aspect is achieving precise control over biological processes. Synthetic biologists aim to regulate specific genes or pathways by manipulating various system components. To achieve this control, pathway analysis plays a vital role in identifying key genes, enzymes, and reactions involved in pathways and understanding their interactions with each other and the environment. By employing pathway analysis, synthetic biologists gain insights into dynamics, regulation, efficiency, and stability of signaling pathways, enabling them to enhance performance and reliability.

One common approach for pathway analysis involves utilizing pathway databases like Reactome [1]. However, such method often overlooks essential details like the positions and roles of genes within pathways, the directions and types of signal transmission, and other valuable biological context. Treating pathways as simple unordered collections of genes discards significant knowledge about the underlying biological phenomena and their interdependencies. To address this limitation and incorporate comprehensive context in the analysis, we introduce knowledge-based (KB) pathway extraction method in this work. This KB pathway extraction method leverages the best first search algorithm and various edge weight assignments.

By incorporating the KB pathway extraction and a rigorous validation scheme, this work aims to provide synthetic biologists with a powerful tool for precise pathway analysis, ensuring more accurate and reliable control over biological processes.

2 METHODOLOGY

2.1 Overview

During the pathway extraction process, for each source-target node pair of interest and each weight assignment, we first determine an extraction threshold, which denotes the number of top ranked paths to extract according to their path scores calculated from weight assignment.

Upon extractions, we also propose a scheme to validate and assess these pathway extractions (as illustrated in Figure 1 top). This validation method relies on preserving

the biological properties of the modeled system. In essence, **the biological properties should be impacted when removing an important pathway, while the removal of a less significant pathway should not substantially alter the system's properties**. To achieve this, we define a set of properties that hold true for the modeled system, such as gene expression levels or changes in proteins over time.

For each extraction, which comprises multiple directed pathways, we acquire a set of directed edges that represents a set of regulation. Then, we evaluate partial models, constructed by knocking down the corresponding set of edges from the baseline model, with respect to the predefined set of properties. We then calculate probabilities for satisfying each property using statistical model checking and element-based simulations [2].

During the validation process, when we knock down the regulation of element x_i to x_j , we essentially modify element x_j 's update function to eliminate the influence of x_i while keeping the effects of other elements unchanged. In the case of Boolean update functions, this is achieved by setting x_i to a non-controlling value, that is, value 0 in OR operations and value 1 in AND operations.

2.2 Best First Search

Given a weighted directed graph, a source node and a target node, best first search algorithm can find the most "important" path(s) from source to target. Note that interpretation of path importance varies, in other words, given a definition of path importance, the extraction of important pathways could be deduced to problems of maximizing some importance index (or minimizing inversed index) in graph search. Therefore, for weighted directed graph $G(V, E, W)$ with node set V , edge set E and edge weight set W , source node x_s and target node x_t , we aim to find a path that has the minimum weight sum (also known as cost) along it. We apply graph search via maintaining a tree of paths originating at the source and extending those paths one edge at a time until the target is reached. At each node, the search algorithm needs to determine which of its paths to extend. Specifically, it selects the path that minimizes $g(x)$, where x is the next node on the path, and $g(x)$ is the cost of the path from the source node to node x . The trace of nodes along which we reach the target is the path with minimum cost.

2.2 Weight Assignment

Different interpretations of importance will lead to different edge weight assignments. We propose nine attributes and define the corresponding weight for the purpose of minimizing the weight sum.

in-degree: $a_1(x)$ is the size of node x 's direct regulator set. We prefer to extract path with more nodes with high in-degrees (equivalently, low in the reciprocal value of in-degree). Under such interpretation, the weight for edge from node x_i to node x_j is defined as $w_{ij} = 1/a_1(x_j)$.

out-degree: $a_2(x)$ counts how many times an element x occurs in other elements' regulator set. Similar to the interpretation of in-degree, we define $w_{ij} = 1/a_2(x_j)$.

shortest_link: $a_3(x)$ is defined as the length of the shortest path that goes through x and connects given source and target elements. It is obvious that elements occurring in shorter paths linking the source and target elements have higher impact in the relationship between source and target. Therefore, we define $w_{ij} = a_3(x_j)$.

loop_count: $a_4(x)$ counts how many unique loops go through x . A loop in a directed graph is a one-direction path from a node x to the same node. Even a minor alteration to any element within a loop has the potential to be magnified when the loop is iterated multiple times. Thus, we expect paths containing elements that occur in more loops to be more influential, and therefore, we compute the weight as $w_{ij} = 1/(a_4(x_j) + \varepsilon)$. A small constant ε is added to accommodate the cases when $a_4(x_j) = 0$.

non-bias: for a Boolean node, $a_5(x)$ is defined as $(1 - |\Pr\{x = 1\} - 0.5|)$, ranging from 0 to 1. If the state of an element is biased towards 0 or 1 (i.e., the non-bias value approaches 0), the element is robust against perturbations, prevents further signal propagation and thus suggests high impact. We define $w_{ij} = -\log(a_5(x_j) + \varepsilon)$.

edge_influence: $a_6(x_i, x_j)$ is the attribute from a_i^j [3], this attribute is further categorized as $a_{6-uniform}$ and $a_{6-scenario}$, which return the uniform and scenario-dependent analysis of immediate influence of x_i in x_j , respectively. We define $w_{ij} = -\log(a_6(x_i, x_j) + \varepsilon)$.

element_sensitivity: $a_7(x_i, x_j) = a_6(x_i, x_j)/\sum_i a_i^j$. This attribute could also be categorized into $a_{7-uniform}$ and $a_{7-scenario}$, which denote the relative immediate influence of x_i in x_j , normalized by all immediate influences on x_j . We define $w_{ij} = -\log(a_7(x_i, x_j) + \varepsilon)$.

3 CASE STUDY

We use a model of T-cell differentiation into regulatory (Treg) and helper (Th) cell phenotypes as case study [4]. These cells can be distinguished by the expression of specific markers like *Foxp3* and *IL-2*. We also define three

scenarios to conduct scenario-dependent analysis: (1) high antigen dose, (2) low antigen dose, and (3) toggle.

To demonstrate our pathway extraction method, we select two source-target pairs, (*TCR*, *Foxp3*) and (*CD28*, *IL-2*), and apply best first search to find paths with minimum cost for each edge weight assignment. We then create new models by removing these corresponding edges and test the performance using model checking. Specifically, we assess whether steady-state values of *IL-2*, *Foxp3*, *AKT*, and *PTEN* are reached for partial models in each scenario. The property match probability is determined by performing model checking for each model, and we also find average probability for each model across four key properties.

The findings are in Figure 1 bottom, indicating that the removal of pathways associated with a_6 (**edge_influence**) has a profound impact on the model's behavior in scenarios 1 and 2, ultimately resulting in complete system failure. A particularly interesting aspect is the pathway extraction based on scenario-dependent edge influence value (i.e., $a_{6-scenario}$) under scenario 2, which exhibits an even more destructive effect on the model behavior within that specific scenario. When we eliminate pathways with respect to $a_{6-scenario}$, three out of the four key elements exhibit notable deviations from their intended properties. This underscores the pivotal role played by the removed pathways in preserving the desired behavior of the T-cell differentiation process.

4 CONCLUSION

In summary, the knowledge-based pathway extraction method we propose provides a holistic approach to pathway analysis. It effectively overcomes the shortcomings of traditional methods and elevates the precision of incorporating biological context. By integrating this method with a rigorous validation process, we introduce a framework that empowers synthetic biologists with the tools necessary for precise control and optimization of biological pathways and processes. This paves the way for fresh avenues in scientific exploration and practical application.

REFERENCES

- [1] Croft, David, et al. "Reactome: a database of reactions, pathways and biological processes." Nucleic acids research 39. D691-D697.
- [2] Sayed, Khaled, et al. "DiSH simulator: Capturing dynamics of cellular signaling with heterogeneous knowledge." 2017 WSC IEEE
- [3] Zhou, Gaoxiang, et al. "Sensitivity Analysis of Discrete Models and Application in Biological Networks." Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. 2018.
- [4] Miskov-Zivanov, Natasa, et al. "The duration of T cell stimulation is a critical determinant of cell fate and plasticity." Science signaling 6.300 (2013).

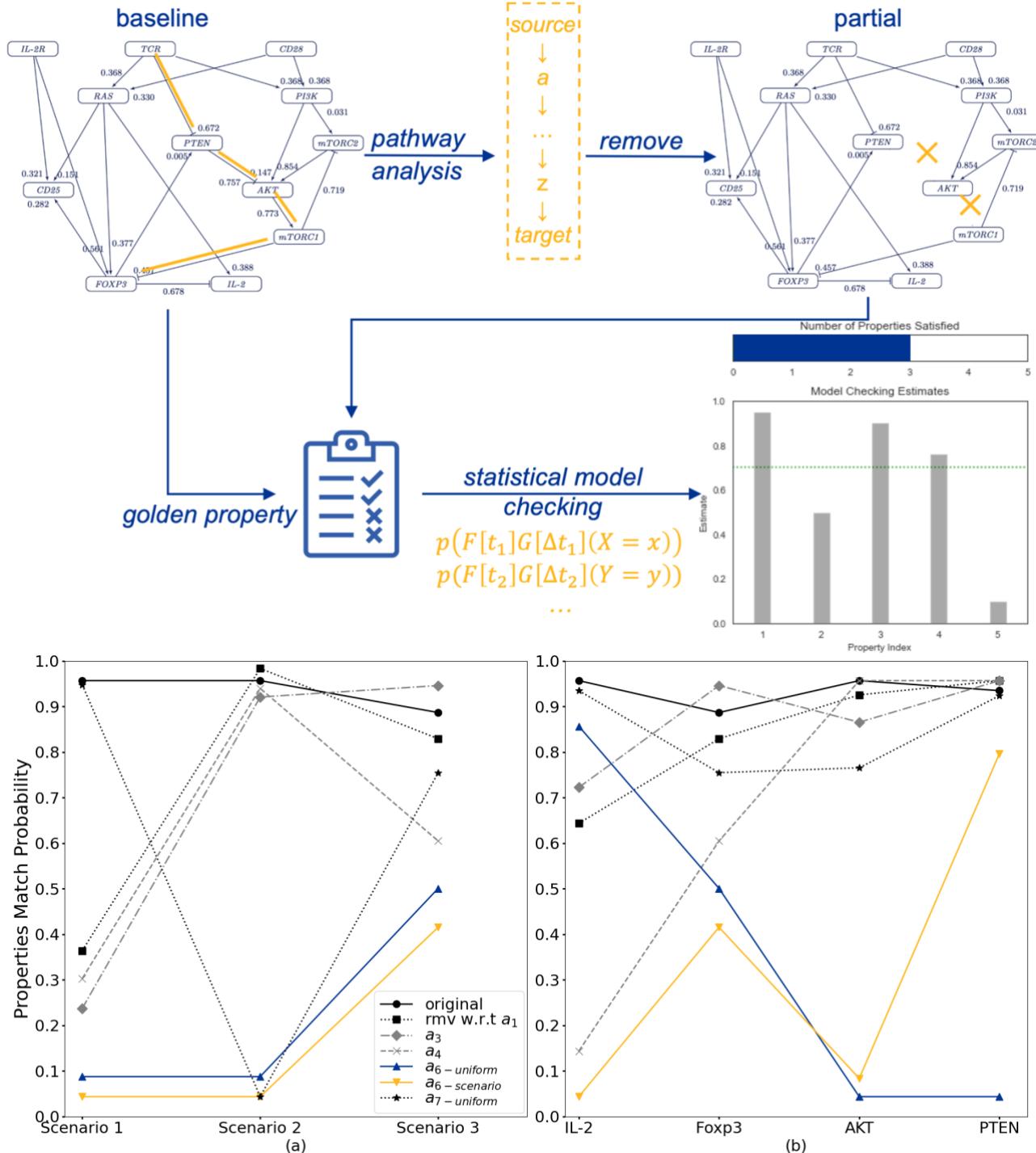


Figure 1. Top: outline of pathway removal and validation procedures using KB pathway extraction and statistical model checking. **Bottom: (a)** results for average match probability for model versions across four key properties, in the original model (solid line with point marker), properties are highly satisfied (i.e., average value of these four probabilities is close to 1). Removing pathways with respect to in-degree (a_1) or out-degree (a_2 , whose curve perfectly overlaps a_1) distribution, minimum length (a_3), and loop count (a_4) affects the properties only to some extent under scenarios 1 and 3, removing pathways with respect to edge influence (a_6 -uniform and a_6 -scenario) significantly breaks down the model (since the property match probability approaches 0) under scenarios 1 and 2; **(b)** four detailed probabilities under scenario 2, removing pathways with respect to the scenario-dependent influence value a_6 -scenario calculated under scenario 2, has an even more destructive effect on the model behavior under that scenario.

Encoding Process Markers with Neural Networks to Simplify the Complexity of Engineering CAR T Cells

Haomiao Luo

University of Pittsburgh
Pittsburgh, PA, United States

Anya Zivanov

Taylor Allderdice High School
Pittsburgh, PA, United States

Natasa Miskov-Zivanov

University of Pittsburgh
Pittsburgh, PA, United States
nmzivanov@pitt.edu

1 INTRODUCTION

A major difficulty facing the field of immunotherapy is breaking the tolerance of self-antigens that the immune system derives from the body it is protecting [1]. In recent years, T-cell engineering has revolutionized immunotherapy, with Chimeric Antigen Receptor (CAR) T cells standing out as a promising approach for cancer treatment [2]. The Food and Drug Administration (FDA) has already approved several CAR-T cell therapies. CAR-T cells send the signal to cancer cells, resulting in localizing tumors and regulating the death of malignant cells. However, the widespread adoption of this technology faces challenges, primarily due to the labor-intensive throughput method and complex process of synthesizing effective CARs. In the early development of CARs about three decades ago, Irving and Weiss determined that a CAR composed of CD8 and the CD3 chain could mediate T-cell activation independently of the endogenous T cell receptor (TCR) [3]. CAR encodes the extracellular domain for tumor cell recognition by the antigen-binding single chain variable fragment (scFv). Once recognition of the tumor is activated, the inter-cellular co-stimulatory domains will start signaling pathways to secrete the cytokines, resulting in the death of cancer cells. The first generation of CARs only consisted of CD3, which is neither persistent *in vivo* nor clinically effective.

Engineering the third generation of CARs is more complex than the other generations since it requires a combination of more co-stimulatory domains. T cells with different codomains exhibit variations in expansions, lifespans, and cytotoxicity. The positions of co-domains also influence the fate of immune cells [4]. To streamline the CAR-T cell design, computational methods, namely machine learning (ML) and neural network (NN)-based approaches trained on data, have emerged to predict CAR-T cell behavior with various receptor structures [5, 6, 7]. To identify additional candidates for CAR, Gordon et al. introduced a method called "CAR Pooling", capable of sorting T cell differentiation among 700,000 combinations. Goodman et al. developed a high-throughput method to screen all the combinations for 40 co-stimulatory and co-inhibitory domains, sorting CAR T cells with CAR expression using a 2A-green fluorescent protein marker. Daniels et al. employed ML to decode CAR T cell phenotype and predict the

cytotoxicity and stemness of CAR T cells. Despite these efforts, existing predictive approaches have demonstrated limited accuracy.

Using expert knowledge and literature, Miskov-Zivanov et al. developed a mechanistic model to simulate intracellular signaling leading to T cell differentiation [8]. This and other existing T cell models can be used as a baseline when merging data and existing knowledge about mechanisms to explore CAR candidates. Furthermore, state-of-the-art natural language processing (NLP) methods and large language models (LLM) can be utilized to collect expert knowledge in an automated manner [9, 10]. We have developed a novel methodology that leverages both data and existing knowledge when predicting the behavior of T cells engineered with different CARs. This new approach aims to enhance the NN ability to learn essential features by incorporating additional information related to the T cell signaling on pathways downstream of CAR candidates. The main objective of the work presented here is to demonstrate a proof-of-concept for this methodology.

2 METHODOLOGY

Cell behavior can be measured by the presence and activity of biological process markers, for example, cytotoxicity and stemness. For the cytotoxicity of T cells, the co-stimulatory domain candidates activate downstream signaling pathways and eventually lead to the secretion of process markers, often referred to as cytokines.

Overview. The two main inputs in our approach contain the information about the relationship between CAR intracellular domain candidates (CARids), on one side, and process markers, on the other. One of the inputs provides this information *explicitly*, with existing knowledge about pathways, collected from literature and the other one includes the information *implicitly*, with experimental data, from Daniels et al. [8], about cells with different CARidc and process marker activities. The output of our method is a quantitative measure of the relationships between CARids and cell behaviors.

Knowledge encoding. The knowledge used in this work is from the corpora of papers found by PubMed to be most relevant to CARids. This corpora is read by the INDRA toolbox, which in turn outputs a list of events between biological entities [11]. We encode the relationship between

process markers (cytokines), and each CARide using the one-hot encoder. This encoder translates existing knowledge into a string of binary numbers, which is then used in conjunction with various NNs to predict CAR-T cell behavior accurately. This process generates a vector, where each element corresponds to a cytokine, with values of 1/0 indicating its presence/absence. The CARide row vectors are then merged to create a matrix that serves as the input for the NN, as depicted in Figure 1. For example, the receptor composed by LAT, Gab1, and IL7 α would be translated as matrix $M \in \mathbb{R}^{3 \times 14}$:

$$M = \begin{bmatrix} 01100100100011 \\ 00000000010100 \\ 01000100010001 \end{bmatrix}$$

if LAT produces process markers 2, 3, 6, 9, 13, and 14, Gab1 produces process markers 9 and 10, and the knowledge of IL7 α contains process markers 2, 6, 10, and 14.

NN training. The data used for training is the cytotoxicity of T cells corresponding to different CARs, which was assayed in lab. Here, we adapted Convolutional Neural Network (CNN) and Long- and Short-Term Memory (LSTM) network as our training models. One structure of CNN+LSTM is illustrated in Figure 2. Both channels are processed independently through the CNN+LSTM layer and are subsequently concatenated to proceed with further processing. We used the random search method to find the hyperparameters of the NN.

3 RESULTS

To evaluate the efficacy of our methodology, we conducted a comprehensive case study, employing a selection of literature search terms to obtain relevant articles. To evaluate the effectiveness and generalizability of the approach, we used cross-validation and a separate testing set. Using NLP-based machine reading, we extracted information from a literature corpus from PubMed. This corpus consists of 30 papers that were found most relevant to CARides.

Furthermore, we compared our proposed approach with an existing ML-based model [7] to showcase the potential in cellular feature prediction, the results are shown in Table 1. The evaluation metrics we used include average R-squared (R^2) and Mean Squared Error (MSE), calculated as the mean and minimum values of 10 validation runs, respectively. From the results presented in Table 1, it is evident that our proposed method demonstrates superior performance in terms of R^2 , indicating a better overall fit to the data compared to the other approach. However, when considering MSE, our approach exhibits slightly higher errors than the other one. To further assess the effectiveness of our approach, we conducted an additional comparison using an independent testing set. The highest predicted result, R^2 , shown in Figure 3, by our approach is higher by 5.6% than the existing state-of-the-art approach. Still, the current results are in the same scalar as the existing published model, and it is essential to acknowledge that the observed enhancement in R^2 could

potentially be attributed to the addition of new features in the input data. Therefore, further validation using real data is necessary to confirm the reliability and generalization capabilities of predictions obtained using our approach.

4 CONCLUSION

In this work, experimental data and expert knowledge were used to train NNs in order to facilitate more effective and efficient CAR-T cell design. Our methodology aims to expedite the synthesis of CAR-T cells with desired functions and fates, minimizing time and resources. Given that this is an early attempt to provide an automated procedure to guide this immunotherapeutic approach, the automated knowledge graphs retrieved from literature still lack accuracy. Through our future work we plan to further explore methods to improve the accuracy of integrating knowledge and data, for the design of CAR-T cells and other synthetic biology-driven immunotherapy approaches.

5 REFERENCES

- [1] Feins, S., Kong, W., Williams, E. F., Milone, M. C., and Fraietta, J. A. An introduction to chimeric antigen receptor (car) t-cell immunotherapy for human cancer. *American Journal of Hematology* 94 (02 2019), S3–S9.
- [2] Kuwana, Y., Asakura, Y., Utsunomiya, N., Nakanishi, M., Arata, Y., Itoh, S., Nagase, F., and Kurosawa, Y. Expression of chimeric receptor composed of immunoglobulin-derived v regions and t-cell receptor-derived c regions. *Biochemical and Biophysical Research Communications* 149 (12 1987), 960–968.
- [3] Irving, B. A., and Weiss, A. The cytoplasmic domain of the t cell receptor chain is sufficient to couple to receptor-associated signal transduction pathways. *Cell* 64 (03 1991), 891–901.
- [4] Weinkove, R., George, P., Dasyam, N., and McLellan, A. D. Selecting costimulatory domains for chimeric antigen receptors: functional and clinical considerations. *Clinical Translational Immunology* 8 (01 2019), e1049.
- [5] Gordon, K. S., Kyung, T., Perez, C. R., Holec, P. V., Ramos, A., Zhang, A. Q., Agarwal, Y., Liu, Y., Koch, C., Starchenko, A., Joughin, B. A., Lauffenburger, D. A., Irvine, D. J., Hemann, M. T., and Birnbaum, M. E. Screening for cd19-specific chimaeric antigen receptors with enhanced signalling via a barcoded library of intracellular domains. *Nature Biomedical Engineering* 6 (06 2022), 855–866.
- [6] Goodman, D. B., Azimi, C. S., Kearns, K., Talbot, A., Garakani, K., Garcia, J., Patel, N., Hwang, B., Lee, D., Park, E., Vykuanta, V. S., Shy, B. R., Ye, C. J., Eymem, J., Marson, A., Bluestone, J. A., and Roybal, K. T. Pooled screening of car t cells identifies diverse immune signaling domains for next-generation immunotherapies. *Science Translational Medicine* 14 (11 2022).
- [7] Daniels, K. G., Wang, S., Simic, M., Bhargava, H. K., Capponi, S., Tonai, Y., Yu, W., Bianco, S., and Lim, W. A. Decoding car t cell phenotype using combinatorial signaling motif libraries and machine learning. *Science* 378 (12 2022), 1194–1200.
- [8] Miskov-Zivanov, N., Turner, M.S., Kane, L.P., Morel, P.A. and Faeder, J.R., 2013. The duration of T cell stimulation is a critical determinant of cell fate and plasticity. *Science signaling*, 6(300), pp.ra97-ra97.
- [9] Holtzapple, E., Telmer, C. A., & Miskov-Zivanov, N. (2020). FLUTE: Fast and reliable knowledge retrieval from biomedical literature. *Database*, 2020, baaa056.
- [10] Ahmed, Y., Telmer, C. A. and Miskov-Zivanov, N., 2021. CLARINET: efficient learning of dynamic network models from literature. *Bioinformatics Advances*, 1(1), p.vbab006.
- [11] Gyori B.M., Bachman J.A., Subramanian K., Muhlich J.L., Galescu L., Sorger P.K. From word models to executable models of signaling networks using automated assembly (2017), *Molecular Systems Biology*, 13, 954.

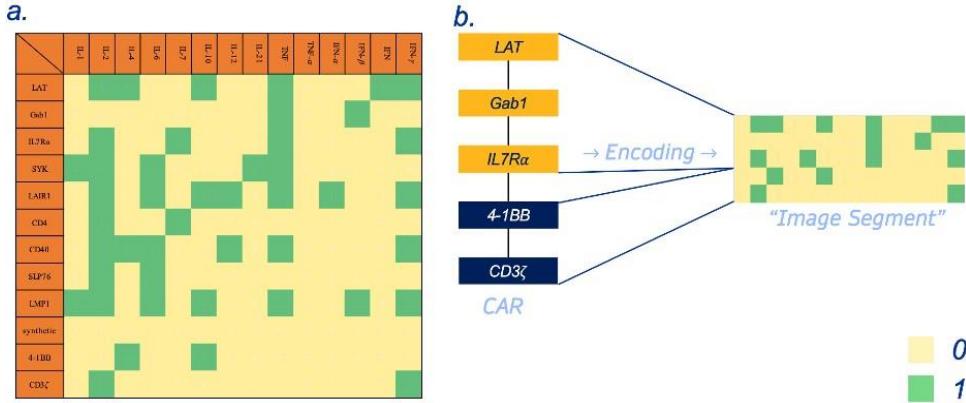


Figure 1: a. Heatmap, the relationship between process markers and source proteins, 0 in yellow and 1 in green; b. An example of the encoding way, the CAR consisting of LAT, Gab1, IL7R α will be encoded to a 5×14 binary image segment by the 2D table in a.

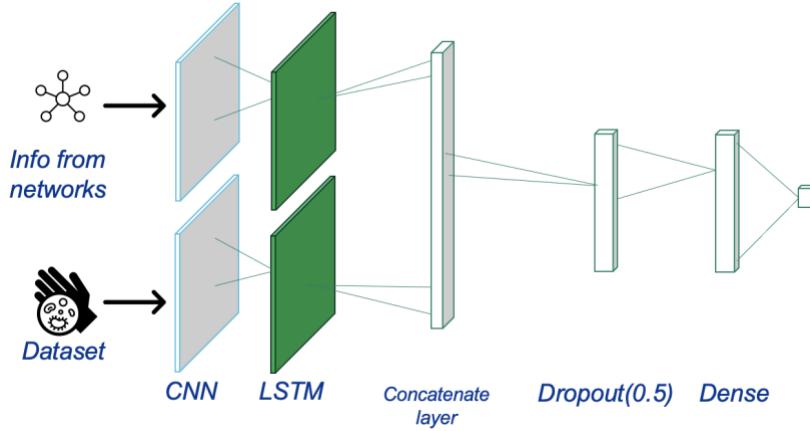


Figure 2: Structure of the CNN + LSTM NN.

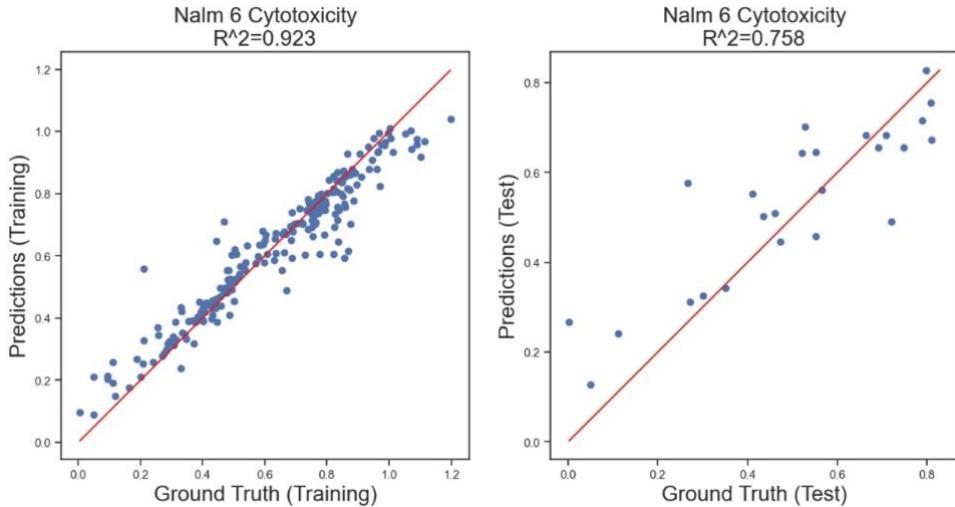


Figure 3: Training data results and R^2 evaluation by an independent set. Left graph: the R^2 value between predictions and ground truth (training data) is 0.923, which indicates a good fit for our neural networks. Right graph: the relevance between the NN predictions and the testing dataset.

Table 1: Comparison of different NN-based approaches.

Model Name	R^2 for 10-cross validation	MSE for 10-cross validation	Average R^2	Average MSE
Daniels et al.	0.7790639846469	0.002407706276827547	0.7349	0.002855
	0.7586944812527932	0.001089058691494258		
	0.731591205600548	0.002505813260953496		
	0.8069579420439886	0.0026741614699660877		
	0.7370699356126569	0.0030301019192742525		
	0.7351644282104646	0.0028685507151754615		
	0.7847506163490039	0.0014996373617922713		
	0.39914161259855985	0.004633374508006201		
	0.5662030263533809	0.005757011736439582		
	0.7772739485806577	0.002241623362026099		
NN with existing knowledge	0.8069031619253737	0.007394923577299734	0.8845	0.004516
	0.9171425991902024	0.0036487367007946793		
	0.8566097441935231	0.0061832574039079775		
	0.8624921605205169	0.006522796776920278		
	0.8391397014067725	0.009119645430698715		
	0.9170385882832085	0.0033494604978087606		
	0.9123687701820337	0.004227237892962219		
	0.8634764282246215	0.00564154213944345		
	0.7416737819184933	0.007359827161305499		
	0.8704931480479448	0.004395839398657315		

DBTL Engineering cycle automation: Improving basic parts characterization in the Learn stage by Automation of the Test stage.

Yadira Boada, Anna Pushkareva, Harold Díaz-Iza, Andrés Arboleda-García, Jesús Picó, Alejandro Vignoni

Synthetic Biology and Biosystems Control Lab, Instituto de Automática e Informática Industrial,

Universitat Politècnica de València (Valencia, Spain).

vignoni@isa.upv.es

1 INTRODUCTION

In the field of Synthetic Biology, the Design-Build-Test-Learn (DBTL) cycle (Figure 1) serves as a fundamental framework for the development and optimization of biological systems [10]. This iterative process involves designing genetic constructs, building them through molecular biology techniques, testing their functionality, and learning from the obtained results to refine future designs. The DBTL cycle has been pivotal in advancing synthetic biology and enabling the engineering of living organisms with desired traits and functions.

While the DBTL cycle has proven its effectiveness in the design and construction of biological systems [12], it is a labor-intensive and time-consuming process that often requires significant manual intervention. As the complexity of bioengineering projects continues to increase, there is a growing need to streamline and automate the DBTL cycle to accelerate the development of biological solutions [10].

There is work being done towards a fully automated DBTL cycle tackling one of the steps at the time; mainly for the automation of the design step [3–5, 9, 15, 17], build step [8, 13], for the test step there are advances in the calibration of the measurements [2, 11], and in the automation of wetlab protocols in general[18]. Finally for the learn step [7, 16, 19]. However, there are not many examples of automation of the test and learn steps combined. In this work we explore the application of automation techniques to the DBTL cycle Test step, specifically focusing on the testing standard bioparts and the further use of the obtained data in a smei-automated characterization of standard bioparts. Standard bioparts, such as promoters, terminators, and coding sequences, form the building blocks of genetic circuits and are widely used in bioengineering projects. Efficiently testing and characterizing these bioparts is crucial for their reliable integration into larger genetic systems and the predictable behavior of engineered organisms [6].

Automation has the potential to revolutionize the testing and characterization of standard bioparts by improving throughput, reliability, and reproducibility [10]. By leveraging advanced laboratory robotics, high-throughput screening

methods, and data analysis algorithms, automation can significantly accelerate the iterative testing process and provide valuable insights for bioengineers. Furthermore, automation can enhance the standardization and quality control of bioparts [9], ensuring their compatibility and reliability across different projects and laboratories [1].

2 AUTOMATION OF THE TEST STEP

This work is focused on the Test and Learn steps of the DBTL cycle, so the starting situation is where there exist several genetic constructs in the lab, and we need to test them and learn from these experiments to improve the models and characterize the used bioparts. First, we deal with the Automation of the Test step. For this we implemented an automated protocol combining the Agilent Biotek plate reader (Figure 2) with the Opentrons OT-2 (Figure 3).

This workflow implements the protocol for the normalization of initial concentrations of 7 bacterial culture samples for a 16 hour incubation/measurement experiment. Specifically, the protocol is divided into two parts, the first is a 1:4 dilution of the culture in the culture medium Minimal Media M9 salts plus 20 % of glucose (blank), this dilution will be measured and the optical density (OD) of the 7 samples will be taken so that in the second part of the protocol all the culture are normalized to an OD of 0.05. Later on, these dilutions are distributed into the measurement 96 well-plate. To carry out this whole process it is necessary to have a specific configuration of the OT-2 and OT-2 deck (Figure 4), to make use of the Agilent Biotek Cytation measurement program, and to use the Jupyter notebook linked to the OT-2 to run the protocol.

The use of this protocol establishes the systematic preparation of the 96 well-plate for the growth/measurement experiment, by ensuring a constant initial concentration of cultures across the plate.

Once the experiment is done, we can have a big dataset of absorbance and fluorescent measurements from the selected samples.

3 USING DATA IN THE LEARN STEP

Using the previously explained workflow for the Test step of the DBTL cycle, we obtain a dataset of measurements for the two genetic circuits shown in Figure 5.

We have 3 technical replicates per circuit with 10 AHL induction concentrations. While incubating at 37°C and 230 rpm in a high-speed double orbital shaking, the absorbance was recorded at 600nm, and the fluorescence was measured at 530nm with an excitation of 488nm for 16 hours. The calibration of both absorbance and fluorescence was done using our calibration protocol [2, 11].

With this dataset, we perform a parameter identification by taking a induced protein production model including the quorum sensing system:

$$\left\{ \begin{array}{l} \dot{R} = \frac{C_N k_R p_R}{d_{mR} + \mu} - (d_R + \mu)R \\ \dot{G} = \frac{C_N k_G p_G}{d_{mG} + \mu} \left(\alpha + \frac{(1-\alpha)(A)^n}{(\frac{k_{dlux} k_r C_N}{R})^n + (A)^n} \right) - (d_G + \mu) \cdot G \\ \dot{A} = D \left(\frac{V_{cell}}{V_{ext}} A_e - A \right) - (d_A + \mu)A \\ \dot{A}_e = ND \left(A - \frac{V_{cell}}{V_{ext}} A_e \right) \\ \dot{N} = \mu N (1 - N/N_{max}) \end{array} \right. \quad (1)$$

where N is the number of particles, C_N is the copy number, $k_{G,R}$ is the transcription rate of the protein, $p_{G,R}$ is the translation rate of the protein, $d_{mG,R}$ is the degradation rate of mRNA, $d_{G,R}$ is the degradation rate of the protein, μ is actually the maximal growth rate taken from the experiments, and G, R are the protein concentrations (GFP and LuxR respectively). In the first place, we perform parameters estimation obtaining the parametric values shown in Table 2 for static data (one data point per induction) using the Genetic Algorithm of [14]. In particular from the dataset, we used $\mu = 0.02321 \text{ min}^{-1}$ for both low and high copy devices. After that, we just simulate the whole dynamic model using the estimated parameters changing only the copy number (C_N), and we do this for two cases: no induction and full induction (300nM). By doing this, we use the same identified parameter values to predict the dynamics of protein production after induction as we have characterized the parts involved in these devices: the Plux promoter (k_G , k_{dlux} , α and n), the RBS (p_{R2}) and the protein degradation (d_G). As shown in Figure 7, we obtain a very good prediction of the dynamics only using one set of parameters that characterize the bioparts.

4 CONCLUSIONS

This paper proposes an automation of the Test step of the DBTL and the use the experimental data obtained for characterization of standard bioparts. With a model of induced protein production and the experimental data we can obtain parameter values that allows us to accurately predict

the protein production level of another device only using the maximal growth rate of that construct and changing the value of the copy number parameter. This paves the road for modeling complex devices and using these models to obtain sensible prediction of their outputs. The integration of automation into the DBTL cycle holds great promise for advancing the field of bioengineering and synthetic biology. It not only increases the efficiency and reliability of biopart testing but also enables the exploration of larger design spaces and the rapid prototyping of complex genetic systems. Ultimately, automation can facilitate the development of novel bioengineered solutions with enhanced functionality and applicability in areas such as healthcare, biomanufacturing, and environmental sustainability. By embracing automation, we can overcome the limitations of manual approaches and unlock the full potential of bioengineering to address pressing societal challenges and pave the way for innovative biological solutions.

ACKNOWLEDGMENTS

This research was funded by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 grants number PID2020-117271RB-C21, TED2021-131049B-I00 and GVA grant CIAICO/2021/159. A.P. is a recipient of a "Práctica de Empresa - Plan de ayudas ai2 a la I+D+i 2021". H.D.I. holds an "Contrato predoctoral para la formación de doctores, convocatoria 2021" (PRE2021-098767) from the Agencia Estatal de Investigación. A.A.G. thanks Grant PAID-01-21 Programa de Ayudas de Investigación y Desarrollo - Universitat Politècnica de València. 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).

REFERENCES

- [1] BEAL, J., FARNY, N. G., HADDOCK-ANGELLI, T., SELVARAJAH, V., BALDWYN, G. S., BUCKLEY-TAYLOR, R., GERSHATER, M., KIGA, D., MARKEN, J., SANCHANIA, V., ET AL. Robust estimation of bacterial cell count from optical density. *Communications biology* 3, 1 (2020), 512.
- [2] BEAL, J., TELMER, C. A., VIGNONI, A., BOADA, Y., BALDWYN, G. S., HALLETT, L., LEE, T., SELVARAJAH, V., BILLERBECK, S., BROWN, B., ET AL. Multicolor plate reader fluorescence calibration. *Synthetic Biology* 7, 1 (2022), ysac010.
- [3] BOADA, Y., PICÓ, J., AND VIGNONI, A. *Multi-objective optimization tuning framework for kinetic parameter selection and estimation*. Methods in Molecular Biology, vol 2385. Springer US, New York, NY, 2021.
- [4] BOADA, Y., SANTOS-NAVARRO, F., VIGNONI, A., AND PICÓ, J. Optimization of the dynamic regulation in a branch-in metabolic pathway. *IFAC-PapersOnLine* 55, 7 (2022), 119–124.
- [5] BOADA, Y., SANTOS-NAVARRO, F. N., PICÓ, J., AND VIGNONI, A. Modeling and optimization of a molecular biocontroller for the regulation of complex metabolic pathways. *Frontiers in Molecular Biosciences* 9 (2022).

Table 1: OT-2 Procedure

Step	Description
1.	Set up the OT2 as shown in the Figure 3. Pipettes used are the Multichannel P300 in the right mount, and the Single Channel P1000 in the left mount.
2.	Set up the deck as follows (also shown in Figure 4): SLOT 1: Opentrons 15 tube rack with falcon 14 ml round SLOT 2: Porvair 96 deep well plate 2ml conical SLOT 3: Genier bio coldblock 96 well plate 400ul SLOT 4: Opentrons 6 tube rack with falcon 50 ml conical SLOT 8: Opentrons 96 tip rack 1000ul SLOT 11: Opentrons 96 tip rack 300ul
3.	In the third step the different liquids that the OT2 will handle will be added. The order in which the samples are placed is of great importance for the correct functioning of the protocol, in this case we have chosen the sequence A1, A2, A3, A4, A5, B1 and B2 (Figure 4).
4.	The fourth step is the calibration and adjustment of the offsets. Calibration is performed by loading a python script with a small protocol in the OT2 APP with the necessary labware previously loaded. Once this protocol is selected the user has only to run the Labware Position Check to start the manual calibration. Once the offsets are obtained, do not close the window with the data, because they have to be copied to each of the protocols to be executed in Jupyter.
5.	Execution of the first part of the protocol from Jupyter. For this it is necessary to change the variable action to 2 of the protocol, to start executing the code and then launch the code by pressing Ctrl+Enter.
6.	Once the whole process has been completed, the SLOT 3 plate will carry the samples in the 12 column, and it must be transferred to the plate reader to perform the Absorbance measurement. The OD measurements obtained should be saved into the provided Template Spreadsheet as a comma separated values files. This template calculates the desired dilutions and volumes to be used later by the OT-2.
7.	Upload the Template csv files with the measured values to the OT-2 Jupyter environment.
8.	Execution of the second part of the protocol as done in the fourth step by putting back the 96 well plate to SLOT 3 without the lid, and adding the offsets and changing the action variable to 2 of the protocol to start executing the code and then launch the code by pressing Ctrl+Enter.

Table 2: Estimated parameters for Low copy number device.

Parameter	Value	Units
k_G	6.3	min^{-1}
k_{dlux}	59478	molecules
α	0.026	adim
n	4	adim
p_{G2}	1.18	min^{-1}
d_{mG}	0.234	min^{-1}
d_G	0.0027	min^{-1}
C_N high copy	40	copies
C_N low copy	5	copies

- [6] 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).
- [7] BOADA, Y., VIGNONI, A., AND PICÓ, J. Multiobjective identification of a feedback synthetic gene circuit. *IEEE Transactions on Control Systems Technology* 28, 1 (2019), 208–223.
- [8] BRYANT JR, J. A., KELLINGER, M., LONGMIRE, C., MILLER, R., AND WRIGHT, R. C. Assemblytron: Flexible automation of dna assembly with opentrons ot-2 lab robots. *Synthetic Biology* 8, 1 (2023), ysac032.
- [9] BUECHERL, L., AND MYERS, C. J. Engineering genetic circuits: advancements in genetic design automation tools and standards for synthetic biology. *Current opinion in microbiology* 68 (2022), 102155.
- [10] CUMMINS, B., VRANA, J., MOSELEY, R. C., ERAMIAN, H., DECKARD, A., FONTANARROSA, P., BRYCE, D., WESTON, M., ZHENG, G., NOWAK, J., ET AL. Robustness and reproducibility of simple and complex synthetic logic circuit designs using a dbtl loop. *Synthetic Biology* 8, 1 (2023), ysad005.
- [11] GONZÁLEZ-CEBRIÁN, A., BORRÀS-FERRÍS, J., BOADA, Y., VIGNONI, A., FERRER, A., AND PICÓ, J. Platero: A calibration protocol for plate reader green fluorescence measurements. *Frontiers in Bioengineering and Biotechnology* 11 (2023).
- [12] GURDO, N., VOLKE, D. C., MCCLOSKEY, D., AND NIKEI, P. I. Automating the design-build-test-learn cycle towards next-generation bacterial cell factories. *New Biotechnology* 74 (2023), 1–15.
- [13] KANG, D. H., KO, S. C., HEO, Y. B., LEE, H. J., AND WOO, H. M. Robomoclo: A robotics-assisted modular cloning framework for multiple gene assembly in biofoundry. *ACS Synthetic Biology* 11, 3 (2022), 1336–1348.
- [14] MATHWORKS. How the genetic algorithm works, 2023.
- [15] RADIVOJEVIĆ, T., COSTELLO, Z., WORKMAN, K., AND GARCIA MARTIN, H. A machine learning automated recommendation tool for synthetic biology. *Nature communications* 11, 1 (2020), 4879.
- [16] VIDAL, G., VIDAL-CÉSPEDES, C., MUÑOZ SILVA, M., CASTILLO-PASSI, C., YÁÑEZ FELIU, G., FEDERICI, F., AND RUDGE, T. J. Accurate characterization of dynamic microbial gene expression and growth rate profiles. *Synthetic Biology* 7, 1 (2022), ysac020.
- [17] VIDAL, G., VIDAL-CÉSPEDES, C., AND RUDGE, T. J. Loica: Integrating models with data for genetic network design automation. *ACS Synthetic Biology* 11, 5 (2022), 1984–1990.
- [18] VIDAL-PEÑA, G. Protocol unified design unit, 2023.
- [19] YANEZ FELIU, G., EARLE GOMEZ, B., CODOCERO BERROCAL, V., MUÑOZ SILVA, M., NUNEZ, I. N., MATUTE, T. F., ARCE MEDINA, A., VIDAL, G., VIDAL CESPEDES, C., DAHLIN, J., ET AL. Flapjack: Data management and analysis for genetic circuit characterization. *ACS Synthetic Biology*

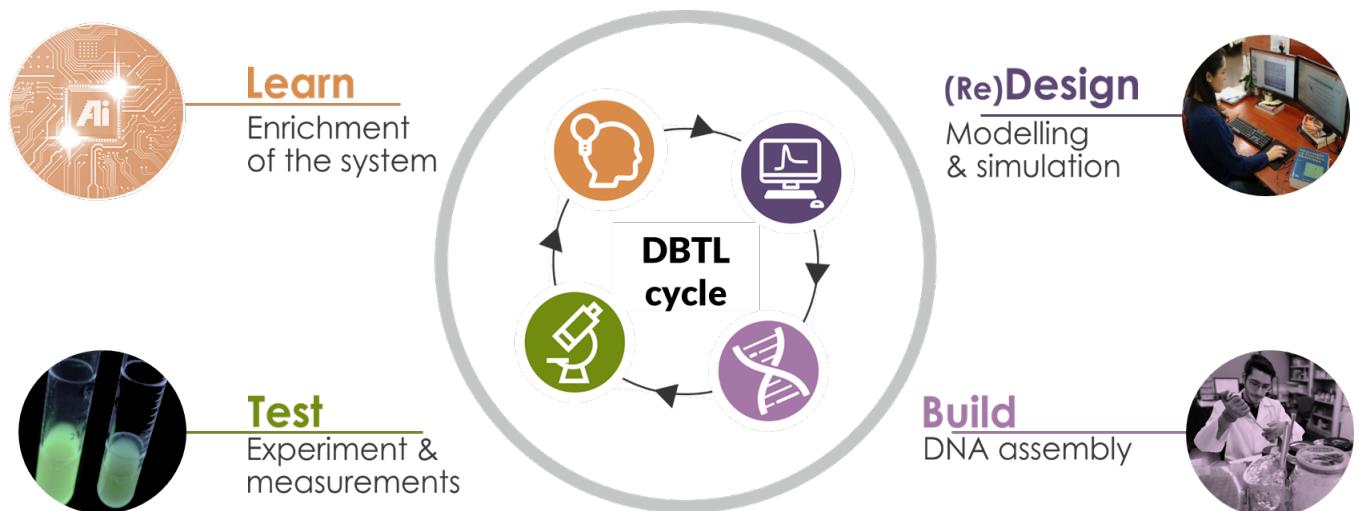


Figure 1: Design, Build, Test and Learn bioengineering cycle used in Synthetic Biology.

10, 1 (2020), 183–191.



Figure 2: Agilent Bioteck Cytation 3 plate reader. This plate reader allows us for incubation at 37°C, agitation, and measurement of both absorbance and fluorescence.

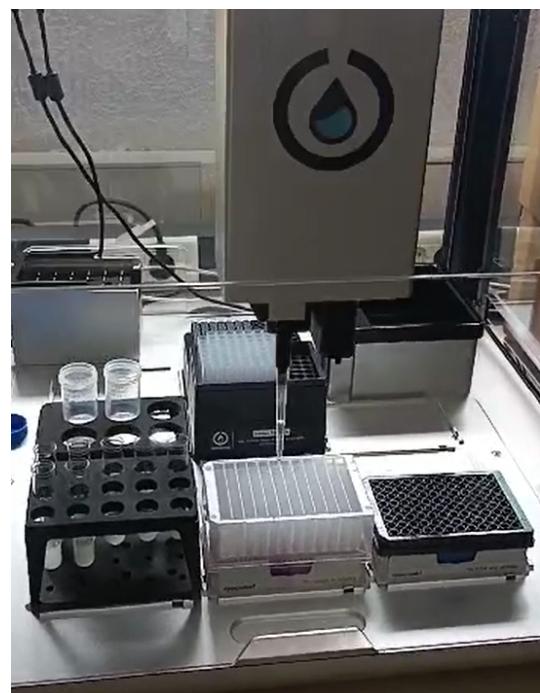


Figure 3: Opentrons OT-2 liquid-handling robot. The OT-2 is already prepared to execute the protocol with all the labware in the right position. The pipettes used are the Multichannel P300 on the right mount and the Single Channel P1000 on the left mount.

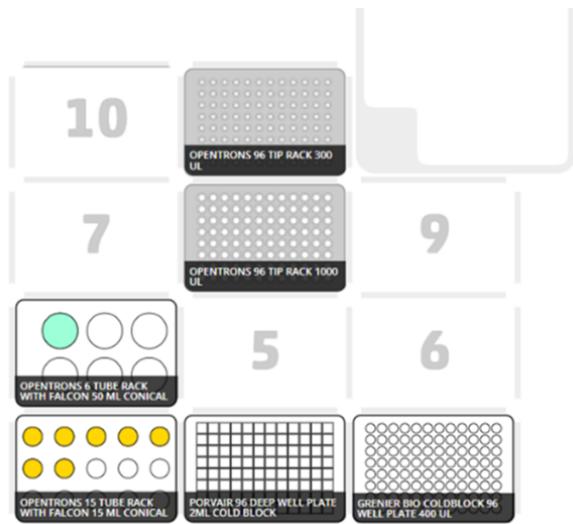


Figure 4: Deck of the Opentrons OT-2 liquid-handling robot prepared for the protocol.

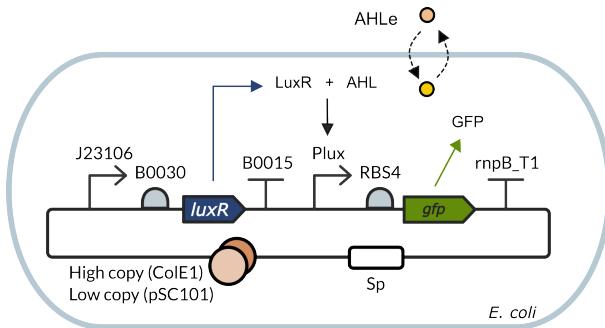


Figure 5: A constitutively expressed LuxR with RBS BBa_B0030 and Promoter BBa_J23106 together with GFP expressed under the control of Plux promoter (BBa_R0062) with a weak RBS (BBa_B0032) in a high copy plasmid (ColE1 ori) and a low copy plasmid (pSC101 ori).

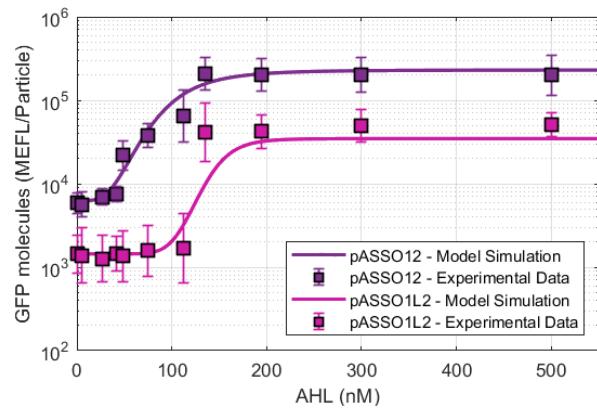


Figure 6: Experimental data and optimized model for the two devices obtained using the automated workflow for the Test step. Fluorescence calibrated in MEFL/Particles for different AHL induction concentrations. High copy number device in Dark purple, low copy number device in magenta, experimental data in squares.

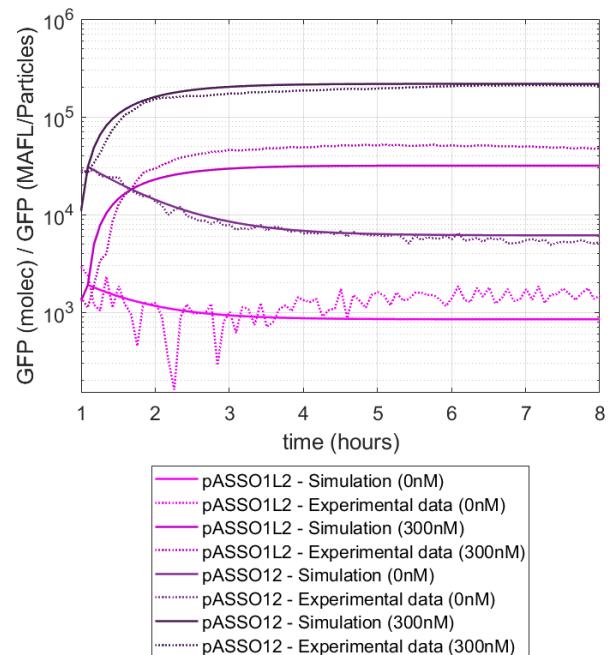


Figure 7: Predicted protein production dynamics after induction for the low and high copy number devices, both for no induction (0nM of AHL) and full induction (300nM of AHL).

Multi-Site Mutagenic Protein Library Design with Controlled Annealing Temperature

Yehuda Binik
Akira Takada
Ayesha Chaudry
The College of New Jersey
Ewing, NJ, USA

Georgios Papamichail
New York College
Athens, Greece

Dimitris Papamichail
The College of New Jersey
Ewing, NJ, USA
papamicd@tcnj.edu

Introduction

Mutant libraries representing protein variants are increasingly used to optimize protein function, by screening for novel proteins with enhanced properties such as expression levels, solubility, stability, or enzymatic activity [9, 10]. Computational design of combinatorial libraries [5, 6, 12, 13] provides a reasonable approach in the development of improved variants. The design of mutant protein libraries typically involves a manual process in which targeted sites for mutation are selected and ambiguous *degenerate* codons (those containing mixtures of nucleotides) are designed to introduce controlled variation in these positions. This is particularly useful in cases where definitive decisions regarding specific amino acid substitutions are non-obvious [10]. The design of the protein variant library is complemented by use of synthesized degenerate oligonucleotides (*oligos*) which enable annealing based recombination. Custom oligonucleotide overlaps allow the targeted introduction of crossovers at selected positions, in turn enabling the desired level and type of diversity in a combinatorial library.

Traditional mutant protein library design methods involve the incorporation of a single degenerate codon (hereafter referred to as *decodon*) at each position where amino acid substitutions are considered. Decodons contain ambiguous (*degenerate*) bases, which are represented as one letter codes, are used to represent (i.e. code) sets of DNA bases.

Online tools such as CodonGenie [11] can aid the design of decodons that code for any provided set of amino acids. The CodonGenie tool ranks candidate decodons by specificity, attempting to minimize coding of undesired amino acids and/or STOP codons. Even so, when using a single decodon to code for a set of amino acids, it is often unavoidable to code for additional unwanted amino acids. Using an example from [11], when coding the non-polar residues A, F, G, I, L, M and V, CodonGenie picks decodon DBK ([AGT][CGT][GT]) as its top choice, which, in addition to the desired set, codes also for amino acids C, R, S, T, and W. In total, the decodon DBK codes for 26 DNA variants, 18 of which code for desired amino acids, and 8 DNA variants for undesired ones.

In our work we explore specifying a set of amino acids by using potentially multiple decodons. The usage of annealing

based recombination of degenerate oligos containing such decodons can produce libraries on the productive portion of the space by eliminating unwanted mutations, therefore improving the yield of beneficial variants and the overall quality of the library. In turn, this method can significantly reduce labor costs assaying the pool of variants, at the expense of additional oligo synthesis, whose comparative cost is modest. We further use the design of minimum cardinality decodon sets specifying any amino acid set (henceforth referred to as AA-set) to create an algorithm that, given a target protein mutant library, it designs oligos whose assembly generates the target library without any unwanted variants, while minimizing the total cost of DNA synthesis.

Problem Definition

The input to our problem consists of an amino acid sequence of length m and a list p of b positions $p_i, 1 \leq i \leq b$. For each given position, we are provided an amino acid set $aa_i, 1 \leq i \leq b$ of desired amino acid substitutions. This input represents a protein variant library L with size $|L| = \prod_{i=1}^b |aa_i|$.

The desired output to our problem is a set of partially overlapping oligos that, once assembled, generate all $|L|$ mutant protein variants in the target library, and only those. In addition, the total number of DNA nucleotide bases in the produced oligos, necessary to assemble the target library, is minimized.

Traditionally, mutagenic protein variant libraries are constructed by incorporating a single decodon at each variable position which generates all target residues. Often that single decodon additionally codes for non-targeted residues or STOP codons. Instead, we could design several decodons, each coding for a subset of the target AA-set, where their union equals the input set. Then, for the creation of the protein variant library, we could synthesize multiple oligos, each incorporating a different individual decodon at the target mutant site.

Our aim is to minimize the amount of DNA we synthesize, which is directly proportional to the total cost of synthesis, while generating a targeted variant library with no undesired variants. Thus we seek to minimize the number of decodons

107 at each variable amino acid position. The question thus becomes, what is the minimum number of decodons necessary
 108 to code any given amino acid set?
 109

110 111 112 Optimal coding of amino acid subsets using 113 degenerate codons for specific organisms

114 We have designed and implemented an algorithm to generate
 115 minimum cardinality decodon sets for a given set of amino
 116 acids using preferential codons for a target organism. Details
 117 of the algorithm and the proof of correctness will be included
 118 in the full version of our paper.

119 Using our algorithm we calculated minimum cardinality
 120 decodon sets for all 1,048,575 possible amino acid subsets.
 121 Our results indicate that 6 decodons are always sufficient to
 122 code for any amino acid subset, where at most 4 decodons are
 123 sufficient to encode more than 90% of all amino acid subsets.

124 We also built a version of our *Decodon Calculator* web tool
 125 [7] that allows researchers to view the minimum number of
 126 decodons needed to code for any input amino acid subset and
 127 provides the optimal decodon set according to the organism
 128 compatibility score in [11]. The organism specific Decodon
 129 Calculator with can be accessed at
 130 <http://algo.tcnj.edu/decodoncalc2/>.

131 132 Optimal oligo design for synthesis cost minimization

133 In creating libraries of targeted protein variants, we enable
 134 substitutions of residues at pre-specified positions with
 135 alternatives drawn from AA-sets of beneficial variants, each
 136 corresponding to a mutation site. We aim to optimize the
 137 combinatorial assembly of all such protein variants without
 138 any undesired residues at any position, while minimizing
 139 the total cost of synthesis. To achieve this goal, we limit the
 140 use of decodons at each mutation site to the exact minimal
 141 set that can code exactly for the corresponding AA-set, using
 142 the algorithm mentioned in the previous section.

143 Each protein in a mutant variant library is translated and
 144 transcribed from synthetic DNA, which is in turn assembled
 145 by joining multiple DNA oligos. It is this process that allows
 146 us to distribute degeneracies among different oligos and
 147 combinatorially combine the oligos to create libraries with
 148 large numbers of variants without synthesizing separate
 149 protein coding DNA sequences for each. Assembly methods
 150 such as the Gibson isothermal assembly [3] provide certain
 151 freedom for varying the annealing temperature of the oligos
 152 based on their overlap length and composition. By carefully
 153 selecting the breakpoints where the sequence is partitioned
 154 into oligos, we can reduce the total amount of DNA sequence
 155 that is required for the synthesis of any given target mutant
 156 library.

157 Based on these observations, we aim to design an algo-
 158 rithm that, given as input

- 160 • an amino acid sequence of length $m = n/3$,
 161 • a list of locations of mutation sites and number of
 162 decodons for each site,
 163 • and length ranges for oligos (l_{min}, l_{max}) and melting
 164 temperature ranges (t_{min}, t_{max}), with $l_{min} \leq l_{max}$,

165 seeks to output a set of oligo set breakpoints, defined as
 166 pairs of start and end positions for each oligo set, such that,
 167 when combinatorially assembled, they generate the targeted
 168 mutagenic library at minimum synthesis cost.

169 Our algorithm uses dynamic programming to exhaustively
 170 consider all possible solutions to our problem, while storing
 171 partial optimal solutions for prefixes of the protein coding
 172 DNA sequence in a two-dimensional array of size $n \cdot (t_{max} -$
 173 $t_{min})$. The sufficiency of a linear partial solution space is
 174 based on the observation that, to compute the optimal cost
 175 of a final oligo of length l_{final} sharing an overlap of length
 176 o_{final} with a prefix of the DNA sequence ending at position
 177 $x = n - l_{final} + o_{final}$, only the cost of synthesis of that prefix
 178 is required as prior knowledge.

179 Our algorithm takes in as input melting temperature ranges.
 180 This is the minimum and maximum desired melting temper-
 181 ature of the oligos generated by our algorithm. The goal of
 182 our algorithm is to generate oligos that have an annealing
 183 temperature within 2 °C of each other. Melting temperature
 184 is used as a proxy for annealing temperature with the as-
 185 sumption that the annealing temperature of DNA is always
 186 around 10 °C lower than the melting temperature. To that
 187 extent, the minimum and maximum allowable temperature
 188 parameters are used to constrain our algorithm to only search
 189 for solutions with a melting temperature within that range.

190 Further details of the oligo design algorithm, including
 191 pseudocode and the proof of correctness, will be included in
 192 the full version of this paper.

193 194 Experimental results

195 We performed computational experiments to compare our
 196 design method against ordering a single synthetic construct
 197 with degeneracies, which utilizes single decodons as sug-
 198 gested by the CodonGenie tool. We assumed the use of the
 199 Gibson method for oligo assembly [2–4], with oligos varying
 200 in length from 40 to 80 base pairs, oligo melting tempera-
 201 tures in the 60°C - 80°C, and a synthesis cost of \$0.38 per
 202 nucleotide, with the simplifying assumption that this cost
 203 includes possible degeneracies.

204 Initial results on application of our design algorithms on
 205 protein libraries of varying complexity as found in Parker
 206 et al. [8] and Chen et al. [1], indicate that, for a modest
 207 increase of the oligo synthesis cost, libraries become orders
 208 of magnitude more specific (up to 65x for libraries with 10^7
 209 variants), devoid of unwanted variants, and thus dramatically
 210 reducing lab costs to evaluate the libraries.

213 REFERENCES

- 214 [1] CHEN, T. S., PALACIOS, H., AND KEATING, A. E. Structure-based redesign
215 of the binding specificity of anti-apoptotic Bcl-xL. *Journal of Molecular
216 Biology* 425 (1) (2013), 171–185.
- 217 [2] GIBSON, D. G., BENDERS, G. A., ANDREWS-PFANNKOCH, C., DENISOVA,
218 E. A., BADEN-TILLSON, H., ZAVERI, J., STOCKWELL, T. B., BROWNLEY,
219 A., THOMAS, D. W., ALGIRE, M. A., MERRYMAN, C., YOUNG, L., NOSKOV,
220 V. N., GLASS, J. I., VENTER, J. C., HUTCHISON 3RD, C. A., AND SMITH, H. O.
221 Complete chemical synthesis, assembly, and cloning of a Mycoplasma
222 genitalium genome. *Science* 319, 5867 (2008), 1215–1220.
- 223 [3] GIBSON, D. G., SMITH, H. O., HUTCHISON 3RD, C. A., VENTER, J. C.,
224 AND MERRYMAN, C. Chemical synthesis of the mouse mitochondrial
225 genome. *Nat Methods* 7, 11 (2010), 901–903.
- 226 [4] GIBSON, D. G., YOUNG, L., CHUANG, R. Y., VENTER, J. C., HUTCHISON
227 3RD, C. A., AND SMITH, H. O. Enzymatic assembly of DNA molecules
228 up to several hundred kilobases. *Nat Methods* 6, 5 (2009), 343–345.
- 229 [5] MEYER, M. M., SILBERG, J. J., VOIGT, C. A., ENDELMAN, J. B., MAYO, S. L.,
230 WANG, Z.-G., AND ARNOLD, F. H. Library analysis of SCHEMA-guided
231 protein recombination. *Protein Science* 12 (2003), 1686–1693.
- 232 [6] PANTAZES, R. J., SARAF, M. C., AND MARANAS, C. D. Optimal pro-
233 tein library design using recombination or point mutations based on
234 sequence-based scoring functions. *Protein Engineering, Design and
235 Selection* 20 (8) (2007), 361–373.
- 236 [7] PAPAMICHAIL, D., CARPINO, N., ABERBACH, T., AND PAPAMICHAIL, G.
237 Decodon Calculator: Degenerate Codon Set Design for Protein Vari-
238 ant Libraries. *12th International Workshop on Bio-Design Automation*
239 (2020).
- 240 [8] PARKER, A. S., GRISWOLD, K. E., AND BAILEY-KELLOGG, C. Optimization
241 of Combinatorial Mutagenesis. *J Comput Biol* 18 (11) (2011), 1743–1756.
- 242 [9] PARKER, A. S., ZHENG, W., GRISWOLD, K. E., AND BAILEY-KELLOGG, C.
243 Optimization algorithms for functional deimmunization of therapeutic
244 proteins. *BMC Bioinformatics* 11 (2010), 180.
- 245 [10] REETZ, M. T., AND CARBALLEIRA, J. D. Iterative saturation mutagenesis
246 (ISM) for rapid directed evolution of functional enzymes. *Nature
247 Protocols* (02 2007), 891–903.
- 248 [11] SWAINSTON, N., CURRIN, A., GREEN, L., BREITLING, R., DAY, P. J., AND
249 KELL, D. B. CodonGenie: Optimised ambiguous codon design tools.
250 *PeerJ Computer Science* 3 (2017), e120.
- 251 [12] TREYNOR, T., VIZCARRA, C., NEDELCU, D., AND MAYO, S. Computation-
252 ally designed libraries of fluorescent proteins evaluated by preserva-
253 tion and diversity of function. *Proceedings of the National Academy of
254 Sciences of the United States of America* 104 (02 2007), 48–53.
- 255 [13] VOIGT, C. A., MARTINEZ, C., WANG, Z. G., MAYO, S. L., AND ARNOLD,
256 F. H. Protein building blocks preserved by recombination. *Nature
257 Structural Biology* 9 (2002), 553–558.
- 258 [266] 267
259 [268] 269
260 [270] 271
261 [272] 273
262 [274] 275
263 [276] 277
264 [278] 279
265 [279] 280
266 [280] 281
267 [281] 282
268 [282] 283
269 [283] 284
270 [284] 285
271 [285] 286
272 [286] 287
273 [287] 288
274 [288] 289
275 [289] 290
276 [290] 291
277 [291] 292
278 [292] 293
279 [293] 294
280 [294] 295
281 [295] 296
282 [296] 297
283 [297] 298
284 [298] 299
285 [299] 300
286 [300] 301
287 [301] 302
288 [302] 303
289 [303] 304
290 [304] 305
291 [305] 306
292 [306] 307
293 [307] 308
294 [308] 309
295 [309] 310
296 [310] 311
297 [311] 312
298 [312] 313
299 [313] 314
300 [314] 315
301 [315] 316
302 [316] 317

PUDU: Simple Liquid Handling Robot Control for Synthetic Biology Workflows

Gonzalo Vidal¹, Carolus Vitalis², Matt Burridge¹, Lukas Buecherl², David Markham¹, Chris Myers², Timothy Rudge¹

¹Newcastle University, Newcastle Upon Tyne, United Kingdom. ²University of Colorado Boulder, Boulder, United States.

1 INTRODUCTION

Lab automation tools have the capability to increase scientific throughput by reducing the time that researchers spend in the lab, reducing pipetting errors and standard deviation, and increasing metadata capture, traceability and reproducibility. One of the first barriers for lab automation is the cost of liquid handling robots. This has been addressed by companies like Opentrons that have substantially reduced the cost of liquid handling robotics by making them open source. Although a lot of liquid handling robotics have been implemented in research and industrial laboratories there is still a challenge in the training of new users and the creation of new protocols. PUDU is a Python package for liquid handling robot control in synthetic biology workflows. It is composed of a set of classes that represent different common protocols in cloning such as DNA assembly, transformation, test plate setup and even calibration. These protocols are easy to modify and adapt to different laboratory needs, reducing the barrier for new students and researchers to use the OT-2. Furthermore, PUDU connects to standards by accepting SBOL build designs as input and generating SBOL files of the protocol products as output.

2 RESULTS

Protocol Unified Design Unit (PUDU) provides a high-level abstraction for liquid handling robot control using a simple object-oriented programming approach in Python. Each object corresponds to a protocol template that can be easily customized changing its attributes. This allows the user to change the samples, volumes, pipette position, labware and more, but the protocol remains the same. To create a new class, or protocol template, users can inherit from existing ones and build on top of them making this process more straight forward, easy to update and reducing errors. For example, lets define a protocol as the series of steps that the human and the machine has to follow to achieve a goal. This goal can be to obtain a sequence of DNA, transform a cell or calibrate a plate reader. Then inside a protocol there are steps that a human has to perform and steps that the machine has to perform. The steps that the machine has to perform are encoded in the script. To define a script, or OT-2 protocol, you need to create a run function that encodes the labware

type and position, as well as the liquid transfers. PUDU simplifies this process of making and using scripts to just two lines of code. Inside the script's run function, the user would need to instantiate a PUDU class and then call its own run method. Users can also chain actions such as creating an SBOL file, capturing metadata in a machine readable format, and/or adding the creation of a xlsx file, capturing relevant information for deck setup, position of reagents and products in a human readable format. For a typical PUDU protocol, the user creates a script and simulates it. The simulation will output an xlsx file with information about reagent position. Then, the user can load the script into the Opentrons App and get information to set up the OT-2 deck and run the script. These scripts automate different stages of the cloning process, the test setup and calibration (Figure 1). All the code, protocols, examples and documentation are publicly available at <https://github.com/RudgeLab/PUDU>

DNA Assembly

The DNA Assembly class has three child classes, SBOL DNA Assembly, Domestication and Loop DNA Assembly. The DNA Assembly class is intended to be a template for default values and structure of other assembly classes, therefore it does not have a run method. The SBOL DNA Assembly class takes an assembly plan as input in SBOL format. This assembly plan must follow the representation of parts and devices for build planning best practice (<https://github.com/SynBioDex/SBOL-examples/tree/main/SBOL/best-practices/BP011>). This input provides the parts to mix and the restriction enzyme(s) to use. The Domestication class takes two inputs as either strings or SBOL Components. The first input is a list or dictionary of parts and the second is the acceptor backbone. It is designed to insert DNA parts from linear fragments (e.g. gBlocks) into plasmids (e.g. universal acceptor backbone pSB1C00) assuming the use of SphI. The Loop DNA Assembly class takes a list of dictionaries as an input. Each dictionary describes the assembly of a combinatorial derivation of parts, for example: {"promoter": ["j23101", "j23100"], "rbs": "B0034", "cds": "GFP", "terminator": "B0015", "receiver": "Odd_1"}. In this example the code takes the Cartesian product of values per each key, building two transcriptional units, where the only difference between them is that one has J23101 and the other has J23100. Dictionary keys

or roles can be defined by the user apart from the receiver, which always needs to be included. The receivers must start with Odd or Even to define the restriction enzyme to use in each assembly.

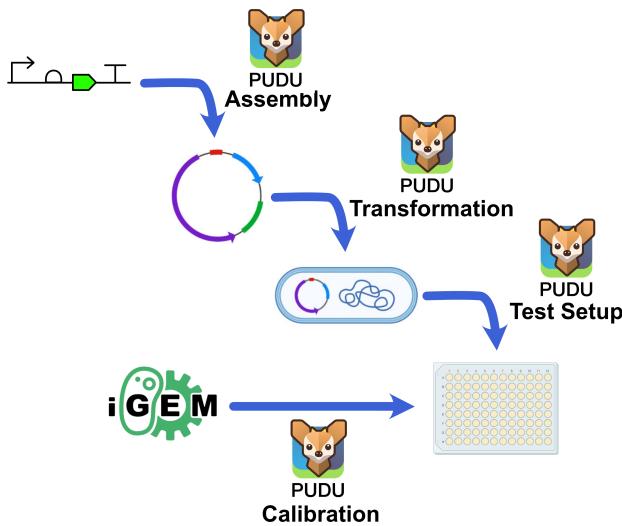


Figure 1: PUDU workflow diagram. Synthetic biology workflow automated using PUDU. The DNA Assembly class automates domestication and the assembly using a design in SBOL or a list of parts using Loop. This process can be automated using a DNA Assembly script. Then, the Transformation class automates the transformation of bacteria with the assembled DNA. Finally, the Test Setup class automates the setup of a 96 well plate with the transformed bacteria under different conditions. Furthermore, The Calibration class automates the preparation of a 96 well plate to obtain calibrated data.

Transformation

The Transformation class has one child class, Chemical Transformation. The Chemical Transformation class takes DNA and strain as an input.

Test Setup

The Test Setup class has one child class, Plate Supplement Setup. The Plate Supplement Setup class takes the samples, inducer and concentrations as an input.

Calibration

The Calibration class has two child classes, iGEM GFP OD and iGEM RGB. iGEM GFP OD is used to prepare a plate for calibration of green fluorescence [2] and the cell count from OD [1]. iGEM RGB OD is used to prepare a plate for calibration of red, green and blue fluorescence [3] and cell count from OD [1].

3 DISCUSSION

In the future we would like to improve the connection with Design tools such as LOICA [7] and SynBioSuite [6]. We would also like to improve the connection to Learn tools by capturing more and better metadata in an semi-automated way. Although this process is difficult to fully automate as users will always be the ones that know what samples and reagents are provided, we think that creating better interfaces would better facilitate the process of metadata acquisition and standardization. We aim to generate a digital plate, ready to be connected to tools like Flapjack [9] and SynBioHub [5]. Compared to other tools such as PyLabRobot [8] or the Opentrons API that help users to create protocols, PUDU has a set of protocols defined as classes where their arguments are required inputs from the user or small modifications, and PUDU captures metadata in a standardized format. Finally, we want to generalize our liquid handling control generating protocols in the Laboratory Open Protocol (LabOP) standard and expand the calibration scripts using absolute protein quantification [4].

4 ACKNOWLEDGEMENTS

The authors would like to thank Jake Beal, the iGEM foundation and the iGEM engineering committee.

REFERENCES

- [1] BEAL, J., FARNY, N. G., HADDOCK-ANGELLI, T., SELVARAJAH, V., BALDWIN, G. S., BUCKLEY-TAYLOR, R., GERSHATER, M., KIGA, D., MARKEN, J., SANCHANIA, V., ET AL. Robust estimation of bacterial cell count from optical density. *Communications biology* 3, 1 (2020), 512.
- [2] BEAL, J., HADDOCK-ANGELLI, T., BALDWIN, G., GERSHATER, M., DWIJAYANTI, A., STORCH, M., DE MORA, K., LIZARAZO, M., RETTBERG, R., AND WITH THE iGEM INTERLAB STUDY CONTRIBUTORS. Quantification of bacterial fluorescence using independent calibrants. *PloS one* 13, 6 (2018), e0199432.
- [3] BEAL, J., TELMER, C. A., VIGNONI, A., BOADA, Y., BALDWIN, G. S., HALLETT, L., LEE, T., SELVARAJAH, V., BILLERBECK, S., BROWN, B., ET AL. Multicolor plate reader fluorescence calibration. *Synthetic Biology* 7, 1 (2022), ysac010.
- [4] CSIBRA, E., AND STAN, G.-B. Absolute protein quantification using fluorescence measurements with fpcountr. *Nature Communications* 13, 1 (2022), 6600.
- [5] McLAUGHLIN, J. A., MYERS, C. J., ZUNDEL, Z., MISIRLI, G., ZHANG, M., OFITERU, I. D., GONI-MORENO, A., AND WIPAT, A. Synbiohub: a standards-enabled design repository for synthetic biology. *ACS synthetic biology* 7, 2 (2018), 682–688.
- [6] SENTZ, Z., STOUGHTON, T. E., BUECHERL, L., THOMAS, P. J., FONTANAROSA, P., AND MYERS, C. J. Synbiosuite: A tool for improving the workflow for genetic design and modeling. *ACS Synthetic Biology* 12, 3 (2023), 892–897.
- [7] VIDAL, G., VITALIS, C., AND RUDGE, T. J. Loica: Integrating models with data for genetic network design automation. *ACS Synthetic Biology* 11, 5 (2022), 1984–1990.
- [8] WIERENGA, R., GOLAS, S., HO, W., COLEY, C., AND ESVELT, K. Pylabrobot: An open-source, hardware agnostic interface for liquid-handling robots and accessories. *bioRxiv* (2023).

- [9] YANEZ FELIU, G., EARLE GOMEZ, B., CODOCEO BERROCAL, V., MUÑOZ SILVA, M., NUNEZ, I. N., MATUTE, T. F., ARCE MEDINA, A., VIDAL, G., VIDAL CESPEDES, C., DAHLIN, J., ET AL. Flapjack: Data management and analysis for genetic circuit characterization. *ACS Synthetic Biology* 10, 1 (2020), 183–191.

Computational analysis of ASR inducible promoter under a range of pH conditions

Maria Jose Mesa-Rodriguez^{1*},
Domenica Cuneo-Campodonico^{1*},
maria.mesa@utec.edu.pe,
domenica.cuneo@utec.edu.pe

Alberto J. Donayre-Torres^{1*}
Universidad de Ingeniería y
Tecnología (UTEC), Lima, Perú¹
adonayre@utec.edu.pe

Martín Gutiérrez^{2*},
Universidad Diego Portales (UDP),
Santiago de Chile, Chile²
martin.gutierrez@mail_udp.cl

1 INTRODUCTION

Cell metabolism requires strict regulation of pH for a normal enzymatic function [1]. Sensing and responding to pH could be a powerful tool to engineer microorganisms for specific functions. In nature, soil microbiomes respond to pH fluctuations [2], commensal bacteria adapt to gastric conditions, and pH-monitoring is essential for industrial fermentation processes [3]. pH adaptation mechanisms have been studied, but genetic elements regulating pH responses, have been minimally explored. Here, for the first time, we selected genetic components controlling the acid-responsive acid shock RNA (ASR), and elaborated a mathematical prediction for the ASR promoter. Our computer simulations will lead future studies *in vivo*, for the fine-tuning of protein production under acidic conditions.

Acidification of the medium triggers survival mechanisms in *Escherichia coli*. Among the most established regulatory systems that respond to external acidification are the lysine decarboxylase (Cad) and glutamate decarboxylase (Gad) systems [4]. However, another regulatory element that remains relatively unexplored is the acid shock RNA (asr) gene. This component is induced under external pH fluctuations between pH 7.0 and pH 5.0 or even lower, having a peak expression between pH 5 and 4.5, resulting in the production of a preprotein with a theoretical mass of 10.6 kDa [5].

The mechanisms driving the expression of the ASR promoter (pASR) are still unclear. However, previous reports suggest its regulation and activation relies on both the PhoB/PhoR and PhoQP-RstBA operons [6][2], as well as on other regulatory elements such as RpoS and H-NS [7].

Previous efforts attempted elucidation of the mechanisms and regulatory components of ASR promoter in acidic response. We propose a model for estimating GFP expression signal driven by ASR promoter in comparison to the synthetic J23100 strong constitutive promoter from *Escherichia coli*. We aim to understand the mechanism of acidic response and protein level accumulation of GFP to be used for production of recombinant proteins in different biotechnological

applications. ASR promoter is a powerful tool that requires further analysis. A limiting factor is the harsh conditions of low pH affecting bacteria biomass and therefore GFP signal readouts. To address those conditions in multiple configurations (time, pH, and promoter strength), we propose a mathematical tool for elucidating ASR genetic switch. We could propose a precise time of response for a future *in vivo* GFP quantification. In this particular genetic switch, considering that acidic pH (4.0) are harsh conditions for bacteria and could impact in GFP signal readouts and low biomass accumulation.

2 SELECTION OF GENETIC ELEMENTS FOR ACID RESPONSE IN BACTERIA

We intend to model components using the *gro* simulator. Therefore, we selected and connected genetic components corresponding to the ASR promoter region, RBS B0034 [9], GFP, and a terminator [6, 8-11]. A control construct bearing the pj23100 (BBa_J23100) promoter [12] driving the expression of GFP [10] fused to a bacterial terminator. The strong constitutive promoter J23100 is included as a reference to compare with ASR pH-inducible promoter GFP signal [10] (Figure 1). The ASR genetic and the pj23100 expression modules (Fig 1A) were modeled as part of a regular high copy number plasmid.

3 ASR PROMOTER COMPUTER MODELING

The described settings were adapted for simulation in *gro* [13]. *gro* is an Agent-based Model 2-D simulator in which synthetic circuits are specified, and population-level behaviors of growing colonies can be prototyped. The simulator uses a binary protein-based specification language to describe circuit structures. The simulator also handles environmental signals and several intercell communication systems. Using these tools, 5 populations were analyzed under distinct pH conditions, following the structure shown in Figure 1: 1) a gene circuit using a J23100 constitutive promoter to activate a reporter gene GFP. The implementation of this circuit was straightforward, as only the expression time parameters required by *gro* had to be set. 2-5) gene circuits using a pASR promoter that is induced by pH levels and using a

GFP reporter gene. This simulation setting was more difficult, since *gro* natively uses binary proteins. Time parameters were adapted (from data in [5]) to replace the intensity level of GFP expression, since the simulator uses binary expression. Thus, cell counts were used to compare the different circuit dynamics. Results for such simulations are shown in Figure 2. The simulation parameter settings considered a running time of 150 simulated minutes for each of the simulations, and a starting population of 1000 bacteria. Previous attempts of experimental analysis reported an exposition time of two and a half hours under acidic conditions. Time points beyond this threshold promote cell decrease due to harsh conditions, and GFP measurements drastically are reduced. We aimed to validate and monitor the performance using *gro*. We evidenced the dynamics of the constitutive promoter are different from the pH 4.5 and 6.8 treatments which show a premature increment around 25 minutes. Experimental data models the time of GFP data collection, and we consider this evidence could be used to conduct early measurement post acidic pH media exposition shift. In fact, in silico modeling helps predict key timestamps during which the researcher will take data for the in-vitro experiments to assess promoter function more accurately. For instance, simulations tend to show that an optimal timestamp to assess promoter activity is early in the experiment, little after 10 minutes and before 90 minutes (as seen in Figure 3).

4 CONCLUDING REMARKS

This study explores the use of the acid-responsive acid shock RNA (ASR) promoter for pH-sensing applications using a computational approach. The simulation conducted in *gro* allowed us to differentiate the expression of GFP using pASR promoter at different pH compared to the J23100 promoter based on time. Even though both promoters achieve similar levels of GFP expression, pASR reaches an ON state quicker than the constitutive promoter. Indicating an ON/OFF switch-like performance driving future experiments settings for their optimization. There are critical time points that should be considered in this system due to harsh conditions of extreme pH exposition. We intend to follow simulations of early readout of 25 minutes to start GFP quantifications. The latter avoids biomass decrease affecting GFP levels. Nevertheless, the goal of characterizing ASR genetic switch aims to determine for how long exposing cells to extreme low pH without affecting the GFP signal due to biomass depletion. This

would allow a promoter fine-tuning and propose future models and experiments. Thus, suggesting that this component could represent a useful tool for biotechnological applications and recombinant protein production. From the findings, we will continue evaluating based on mathematical modeling of the proposed system *in vivo*.

5 ACKNOWLEDGEMENTS

This project was funded by the Provost Scholarship for Undergraduate Research from University of Engineering and Technology, UTEC.

REFERENCES

- [1] W. Aoi and Y. Marunaka, "Importance of pH Homeostasis in Metabolic Health and Diseases: Crucial Role of Membrane Proton Transport," *BioMed Res. Int.*, vol. 2014, pp. 1–8, 2014, doi: 10.1155/2014/598986.
- [2] F. Stirling et al., "Synthetic Cassettes for pH-Mediated Sensing, Counting, and Containment," *Cell Rep.*, vol. 30, no. 9, pp. 3139–3148.e4, Mar. 2020, doi: 10.1016/j.celrep.2020.02.033.
- [3] X. Yao et al., "Synthetic acid stress-tolerance modules improve growth robustness and lysine productivity of industrial Escherichia coli in fermentation at low pH," *Microb. Cell Factories*, vol. 21, no. 1, p. 68, Apr. 2022, doi: 10.1186/s12934-022-01795-4.
- [4] S. Y. Meng and G. N. Bennett, "Nucleotide sequence of the Escherichia coli *cad* operon: a system for neutralization of low extracellular pH," *J. Bacteriol.*, vol. 174, no. 8, pp. 2659–2669, Apr. 1992, doi: 10.1128/jb.174.8.2659-2669.1992.
- [5] V. Seputiené et al., "Molecular Characterization of the Acid-Inducible *asr* Gene of Escherichia coli and Its Role in Acid Stress Response," *J. Bacteriol.*, vol. 185, no. 8, pp. 2475–2484, Apr. 2003, doi: 10.1128/JB.185.8.2475-2484.2003.
- [6] E. Suziedelienė, K. Sužiedelis, V. Garbenciūtė, and S. Normark, "The Acid-Inducible *asr* Gene in Escherichia coli : Transcriptional Control by the *phoBR* Operon," *J. Bacteriol.*, vol. 181, no. 7, pp. 2084–2093, Apr. 1999, doi: 10.1128/JB.181.7.2084-2093.1999.
- [7] K. Shimizu, "Regulation Systems of Bacteria such as Escherichia coli in Response to Nutrient Limitation and Environmental Stresses," *Metabolites*, vol. 4, no. 1, pp. 1–35, Dec. 2013, doi: 10.3390/metabo4010001.
- [8] iGEM, "Part:BBa_K123100," Registry of Standard Biological Parts.
- [9] iGEM, "Part:BBa_B0034," Registry of Standard Biological Parts. http://parts.igem.org/Part:BBa_B0034
- [10] L. Baumgart, W. Mather, and J. Hasty, "Synchronized DNA cycling across a bacterial population," *Nat. Genet.*, vol. 49, no. 8, pp. 1282–1285, Aug. 2017, doi: 10.1038/ng.3915.
- [11] iGEM, "Part:BBa_B0015," Registry of Standard Biological Parts. https://parts.igem.org/Part:BBa_B0015
- [12] iGEM, "Part:BBa_J23100," Registry of Standard Biological Parts, Mar. 13, 2023. http://parts.igem.org/Part:BBa_J23100
- [13] 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.



Figure 1: Schematic illustration of the pASR promoter, inducible by acidic pH and including a reporter gene GFP.

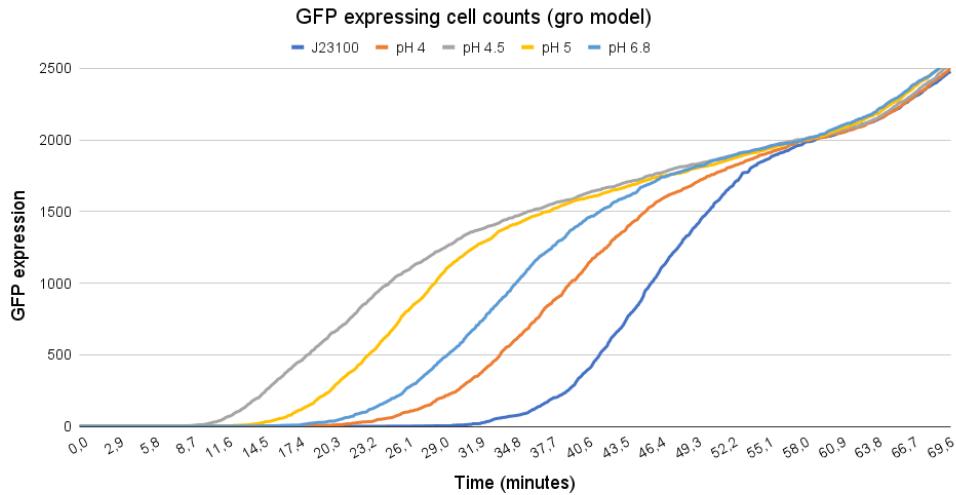


Figure 2: Simulation of GFP expression in cell colonies using *gro*. A constitutive promoter (J23100) inducing GFP was used for one of the simulations, while pH induced promoter (pASR) was used in the other 4 at different pH levels. GFP-expressing cell counts tend to converge as the growth rate does, which does not vary significantly between all cases after $t=58$ minutes. However, the rate at which cells reach their expression before this time characterizes how the constitutive and the ASR promoter differ: pH 4.5 is the fastest, while J23100 constitutive promoter cells are the slowest.

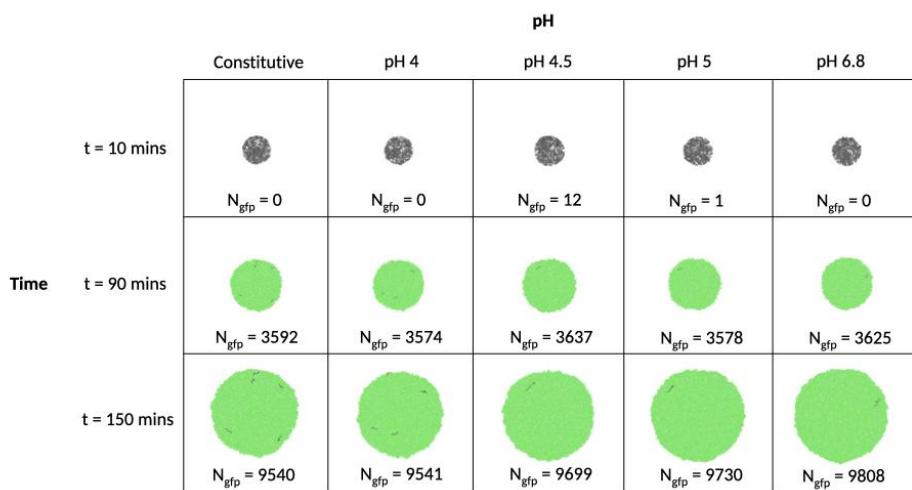


Figure 3: Simulation of GFP expression in cell colonies in *gro*. Constitutive promoter (J23100) and pH induced promoter (pASR) at pH

4, 4.5, 5 and 6.8 induce GFP expression in the simulation. Visual representation of the model at 3 different time frames: 10 minutes, 90 minutes and 150 minutes after induction. At 10 and 90 minutes, the cell count for pH 4.5 is the highest indicating a faster expression as seen in Figure 2 but reaches a plateau at 150 minutes where all cell counts are similar.

How to build and train your ANN (In-Silico)

From zero to hero

Tomás Fuentes

tomas.fuentes@mail.udp.cl
Universidad Diego Portales
Santiago de Chile, Chile

Martín Gutiérrez

martin.gutierrez@mail.udp.cl
Universidad Diego Portales
Santiago de Chile, Chile

1 INTRODUCTION

Cell colonies provides an ideal environment for the execution of multiple algorithms. Nowadays the field of artificial intelligence (AI) has experienced a significant boost in its capacities. These algorithms require high amounts of computational processing power. In fact, cell-based AI algorithms have garnered a significant amount of recent research attention, as evidenced by several studies [1][4][5][6]. This has established a starting point for various types of research, fostering exploration and advancements relating synthetic biology and AI.

Current works on the topic have certain limitations that are being eliminated. One of them, propose a framework that implements heuristic search algorithms (MH) using communication between cells. Other two works involve pre-training a neural network and presents its own non-linear activation function, with manually adjusted weight and bias settings. Another limitation is that during the experiments, some promoters exhibit discrete values, '0' for absence and '1' for presence of agents or proteins. But the goal is to have bacteria themselves perform this task, allowing for weight updates and learning from the environment they are in. Remember in real life, behaviors are exhibited with continuous values, and that is what is proposed in this work.

2 TOWARDS SIMULATED IN-SILICO VERSIONS OF ANN

As mention previously Artificial Neural Networks (ANN) require a large amount of computational power and cell colonies can provides it. Bacteria with their remarkable adaptability and self-regulation capabilities, can be harnessed as key elements in this process. These bacteria would be organized into colonies, forming a modular and scalable structure. Indeed, disconnected colonies are not sufficient to provide a solution to this problem. Therefore, it is necessary for each individual within the colony to be connected and communicate with each other.

Intercellular communication [3] is one of the most important features in bacterial colonies, as it enables the transfer of information in a local-distance range and over long-distance ranges. This is where Quorum Sensing (QS) comes into play,

which is one of the most studied and straightforward mechanisms to comprehend. QS is a two-component signaling system consisting of an emitter and a receptor. The emitter produces autoinducers (AHL), which are received by the receptor component, triggering a programmable response.

Due to the fact that modern ANNs use weights for their learning processes (and these weights are continuous values), intercellular communication can be a way to represent them. Given their characteristics, it is indeed possible to construct a model that simulates an ANN using bacterial colonies.

3 METHODOLOGY

The working approach consists of three prototypes, the first one is a classic experiment of gene expression, where the output is activated by an AND gate logic, having inputs X and Y induce weights S_1 and S_2 (all four are proteins) in the circuit (See Figure 1a). The experiments were conducted in the gro simulator[2], which is a 2D bacterial colony simulator that simulates the individual behavior of each cell. It features a graphical user interface (GUI) used to visually evaluate the simulation results. However, since it is a simulation, we were limited by the capabilities of this tool. This forced us to work with discrete values, meaning that the presence or absence of proteins triggered the secretion of the weights.

For the next prototype, the weights S_1 and S_2 are replaced with signals (See Figure 1b), which brings about a significant change. The weights now become continuous values. This implies the need to consider new variables, such as signal intensity and activation threshold, as they play an important role in emulating the functioning of the perceptron.

Finally, the third prototype additionally incorporates a new signal. This signal is emitted upon a positive response from the perceptron, and its function is to repress the expression of proteins X and Y (See Figure 1c), which affect the expression of S_1 and S_2 . In the end, this signal manages to mimic what would be a form of "backpropagation" for neural networks in cell colonies, where the weights are updated depending on the final output. Both of them were performed in the gro simulator.

All simulations where performed during 263 minutes (simulation time) and each configuration was repeated 10 times. The first simulation only seeks to model a basic and known

for later validation purposes. Then the second and third simulation types introduce intercell communication in form of QS signals to model non-binary weights and feedback (for backpropagation). For more details see Tables 1, 2, 3. Finally these simulations show how the output induces changes in the weight values of the model, exhibiting a similar behaviour to how the backpropagation works in traditional perceptrons.

4 EXPERIMENTS AND RESULTS

The initial tests were conducted to demonstrate how a basic and classic experiment of gene expression works. In Table 1 we observe the truth table of an AND gate performed in the simulation and the Figure 2 illustrate the various output states of the gate.

The second model tests were conducted to demonstrate the importance of signal intensity and activation threshold. Four experiments were performed, varying the configuration of two variables. These variables refers to signal intensity and activation threshold: low or high signal intensity, low or high activation threshold. This parameters are important because configure the non-binary weight behaviour.

Out of the 4 settings used for simulating the prototype, each one presented a particular state that visually explained how the presence and absence of the signals affected the outcome. In Figure 3, the 4 aforementioned states are depicted. In Figure 3a, we observe a green glow because there are blue and magenta signals present, which are sufficient to trigger the activation of the perceptron. In this same image, only the blue color is visible, as the magenta signal is present in low concentration. In Figure 3b, only the blue signal is present, and the colony remains inactive because the amount of magenta signal is not sufficient to trigger activation. Figure 3c is similar to the previous case, but now it is the magenta signal that is expressed while the blue signal is absent. Finally, in Figure 3d, the scenario is the same as in 3a, but with the difference that the magenta signal predominates while the blue signal is present in low concentrations.

Table 3 shows a total of 8 tests that were performed, varying three essential parameters of the repressive signal: diffusion, degradation, and signal intensity (combinations were tested for high or low values). These parameters establish the behaviour of the feedback output. Numerically, the values remain constant because the conditions for each test remain the same. However, visually, the colony exhibits a distinct pattern characterized by peaks and waves. The green glow shown in Figure 4 corresponds to the activation of the GFP protein conditioned through the implementation of the perceptron that learns the AND gate. It is expected that each of the images will be different due to the various characteristics of the conducted tests.

5 FINAL REMARKS AND FURTHER WORK

The conducted tests demonstrate the importance identified in the characterized parameters, showcasing how the signal intensity and the threshold affect the activation of gene expression. Additionally, the presence of the repressive signal effectively represents the backpropagation of neural networks in an interesting way. The updating of the weights (signals) in this approach is inspired by an oscillator circuit. The GFP expression of the colony "oscillates," indicating that the signal intensity is being repressed by the new signal, and once it no longer affects, the circuit reactivates, displaying previously observed patterns. As future work, we are still seeking an explanation for the obtained pattern from a learning perspective (referring to weights), considering its peculiar peaks and waves, which may indicate the learning process of the network itself. Moreover, we plan to expand the prototypes and break the linearity of a perceptron by working with XOR and soon with Convolutional Neural Networks (CNNs) and validate the output versus the respective version of the first model.

This work could be applied in autonomous calibration of colonies, for instance in biomaterial production. If X and Y are base composites that assemble a biomaterial (Z), the perceptron should calibrate to optimize the production of Z. This is being used in the BiOMATERIA project at Universidad Diego Portales.

Despite there is still work ahead, the conducted process has managed to reveal significant findings. For example, the identification of certain variables that have a strong impact within the simulation. It has also confirmed the hypothesis regarding the importance of intercellular communication and its utilization as weights in an ANN.

REFERENCES

- [1] GARGANTILLA BECERRA, A., GUTIÉRREZ PESCARMONA, M., AND LAHOZ-BELTRA, R. Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network.
- [2] GUTIÉRREZ PESCARMONA, M., GREGORIO-GODOY, P., PEREZ DEL PULGAR FROWEIN, G., MUÑOZ, L., SÁEZ, S., AND RODRÍGUEZ-PATÓN, A. A new improved and extended version of the multicell bacterial simulator gro. vol. 6.
- [3] GUTIÉRREZ PESCARMONA, M. E. A new agent-based platform for simulating multicellular biocircuits with conjugative plasmids.
- [4] LI, X., RIZIK, L., KRAVCHIK, V., ET AL. Synthetic neural-like computing in microbial consortia for pattern recognition.
- [5] ORTIZ Y, CARRIÓN J, L.-B. R., AND M, G. A framework for implementing metaheuristic algorithms using intercellular communication.
- [6] SARKAR, K., BONNERJEE, D., AND BAGH, S. A single layer artificial neural network with engineered bacteria.

How to build and train your ANN (In-Silico)

From zero to hero

IWBDA 2023, September 11-15, 2023, Boston

Table 1: Tests conducted on prototype 1 of the AND logical function. Characterization of the inputs X and Y respectively. The last row correlates each configuration with its respective image in Figure 2

	X (False) Y (False)	X (False) Y (True)	X (True) Y (False)	X (True) Y (True)
Starting Cells	200	200	200	200
Exec. time	263.61	263.61	263.61	263.61
Operon Configuration	GFP: AND	GFP: AND	GFP: AND	GFP: AND
Ending Cells	13247	13140	13309	13204
Figure 2 label	a	b	c	d

Table 2: Tests conducted on prototype 2 of the AND logical function. Characterization of the parameters Signal Intensity and Threshold for S_1 and S_2 , respectively. The last row correlates each configuration with its respective image in Figure 3

Low S_1 signal (0.001) High S_2 signal (3) Low S_1 Threshold (0.001) High S_2 Threshold (3)	Low S_1 signal (0.001) High S_2 signal (3) High S_1 Threshold (3) Low S_2 Threshold (0.001)	High S_1 signal (3) Low S_2 signal (0.001) Low S_1 Threshold (0.001) High S_2 Threshold (3)	High S_1 signal (3) Low S_2 signal (0.001) High S_1 Threshold (3) Low S_2 Threshold (0.001)
Starting Cells	200	200	200
Exec. time (simulated min.)	225.6	225.8	225.8
Operon Configuration	GFP: AND X, Y: True	GFP: AND X, Y: True	GFP: AND X, Y: True
Ending Cells	9269	9490	9295
Figure 3 label	a	b	c
			d

Table 3: Tests conducted on the third prototype of the AND logical function. Characterization of the parameters: Diffusion, Degradation, and Intensity of the repressive signal. The last row correlates each configuration with its respective image in Figure 4

Low Diff. Signal (0.0002) Low Deg. signal (0.0002) Low 'Rep' Signal (0.001)	Low Diff. Signal (0.0002) Low Deg. signal (0.0002) High 'Rep' Signal (3)	Low Diff. Signal (0.0002) High Deg. signal (2) Low 'Rep' Signal (0.001)	Low Diff. Signal (0.0002) High Deg. signal (2) High 'Rep' Signal (3)	High Diff. Signal (2) Low Deg. signal (0.0002) High 'Rep' Signal (3)	High Diff. Signal (2) High Deg. signal (2) Low 'Rep' Signal (0.001)	High Diff. Signal (2) Low Deg. signal (0.0002) Low 'Rep' Signal (0.001)	High Diff. Signal (2) High Deg. signal (2) High 'Rep' Signal (3)
Starting Cells	200	200	200	200	200	200	200
Exec. time (simulated min.)	263.61	263.61	263.71	263.71	263.71	263.61	263.71
Operon Configuration	GFP: AND X, Y: True	GFP: AND X, Y: True	GFP: AND X, Y: True	GFP: AND X, Y: True	GFP: AND X, Y: True	GFP: AND X, Y: True	GFP: AND X, Y: True
Ending Cells	20686	18091	18942	17750	19209	17497	19000
Figure 4 label	a	b	c	d	e	f	g
							h

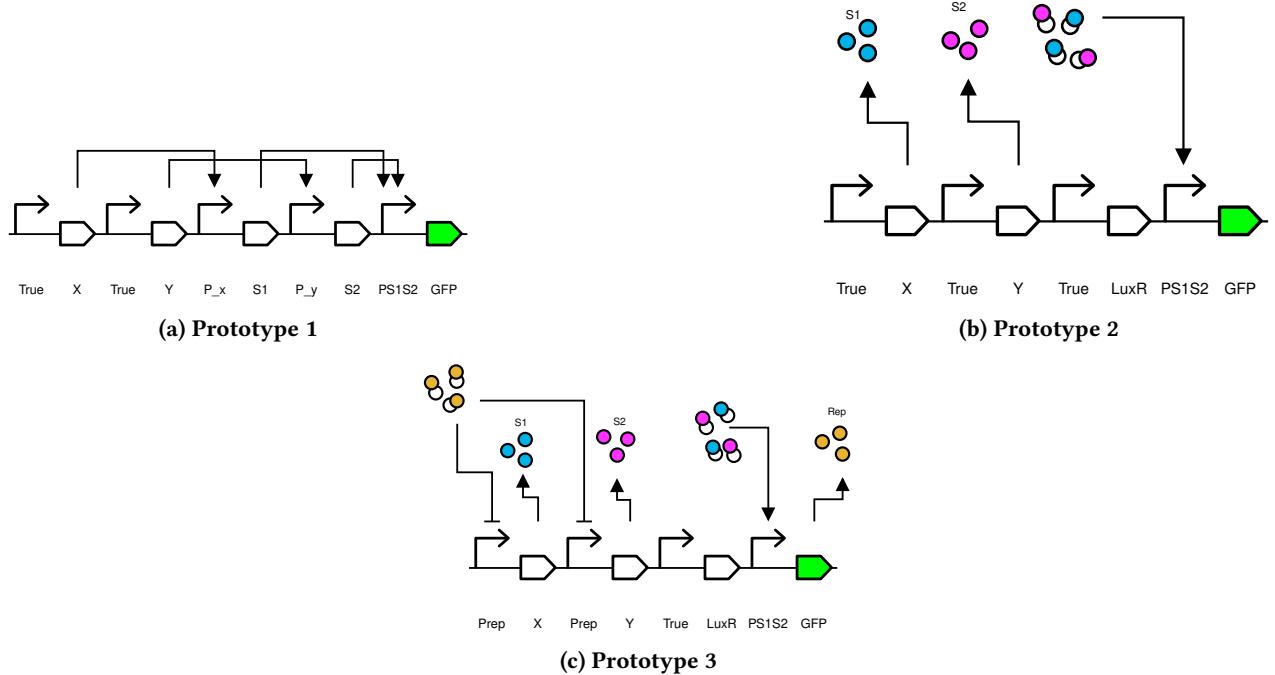


Figure 1: Implemented prototypes. X and Y are inputs proteins that affect the weights S_1 and S_2 . In (a), the presence of X and Y expresses S_1 and S_2 as proteins. In (b), X and Y influence the release of S_1 (blue) and S_2 (magenta) into the environment, which are then detected by LuxR. In (c), the presence of X and Y is conditioned by the presence of the repressive signal (yellow).



Figure 2: Visual representation of the different settings in the Parameter Characterization of X and Y protein in Table 1. The green color corresponds to the activation of the GFP protein by the perceptron that models the AND gate. The figure a, b and c shows how it behaves when there is no presence of proteins. Figure d shows the behavior in the presence of proteins.



Figure 3: Visual representation of the different states in the Parameter Characterization of signal intensity and activation threshold of Table 2. The figure (a) and (d) corresponds to the first and fourth configuration from the table, where the bacterial colony is activated due to sufficient presence of signals S_1 and S_2 . The figure (b) and (c) corresponds to the second and third configuration from the table, where the bacterial colony is deactivated due to insufficient presence of signals S_1 and S_2 . In (b), only signal S_1 (blue) is present, and in (c), only signal S_2 (magenta) is present.

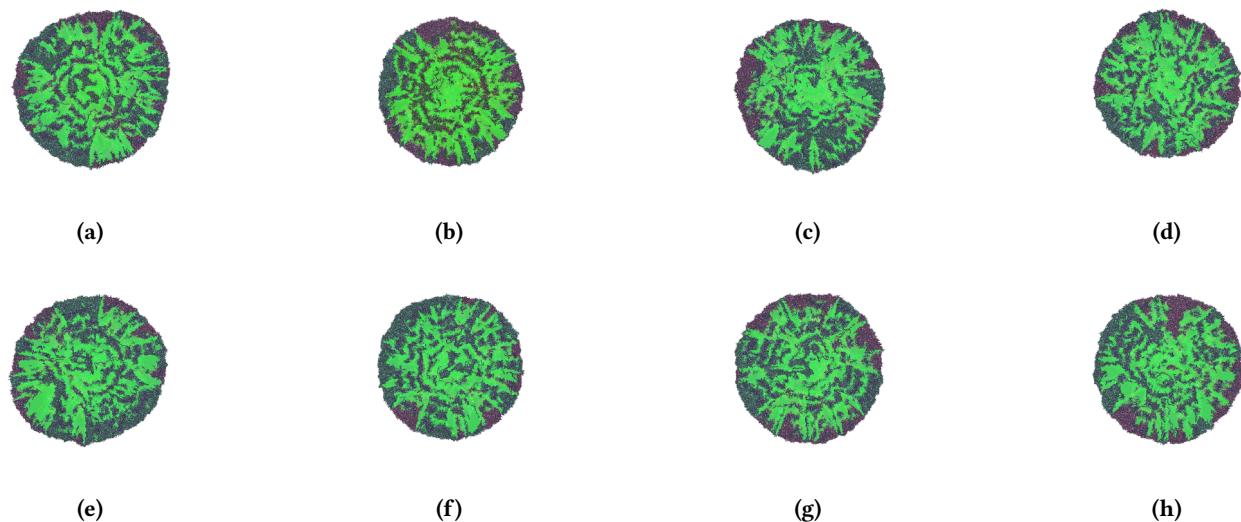


Figure 4: Visual representation of the different settings in the Parameter Characterization of Diffusion, Degradation, and Intensity of the repressive signal in Table 3. The green color corresponds to the activation of the GFP protein by the perceptron that models the AND gate.

AI algorithms for classification and generation of spatial/temporal patterns in cell colonies

Valeria Navarrete, Freddy Aguilar, Martín Gutiérrez

Universidad Diego Portales

Santiago de Chile, Chile

{valeria.navarrete1,freddy.aguilar,martin.gutierrez}@mail_udp.cl

1 INTRODUCTION

Our work builds upon the research by Araya [1], who with its architecture laid the groundwork for the classification and parameter generation for these patterns. We aim to refine this process further using advanced machine learning techniques, to provide a wider and more versatile set of synthetic biology tools and enable the study of more diverse datasets.

A substantial part of our research focuses on creating spatial patterns, specifically Bullseye and Sun patterns, in bacterial colonies, and adapting the original architecture to identify Null patterns. The Bullseye forms concentric circles, the Sun pattern has a central circle with bacterial extensions, while the Null pattern indicates the lack of a specific pattern.

We have two primary objectives. Firstly, we aim to utilize artificial intelligence to generate spatial patterns without simulation. Our research aims to predict bacterial colony patterns with labels using a circuit, but we haven't reached this detail yet. We plan to use AI to create circuits that implement these patterns, a step toward generating new patterns, culminating in a circuit producing specific patterns. These colony patterns will assist in deriving initial parameters (by Araya's architecture) from new AI-generated ones.

Secondly, we want to enhance the range of classifiable patterns by adding the capability to recognize Null patterns. This is crucial in real-world applications, where data often differs from ideal lab conditions. Accurate categorization of such instances as Null patterns can avoid misclassification and improve our understanding of bacterial behavior.

Predicting and generating spatial patterns could accelerate the design of experiments, enable computational testing of hypotheses prior to resource-heavy lab work, and possibly lead to the discovery of new patterns. In this context, one application would be in the area of biomaterials [6]. The mechanisms we are developing facilitate the configuration of these patterns, initially in silico, for subsequent reproduction. This is of utmost importance for biomaterials, as it allows us to precisely define the required locations and proportions in space-time. Our proposed solution directly intervenes in the Design (D) and Build (B) steps of the DBTL cycle, and then, the simulation collaborates with the Test (T) step. In addition, this can be embodied in the BIOMATERiA project of the Diego Portales University (UDP).

2 CGANS AND SPATIAL PATTERN GENERATION

In standard GANs, the generator takes a random noise vector and generates a data sample. The discriminator then attempts to determine whether the data sample is real or fake. The idea is that through this competition process, the generator improves its generation skills to fool the discriminator and ultimately generates samples that are indistinguishable from the real data.

On the other hand, in cGANs, both the generator and the discriminator receive additional information 'y' in addition to the noise vector or the data sample, respectively. This additional information 'y' is used to condition the generation or discrimination process [5].

In this study, we use cGANs to generate artificial spatial patterns that resemble those observed in bacterial colonies. We fed the cGAN with a training dataset of real bacterial colony patterns and their corresponding labels. We then generate new patterns conditional on these labels, which eliminates the need for direct simulations.

3 NULL PATTERN IMPLEMENTATION

While the concept of a Null or "default" pattern is widely recognized in many fields of computer science and AI, it has not been extensively explored in the context of synthetic biology and pattern generation. Null pattern, often designed to act as a default value, essentially encapsulates the "do nothing" or "no specific pattern" condition.

In the context of bacterial colony patterns, the Null pattern serves a similar role. It represents instances where no recognizable pattern is present and provides a contrast to the specific patterns that the model is trained to recognize. It can be a valuable asset in synthetic biology experiments, particularly when studying diverse and unpredictable bacterial behavior.

The implementation of the Null pattern in our approach is achieved by adapting the original architecture to include it as an additional label. If the output from the model does not closely match any of the specific patterns (Bullseye, Sun, etc.), it is classified as Null. This adaptation expands the versatility of our model, allowing it to not only generate specific patterns but also recognize when no specific pattern

is present. This innovation is set to enhance the robustness of our pattern generation and recognition process.

4 EXPERIMENTS, RESULTS AND FUTURE WORK

Firstly, with respect to the generation of patterns without simulation, in our preliminary experiments, we have made modifications to Araya's architecture. Specifically, we replaced the output layer of Araya's CNN [4] architecture, which was originally Dense(4, activation="softmax") [2], with Dense(1, activation='sigmoid') [3]. This change was made to better fit our particular problem space. In the preliminary experiment, the model's performance was analyzed across 100 epochs using batch sizes of 32, 64, and 128.

Figure 1 reveals the discriminator's initial success in discerning real from generated samples, particularly for the batch size of 32. This success, marked by low loss values, is abruptly countered by a spike, signifying an improvement in the generator's performance. In contrast, the discriminator loss for batch sizes of 64 and 128 remains stable throughout, with batch size 128 showing the smallest loss, suggesting potential stability advantages with larger batch sizes.

Generator loss exhibits contrasting patterns. For batch size 32, an initial high loss rapidly decreases and remains low, implying a quick adaptation by the generator to create believable samples. However, the increasing loss values observed for batch sizes 64 and 128 indicate a more challenging environment for the generator as the discriminator's proficiency grows.

As depicted in table 1, the discriminator's near-perfect accuracy at the outset declines over time, especially with a batch size of 32, underlining the increasing difficulty in distinguishing between real and generated samples as the generator improves.

The observed behavior is probably a consequence of the discriminator architecture, predominantly influenced by the Araya model designed for pattern classification. This highly efficient architecture causes the discriminator to initially exhibit high accuracy and low loss values. This highlights a difference in efficiency between the discriminator and generator, as they are not balanced in their operation. This causes the cGAN training to have a high complexity, resulting in generated images that do not come out as expected.

However, as illustrated in figure 2, which shows the original pattern of the dataset with which the cGAN was trained and three generated images with their respective labels. The generator failed to create the desired new temporal patterns, indicating the need for model refinement. These initial results highlight the batch size and performance on training stability and overall model performance, which will inform the next steps of our experimental process.

On the other hand, with respect to the null pattern, we introduced an arbitrary confidence threshold for the Null

pattern and performed experiments to establish the optimal threshold. Table 2 presents the count of falsely identified Null patterns in spatial and temporal patterns at different confidence thresholds in the original dataset.

Temporal patterns begin to exhibit false Null patterns at a 75% threshold, with the count increasing as the threshold raises. Conversely, the first false Null pattern in spatial patterns appears at the 99.999% threshold. This discrepancy arises from the diverse nature of temporal patterns, which have lower scores assigned by the convolutional neural network (CNN) due to time-based training. However, spatial patterns typically score close to 100% due to their more consistent nature.

The reason behind this discrepancy can be traced back to the underlying mechanism of the softmax activation function in the CNN proposed by Araya. The softmax function transforms the network outputs into probabilities that sum to 1. This implies that even for uncertain network outputs, the highest scoring output is considered as the network's prediction. Thus, a high confidence threshold is necessary for efficient Null pattern classification, as even inputs that do not necessarily correspond to any known patterns receive high membership scores.

Considering these findings, we set the confidence threshold at 99.9%, high enough to efficiently classify Null patterns, but not so high as to misclassify known patterns as Null. Figure 3 illustrates the classification results of six input images, including four known patterns and two patterns that should be identified as Null, using this threshold.

As future work, we plan to refine this approach by modifying the architecture at the CNN level, introducing a new model that changes the softmax activation function to a sigmoid activation function. This modification allows us to determine if the input matches any known pattern. If the input does not match, it is classified as a Null pattern. If it does match, the original model classifies which pattern it corresponds to.

REFERENCES

- [1] ARAYA, N. Búsqueda de parámetros iniciales en la generación de patrones espaciales y temporales de colonias bacterianas, Universidad Diego Portales.
- [2] BISHOP, C. M., AND NASRABADI, N. M. *Pattern recognition and machine learning*. vol. 4. Springer, 2006.
- [3] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep learning*. MIT press, 2016.
- [4] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25* (2012).
- [5] MIRZA, M., AND OSINDERO, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
- [6] RATNER, B. D., HOFFMAN, A. S., SCHOEN, F. J., AND LEMONS, J. E. *Biomaterials science: an introduction to materials in medicine*. Elsevier, 2004.

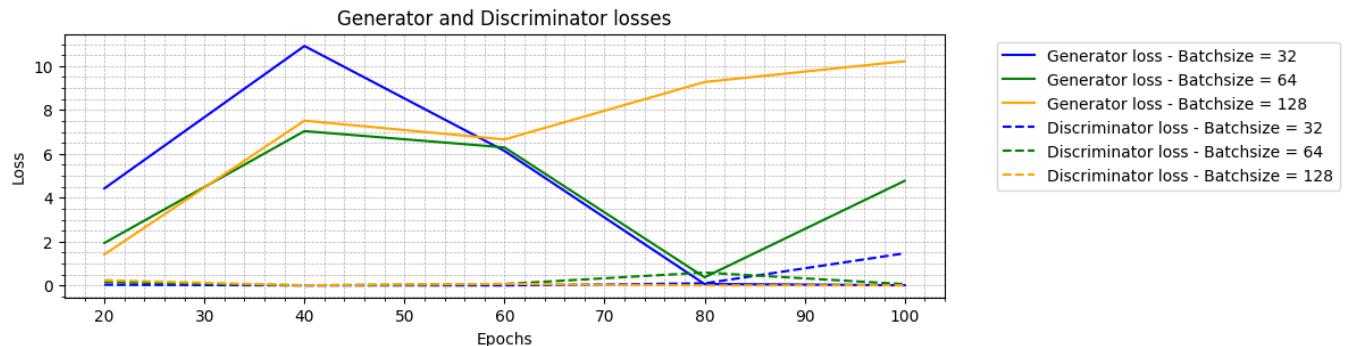


Figure 1: Loss of the discriminator and the generator over 100 epochs with Different Batch Size Sizes. For the generator, with a batch size of 32 and 64 there is a decrease, which is not the case with a batch size of 128. For the discriminator, there is a similar behavior among the 3 batch sizes, however, the 32 batch size shows a growth in the last epochs.

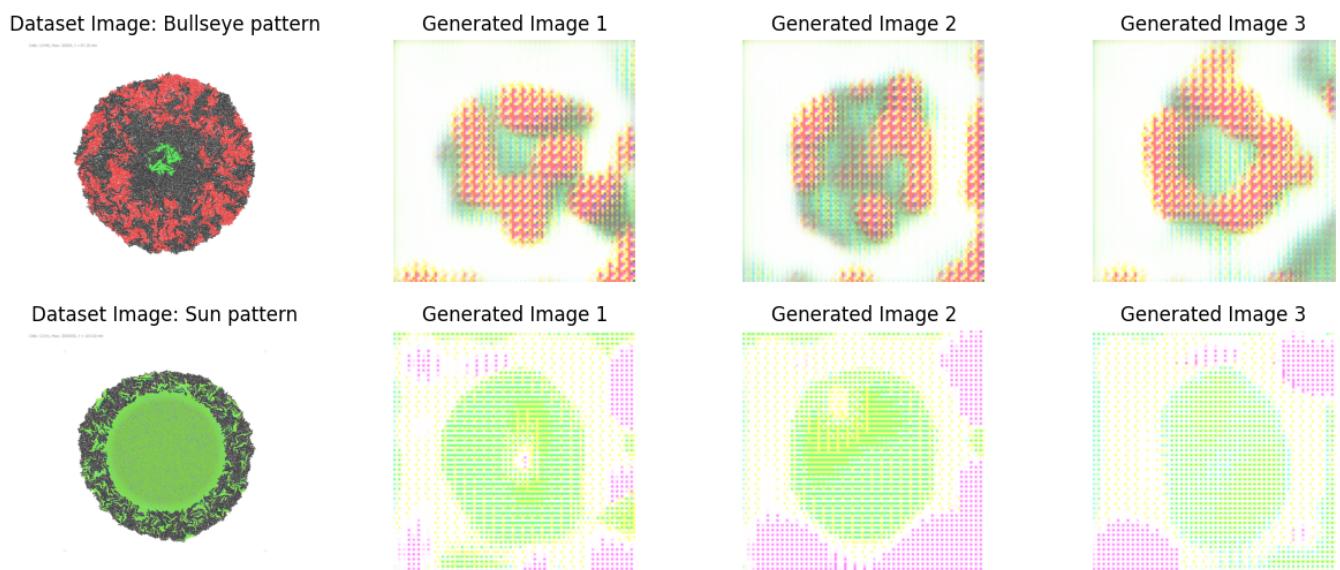


Figure 2: Images generated from the Bullseye pattern and the Sun pattern with cGAN with 500 epoch and a batch_size of 64. Here is the original pattern on the right and then 3 images generated by cGAN with their respective labels. According to the results obtained, it can be seen that these are not conclusive results. It is necessary to explore and change parameters such as the number of epochs, the learning rate, among others.

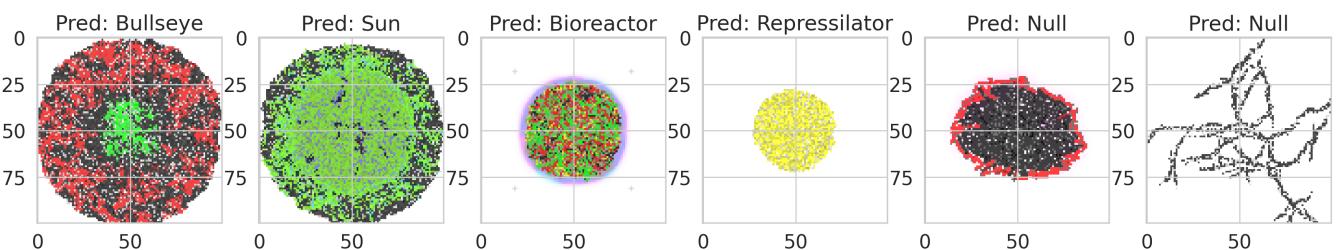


Figure 3: Classification of four known and two Null patterns using the 99.9% confidence threshold. With this threshold, it is possible to correctly classify the inputs of both known and unknown patterns, classifying the last 2 as null. However, tests still need to be carried out on different inputs to test the efficiency of the model and, eventually, to lower the threshold.

Table 1: Comparison of discriminator accuracy with different batch_size sizes over 100 Epochs.

Epoch	Batch size 32	Batch size 64	Batch size 128
20	1	0,992	0,902
40	1	1	1
60	1	0,984	0,980
80	0,953	0,727	1
100	0,593	0,984	1

Table 2: Count of falsely identified Null patterns at different confidence thresholds (false positives)

Threshold	Spatial Pattern	Temporal Pattern
75%	0	1
90%	0	5
99%	0	12
99,9%	0	62
99,999%	1	431

A fully in-silico workflow for treating colon cancer with engineered cells: a study case

Cristobal Hofmann¹, Francisco Salcedo², Martin Gutierrez¹

¹Universidad Diego Portales, ²Universidad de Guadalajara

{cristobal.hofmann,martin.gutierrez}@mail.udp.cl,{f.salcedo}@cucea.udg.mx

1 INTRODUCTION

A lot of work is done in laboratories using trial and error that wastes many valuable resources, so taking the same problems and simulating them on a computer before taking them into a laboratory aids in guiding experiments towards better results without wasting lab resources. Synthetic biology can help with finding new alternatives to fight cancer by simulating cancerous environments using tools that already exist in the field. A workflow for these problems also helps standardize data and have a higher consistency. iGem and SynBioHub databases have more than 20,000 Biobricks to work with, making it a steady base. Our proposal is to use a three-step workflow [2]: design, modeling and simulation to address the problem. Synthetic biology has many software tools pertaining to each of the steps. This work used iBioSim for designing and modeling and gro[1] for simulating. We sought to study a project from iGem and recreate it using this pipeline, since iGem projects have their validated Biobricks already in the database to easily retrieve and use. Since there is a large amount of data within Biobricks, it is possible to develop and train an Artificial Intelligence (AI) to choose the best ones to work with depending on the problem. This helps to have a steady ground to start with. The problem in which the workflow is applied is one a way to detect and inhibit cancer cells in the colon using other cells. Existing data in repositories can help us study if the recreation of these works in a fully computational solution is feasible.

2 PIPELINE AND SOFTWARE

For the first two steps of the pipeline we used iBiosim, since it already has a connection to the iGem and SynBioHub databases it is easier to access Biobricks from the project being recreated. Also, this software uses SBOL[3] and using this standard for the design prevents future problems when exporting data. The fact that we can choose glyphs from a SBOL sheet and then associate a specific part to it makes the design accurate and closely related to the wet-lab counterpart. In this study, we tried to recreate the *B. Hercules the terminator of colon cancer*[4] iGem project from 2012, carried out by the Hong Kong University of Science and Technology. First, we recreate all of the biological circuits one by one choosing the correct symbol and associating it to the appropriate Biobrick. For the modeling phase we used the

same modules as the iGem team did. We constructed three models -one for each module- using the modeling tool. We ran into several problems in this step, some of the components could not be described using Biobricks nor proteins. As for the simulations we used gro. It was used to assess how the BMP2 affects bacterial colonies. Data for configuring the simulation was retrieved from the iGem project: cancer cells were treated with 100 ng/mL BMP2 for 48 hours. We adapted these values to configure the gro parameters by converting ng/mL to $\mu\text{g}/\text{L}$, obtaining a final concentration value of 10 $\mu\text{g}/\text{L}$. For the first simulation we used this concentration. For the second one we used a concentration of 0 $\mu\text{g}/\text{L}$. The simulated cell population was made to grow from 1,000 to 20,000 bacteria with a division time of approximately 40 minutes. When bacteria come into contact with BMP2 and the concentration surpasses the acceptable threshold, they express Red Fluorescent Protein (RFP). The AI was the most difficult part of the project, because component relationships were hard to characterize, assess and generalize. A Convolutional Neural Network (CNN) was the chosen tool to use in learning data relationships. To this end, a first attempt was to use component sankey diagram data in SynBioHub to train this AI. However, many Biobricks or biological parts are too specific, making it impossible to connect a specific part to a frequently used one. The last resort to tackle this problem was to train the AI with a simplified database storing name, kind and amount of uses for the most frequently used promoters, RBS and terminators. The CNN trained with this data was able to choose the most used part depending on which type of Biobrick we needed.

3 EXPERIMENTS AND RESULTS

In the design phase, we recreated the same Biobricks used by the iGem team. The modeling phase used the previously constructed Biobricks and connected the corresponding model components. Simulations explored the application of different concentrations of BMP2, and the observation of bacterial colony behaviour as it came into contact with the BMP2. Finally, the AI tool extended the design phase of the pipeline since it gives us the best options to build the Biobrick. The design phase was successful in that the results produced by iBioSim were the same as in the original work. On the other hand, iBioSim model simulations were really simple. This

stemmed from the fact that some Biobricks or proteins were not available in the software to simulate what was needed. Thus, results were merely the actions of all selected Biobricks connected correctly. In the targeting module, it was not possible to recreate the attachment of one cellular wall to another. In both the anti-tumor molecule secretion module and the regulation and control module in which BMP2 is secreted, we got two Time Series Data (TSD) graphs, in the first case showing an increasing linear line (shown in Figure 1) and in the second an increasing curve (shown in Figure 2). This shows that both models that have BMP2 as an output and composed of different connected Biobricks will produce different results. Both graphs show the amount of BMP2 released over time. However, results on both graphs were the same as with the original circuits. The population level simulations were really successful: when the bacteria came into contact with BMP2 they expressed RFP if the concentration was higher than the acceptable threshold. In the complementary case, RFP was not expressed. The AI was capable of selecting parts for the circuit, but since key data for training was the number of uses, the results did not reflect significant change.

4 LIMITATIONS AND VALIDATIONS

Most of the limitations shown in these experiments were because of the software we chose. In case of the design phase there were complications with some of the glyphs, after downloading their respective SBOL and importing them into iBioSim the design phase was complete. The only way to validate those is through the original work from the iGem team. The modeling phase was successful on building the schematics but, simulating the models was really problematic. Important biological parts or proteins, are not included on iBioSim therefore both of those where only shown visually but doesn't affect the simulations. It was possible to simulate the models but since the simulation time was only 100 seconds so it limited the result graphs, leaving it open to interpretation. In the first module it was not possible to simulate it, since the function of this module is to attach the *B. subtilis* to the cancer colon cell wall. The resulting graph should be a constant straight line on zero and when it attaches the value should change to 1. In the second module the resulting graph is a increasing line and since the function of this module is to secrete BMP2 indiscriminately then the graph should be correct. Lastly the third module function is to control both time and amount of BMP2 secreted. The resulting graph should indeed be an increasing curve, but it should only work for a limited amount of time. So the resulting graph should be either an increasing curve that stops after some time and then it starts from zero again or an increasing curve that after some time decreases at the same speed till it reaches zero based on the function of this

module. Simulations in gro were not what we expected to do, nonetheless the results were really good and important for this experiment. The fact that bacteria reacted to BMP2 concentration and expressed RFP means that this protein indeed affect bacteria, that is why this phase is so important. If bacteria which had the function of expressing a fluorescent protein reacted to BMP2 then it makes sense to assume that changing the function of the bacteria and subdue it to high concentration of BMP2 will also affect them to do the designated function. However the fact that we cannot include Biobricks into the simulation makes it so this phase is not specific for the genetic circuits but the inhibitory protein. The AI implementation was simple, however the training was limited due to the fact that there is no complete Biobrick dataset with type, amount of uses, predecessor, successor, etc.

5 CLOSING REMARKS AND OBSERVATIONS

This project was successful at every step of its execution, but the results were not as good as expected because of the limitations of the chosen software. It is clear that the original goal is feasible, however, the software chosen each had its limitations which where unknown before working on them. Taking this into account, one possibility for improving our results and making them more specific, is to choose a different software combination for the modeling and simulation steps. Another lesson learned from this project, is that no single current software can solve all problems. Therefore specifically tailored software or Domain Specific Languages (DSLs) could be a path to further improve results. In the case of the problem our team chose, an example could be software that emulates dynamics pertaining to colon functioning. This project made clear that the recreation of in-vivo projects within in-silico environments are still in development but are moving in the right direction. The idea of implementing AI has shown that in fact there is a lot of information and parts, but, we did not find a way of organizing them as a complex dataset to train it yet. There is a need for complete datasets. A repository of all of the already known composite Biobricks is also key, so AI algorithms can learn existing patterns.

REFERENCES

- [1] gro the cell programming language.
- [2] ARANCIBIA., F. S. Diseño, modelado y simulación in silico de biosensores bacterianos para la detección de metales pesados en agua para la agricultura de precisión.
- [3] BUECHERL, L., MITCHELL, T., SCOTT-BROWN, J., VAIDYANATHAN, P., VIDAL, G., BAIG, H., BARTLEY, B., BEAL, J., CROWTHER, M., FONTANARROSA, P., ET AL. Synthetic biology open language (sbol) version 3.1. 0. *Journal of integrative bioinformatics* 20, 1 (2023), 20220058.
- [4] THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY. B. hercules — the terminator of colon cancer.

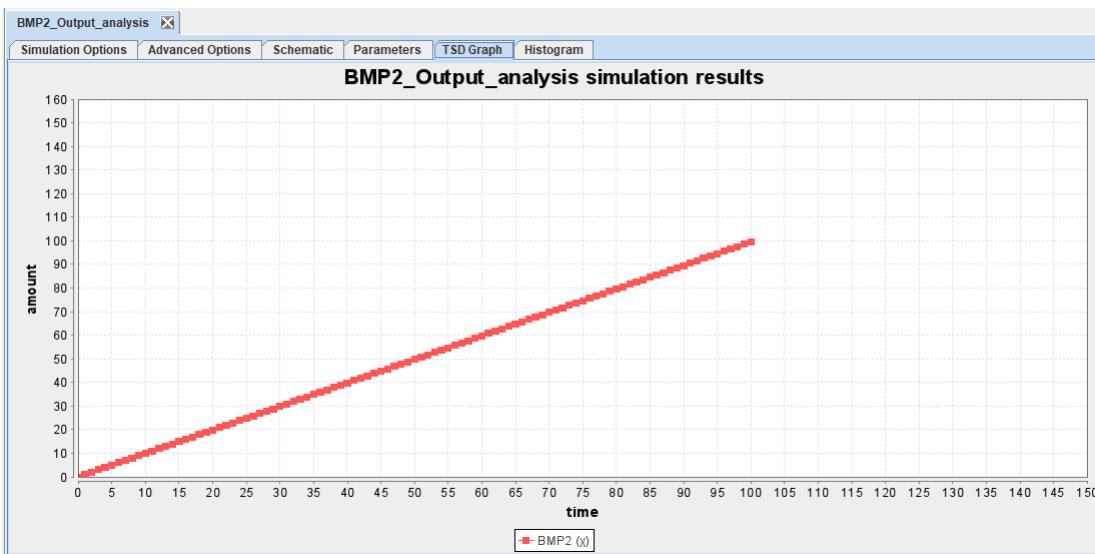


Figure 1: This graph shows the amount of BMP2 release on the anti-tumor molecule secretion module over time. There is no control over the amount of released BMP2, thus it will constantly increase indefinitely.

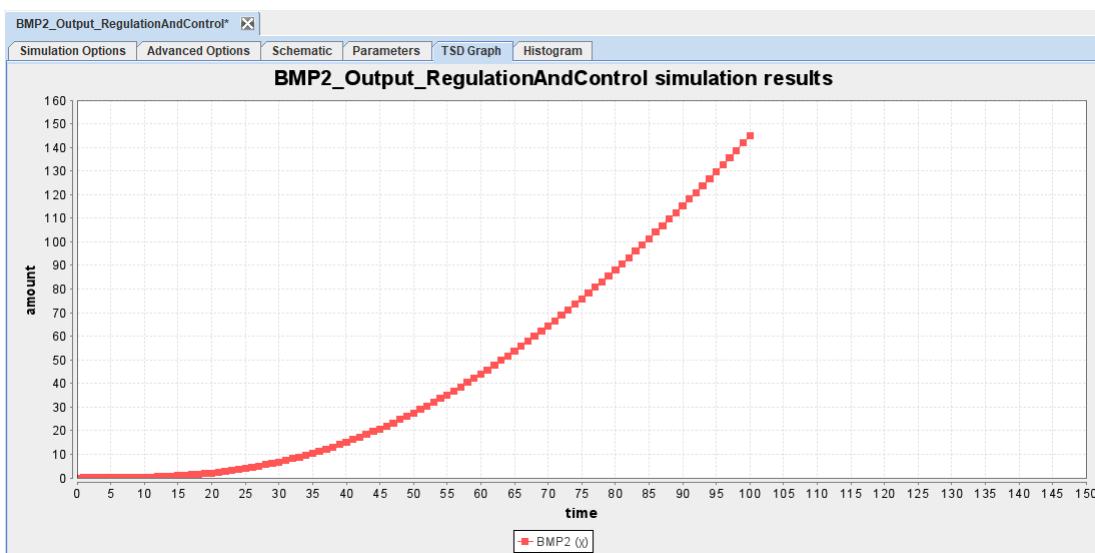


Figure 2: This graph shows the amount of BMP2 release in the regulation and control module over time. In this process, two separate Biobricks control the amount and time of release respectively.

Using Machine Learning to Infer RNA Velocity Fields

Taos Transue*
taos.transue@utah.edu
University of Utah
Salt Lake City, Utah

Payton J Thomas*
p5thomas@ucsd.edu
UC San Diego
La Jolla, California

1 INTRODUCTION

Recent advances in RNA sequencing technologies allow researchers to measure the transcriptional state of single cells (scRNA-seq) [8] and how that state is changing (RNA velocity) [4]. RNA velocity boasts a distinct advantage over more traditional methods of single-cell RNA trajectory reconstruction such as SCUBA, SCENT, and scEpath [3, 5, 9] because it reflects the underlying molecular dynamics of the cell. However, the current RNA velocity paradigm has some weaknesses.

Legacy scRNA-seq datasets without velocity data are abundant but not useful for velocity-based analysis. These datasets cannot be transformed to velocity datasets because velocity calculation algorithms like scVelo [2] require that pre-mRNA and mRNA both be measured when the dataset is created. This means that legacy datasets represent a sunk cost to experimentalists, who may not have the time or resources to collect new data. Furthermore, even modern scRNA-seq datasets with velocity tend to lack sequencing information in some regions of transcriptional state-space, resulting in limited information regarding the general RNA velocity field.

Therefore, it is prudent to develop numerical methods for (1) inferring RNA velocity information from legacy datasets and (2) inferring full RNA velocity fields from sparse modern datasets. One existing method uses machine learning to train a model to infer RNA velocities for the specific (modern) scRNA-seq dataset it was trained on [10], but it fails to address (1) and is difficult to apply for (2). To remedy these shortcomings, we used supervised learning to develop and train a machine learning model for inferring the RNA velocity of modern *and* legacy scRNA-seq datasets.

2 METHODS

Model Architecture

The model must be able to generalize to scRNA-seq datasets it was not trained on. To generalize well, it should not associate RNA velocity with absolute position in transcriptional space or pseudotime [7]. This is due to the symmetries of the RNA velocity field: the field is *translation invariant* and *rotation equivariant* [6]. The model should also be compatible with legacy datasets. Therefore, it should not utilize any RNA velocity data given in a dataset.

*Both authors contributed equally to this research.

Conceptually, our model resembles a researcher relying on their knowledge of RNA velocity fields and overall pseudotime trends to sketch velocity vectors on a scatterplot of scRNA-seq measurements. An scRNA-seq measurement is a tuple

$$r_n = (t_n, \vec{x}_n, \vec{v}_n), \quad (1 \leq n \leq N, t_n \in [0, 1])$$

where t_n is pseudotime, $\vec{x}_n \in \mathbb{R}^d$ is the transcriptional state of a cell (projected to lower dimension d [1]), and $\vec{v}_n \in \mathbb{R}^d$ is the RNA velocity at \vec{x}_n . Our model $M : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ produces an approximation $M(t, \vec{x})$ of \vec{v} using only the differences

$$\begin{cases} t - t_1, \dots, t - t_N \\ \vec{x}_1 - \vec{x}, \dots, \vec{x}_N - \vec{x} \end{cases}.$$

We choose M to have the form

$$M(t, \vec{x}) = w_{-n}\hat{\Delta}_{-n} + \dots + w_{-1}\hat{\Delta}_{-1} + w_1\hat{\Delta}_1 + \dots + w_n\hat{\Delta}_n$$

where $\hat{\Delta}_m = \frac{\vec{x}_m - \vec{x}}{\|\vec{x}_m - \vec{x}\|}$ and $\{x_{-n}, \dots, x_{-1}, x_1, x_n\}$ are the n -neighbors of \vec{x} . The set of n -neighbors has at most n datapoints with pseudotime greater than t , and at most n datapoints with pseudotime less than t . Note that if t is near the boundary of $[0, 1]$, the cardinality of the n -neighbors set may be less than $2n$. We let $w_m = f(t_m - t, \|\vec{x}_m - \vec{x}\|)$ where f is a feedforward neural network. The f used in our experiments is shown in Figure 1. Our loss function is the mean squared error (MSE) between the true and inferred velocity vectors:

$$\mathcal{L} = \frac{1}{B} \sum_{m=1}^B \|M(t_m, \vec{x}_m) - \vec{v}_m\|^2$$

where $B = 2 \leq N$ is the batch size. We train our model using PyTorch's AdamW optimizer with default parameters.

Datasets

The RNA velocity field is representable by a vector field from a system of first-order ordinary differential equations (ODEs) valid for time $t \in [0, 1]$. Each datapoint in an scRNA-seq dataset is a sample from a trajectory, and no two datapoints come from the same trajectory. As a proof-of-concept of our model, we construct three two-dimensional ODE systems that each represent an RNA velocity field motif. With each system, we generate an scRNA-seq measurement by sampling $t_n \sim \text{Unif}([0, t_{\max}])$ to produce

$$r_n = (t_n/t_{\max}, [x(t_n) \quad y(t_n)]^\top, [\dot{x}(t_n) \quad \dot{y}(t_n)]^\top)$$

where $[x(t) \ y(t)]^\top$ is an ODE solution. Random noise is added to the solution's initial condition for each measurement to reflect that no two cells measured in scRNA-seq are from the same trajectory. Sample datasets are presented here and displayed in Figure 2.

Directional Flow. Our first RNA velocity field motif is a simple directional flow. Here, flow is modeled by the ODE system

$$\begin{cases} \dot{x} = 1 \\ \dot{y} = \frac{x}{5}(y - 1) \\ x(0) = 0, \quad y(0) \sim \text{Unif}([0, \varepsilon]) \end{cases} \quad (0 \leq t \leq 3)$$

Bifurcation. Our second RNA velocity field motif is a bifurcation. Here, bifurcation is modeled by the ODE system

$$\begin{cases} \dot{x} = 1 \\ \dot{y} = \frac{x}{5}(y - 1) \\ x(0) = 0, \quad y(0) \sim 1 \pm \text{Unif}([0.5\varepsilon, \varepsilon]) \end{cases} \quad (0 \leq t \leq 6)$$

Oscillator. Our third RNA velocity field motif is an oscillation. Here, oscillation is modeled by the simple harmonic oscillator with angular frequency $\omega = 1$:

$$\begin{cases} \dot{x} = y \\ \dot{y} = -x \\ x(0) \sim \text{Unif}([1 - \varepsilon, 1 + \varepsilon]), \quad y(0) = 0 \end{cases} \quad (0 \leq t \leq 2\pi)$$

Each of these datasets is generated to contain N datapoints, each with exactly $2n$ n -neighbors. Each dataset is split into subsets of size $0.7N$, $0.2N$, and $0.1N$ for training, validation, and testing, respectively.

3 RESULTS & DISCUSSION

We trained our model on each of our three sample datasets as outlined in Section 2 to test our model's performance on the motifs common to scRNA-seq datasets. Figures 2 and 3 show its performance when trained using a 10-neighbors set (with cardinality 20) and $N = 100$.

The model shows high loss and low variance when $n \in [6, 12]$ for all three motifs, while it exhibits low loss and higher variance otherwise (Figure 4). A small n provides limited but relevant information, leading to volatile predictions due to insufficient data for noise averaging. On the other hand, a large n enables the model to estimate velocity based on long-term differentiation behavior, but also introduces spurious correlations between distant points in pseudotime. The consistently decreased performance for intermediate n values suggests the need for distinct models for close and distant points in pseudotime. Future work could explore a model that handles these two categories separately.

In all three motifs, the model using a 4-neighbors set performed better when the sample size was smaller (Figure 5).

With a large sample size, too many neighbor points had pseudotimes approximately equal to the pseudotime of the point of interest. Variation between points with similar pseudotimes is dominated by noise, rather than the dynamics of the underlying differentiation process, so the model produces RNA velocity vectors which trend toward the mean transcriptional state at each pseudotime, rather than along trajectories. This suggests that dense datasets should be deliberately sparsified before the model is applied, although the model could be applied to several sparse subsets of the dataset which could then be recombined. Future work will investigate the relationship between optimal sample size and number of neighbors.

These results are preliminary data collected from simulated scRNA-seq data (Section 2). Future work will apply our methodology to experimentally obtained scRNA-seq datasets and will also include alternative model architectures, including feedforward neural networks with different structures and ansatz functions with optimized parameters. Importantly, our future work will also apply velocity models to biological datasets other than those that they were trained on.

REFERENCES

- [1] BECHT, E., MCINNES, L., HEALY, J., DUTERTRE, C.-A., KWOK, I. W., NG, L. G., GINHOUX, F., AND NEWELL, E. W. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology* 37, 1 (2019), 38–44.
- [2] BERGEN, V., LANGE, M., PEIDL, S., WOLF, F. A., AND THEIS, F. J. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature biotechnology* 38, 12 (2020), 1408–1414.
- [3] JIN, S., MACLEAN, A. L., PENG, T., AND NIE, Q. scopath: energy landscape-based inference of transition probabilities and cellular trajectories from single-cell transcriptomic data. *Bioinformatics* 34, 12 (2018), 2077–2086.
- [4] LA MANNO, G., SOLDATOV, R., ZEISEL, A., BRAUN, E., HOCHGERNER, H., PETUKHOV, V., LIDSCHREIBER, K., KASTRITI, M. E., LÖNNERBERG, P., FURLAN, A., ET AL. Rna velocity of single cells. *Nature* 560, 7719 (2018), 494–498.
- [5] MARCO, E., KARP, R. L., GUO, G., ROBSON, P., HART, A. H., TRIPPA, L., AND YUAN, G.-C. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences* 111, 52 (2014), E5643–E5650.
- [6] PITTS, A. M. *Nominal sets: Names and symmetry in computer science*. Cambridge University Press, 2013, p. 14.
- [7] QIU, X., MAO, Q., TANG, Y., WANG, L., CHAWLA, R., PLINER, H. A., AND TRAPNELL, C. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods* 14, 10 (2017), 979–982.
- [8] SVENSSON, V., VENTO-TORMO, R., AND TEICHMANN, S. A. Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols* 13, 4 (2018), 599–604.
- [9] TESCHENDORFF, A. E., AND ENVER, T. Single-cell entropy for accurate estimation of differentiation potency from a cell's transcriptome. *Nature communications* 8, 1 (2017), 15599.
- [10] WANG, X., AND ZHENG, J. Velo-predictor: an ensemble learning pipeline for rna velocity prediction. *BMC bioinformatics* 22 (2021), 1–14.

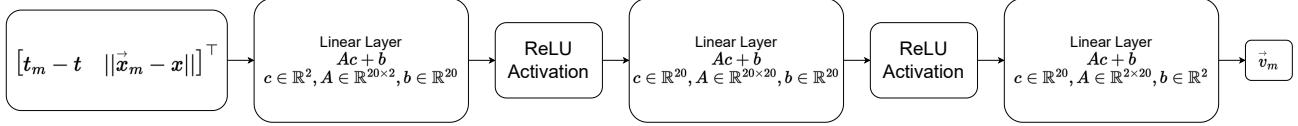


Figure 1: The feedforward neural network architecture used in our experiments. The nodes on the ends are the input and output to the model. The nodes in between are the layers of the model where inside each node is the operation the layer applies to its input. The ReLU activation function is applied element-wise.

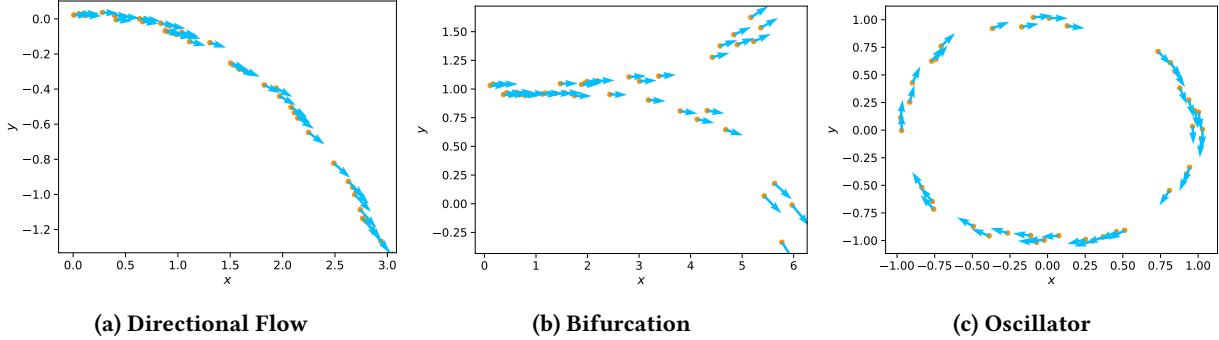


Figure 2: Sample points (orange) and velocity vectors (light blue) for our three toy models of scRNA-seq motifs. In each case, time is normalized to vary from 0 to 1 to reflect pseudotime. The random noise used is distributed uniform randomly with a maximum value of $\varepsilon = 0.05$.

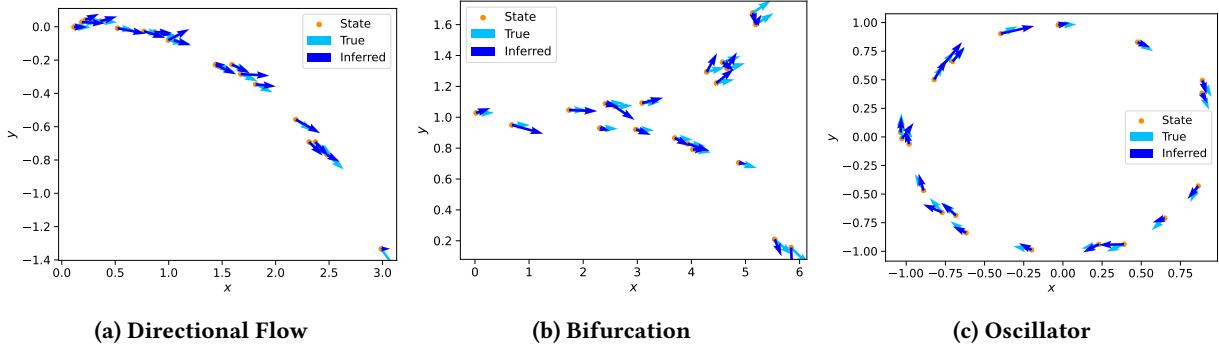


Figure 3: Sample points (orange), true velocity vectors (light blue), and inferred velocity vectors (dark blue) for our three toy models of scRNA-seq motifs. The data displayed are from the validation sets. In each case, time is normalized to vary from 0 to 1 to reflect pseudotime. The random noise used is distributed uniform randomly with a maximum value of $\varepsilon = 0.05$.

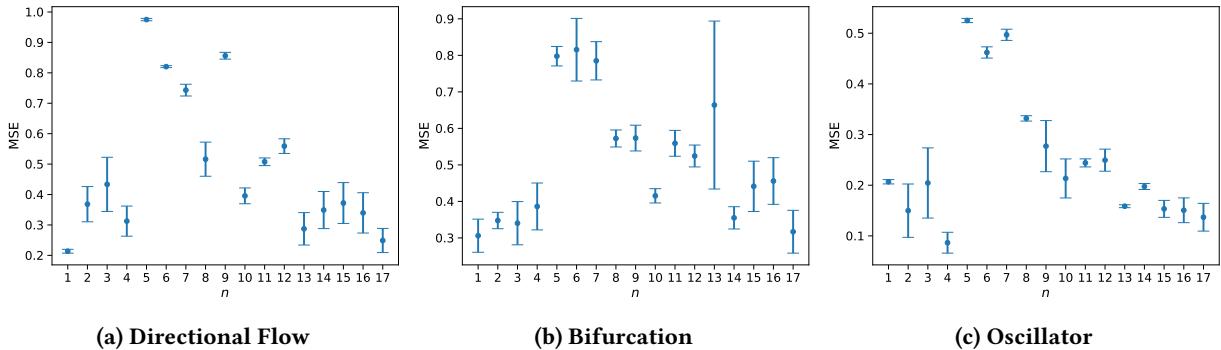


Figure 4: Mean square error (MSE) as number of neighbor points considered n varies. In each case, we observed higher variance but lower mean of MSE for n greater than 12 or less than 6 relative to $6 \leq n \leq 12$. Error bars indicate a 95% confidence interval. Note that loss was systematically lower for the oscillator motif.

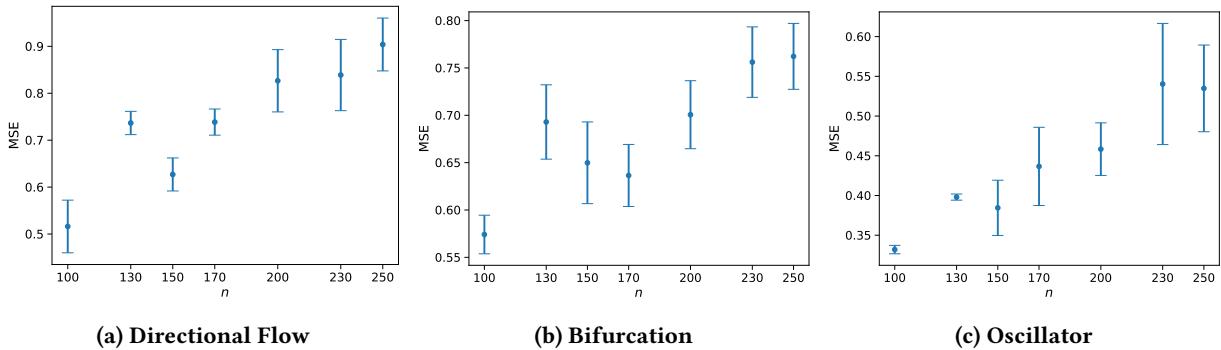


Figure 5: Mean square error (MSE) as number of total points generated (sample size) n varies. In each case, we observed MSE increasing as n increased. This indicates that the method is best suited to sparse datasets. Error bars indicate a 95% confidence interval. Note that loss was systematically lower for the oscillator motif. In all three models, we used a 4-neighbors set.

Model Checking of Interval Discrete-Time Markov Chain for Biochemical Pathways

Krishnendu Ghosh*

ghoshk@cofc.edu

College of Charleston

Charleston, South Carolina

1 INTRODUCTION

Experiments in biology are dependent on environmental conditions. Given the uncertainty in the environment, it is a challenge to create system level models. A way to incorporate uncertainty arising from an unknown environment in biology is to create a formalism using interval discrete time Markov chains (IDTMC). The transmission probabilities in an IDTMC are represented using intervals. A formalism addressing integrating uncertainty in the environment is useful in the study of behavioral properties of the biochemical pathways. The probabilities of the actions (execution of pathways) are represented in the form of transitions. The transition probabilities are recorded from different environments, A to B are represented to be within a range. For example, if the probability of execution of a pathway, X from a given set of biochemicals in environment A is p , then the same pathway, X in the identical system of pathways in an environment B may be assigned probability in the interval within $[l_p, u_p]$ where $p \in [l_p, u_p]$, $l_p, u_p \in (0, 1)$, $l_p < u_p$ and l_p, u_p are lower and upper bounds of the probability interval, respectively.

Probabilistic model checking of IDTMC is shown to be in PSPACE and NP-hard [2, 9]. The large number of species in biological models in addition to the state explosion problem is a challenge. The goal of this work to create approximations for IDTMC such that model checking are computationally feasible. In system modeling in biology, researchers construct a probability distribution from data. More than often, data is imprecise and hence, IDTMC is a better way to model data. The contribution of the work:

- (1) Create a tractable model checking formalism from IDTMC.
- (2) Evaluate computational feasibility of the formalism on published biological pathway.

2 BACKGROUND

We give a state-based definition of discrete-time Markov chain (dtmc). The other representation of dtmc and Markov

decision process (mdp) is in the form of sequences of random variables [1]. The dtmc is the foundational structure for probabilistic model checking.

Definition 2.1. (Discrete-Time Markov Chains [1]) A discrete-time Markov chain (DTMC) is a tuple: $\mathcal{M} = \langle S, S_0, \iota_{init}, \mathbf{P}, L \rangle$ where:

- S is a finite set of states.
- S_0 is the set of initial states.
- $\mathbf{P} : S \times S \rightarrow [0, 1]$, where \mathbf{P} represents the probability matrix and $\sum_{s, s' \in S} \mathbf{P}(s, s') = 1$.
- $\iota_{init} : S \rightarrow [0, 1]$ where $\sum_{s \in S} \iota_{init}(s) = 1$ is the initial distribution.
- $L : S \rightarrow 2^{AP}$, with AP the set of atomic propositions.

Definition 2.2. (Interval Discrete Time Markov Chain [2]) is a tuple $\mathcal{M}_I = \langle S, \iota_{init}, M_l, M_u \rangle$ where

- S is a finite set of states
- $M_l, M_u : S \times S \rightarrow [0, 1]$ with $M_l \leq M_u$, M_l and M_u are transition matrices.
- $\iota_{init} : S \rightarrow [0, 1]$ where $\sum_{s \in S} \iota_{init}(s) = 1$ is the initial distribution.

The intervals in the IDTMC can be closed, open and mixture of open/closed such as $(lb, ub]$ where lb, ub are the lower and upper bounds of an interval, respectively. In this work, closed interval are considered because in modeling of biological pathways, the modeler have imprecise information of the probabilities and is represented in a range. The biological experiments are dependent on physical conditions and variation in data is a reason for imprecision. Often, exact values for the lower and upper bounds are not known. Therefore, it is assumed the value of a probability lies within an interval.

3 MODEL ABSTRACTION

The proposed work is to create an abstraction of the IDTMC that is tractable for model checking. There are infinite probability distributions because of the transition probabilities are in the intervals in IDTMC. A way to create tractable model checking on IDTMC is to create an approximate distributions for the IDTMC. The approximation of the IDTMC is to construct a set of models which are idtmc with intervals of

*This research is funded by NSF CCF 2227898.

smaller sizes and then, construct set of dtmc for each idtmc from the set by sampling. The formalization is stated.

Definition 3.1. (Probabilistic Partial Model) A probabilistic partial model, \mathcal{M}_p is an idtmc where the probability intervals are subintervals for the given idtmc such there exists a probability distribution in the subintervals.

The existence of probability distribution fulfills the approximation from IDTMC to a dtmc. For each probabilistic partial model, \mathcal{M}_p , a dtmc, \mathcal{M}_s is constructed by sampling a distribution from the intervals. The set containing dtmc is denoted by \mathfrak{M} and is called *probabilistic partial model set*. Model checking is performed on the set of probabilistic partial models.

Definition 3.2. (Probabilistic Partial Model Set Checking) Given a set of probabilistic partial models and a temporal logic formula ϕ , probabilistic model set checking is the process of deciding whether ϕ is true in each \mathcal{M}_s and $\mathcal{M}_s \in \mathfrak{M}$.

The representation of IDTMC for modeling a set of biological pathways is the following. The state labels of the IDTMC are the concentration of the biochemicals, the probabilities on the transitions of the IDTMC are intervals and a label on the transition represents the pathway. The probability intervals forming a IDTMC are pruned to create *feasible* intervals. The intervals are feasible if there is a probability distribution. In the abstraction, a probability distribution by sampling using values within the bounds of the probability intervals in IDTMC. The probability distributions are constructed on the paths of the complete structure of the IDTMC representing the system of biochemical pathways. The error estimate for the sampling-based abstraction of each, \mathcal{M}_s is measured by the Kullback-Leibler divergence between each model. A model of IDTMC representing an abstraction of galactose utilization pathway in yeast [6] was simulated [3]. The model construction was based on the formalization of gene regulation such that transitions in the finite state machine represented a *regulator-regulatee* relationship defined on given a set of genes: *regulatee* is regulated by another set of genes, *regulators*. The regulatee change their expression levels during regulation but regulators do not change their expression levels. For a given set of genes, \mathcal{G} , a set of labels, \mathcal{E} representing expression levels, and a labeling function, L where $L : G \subseteq \mathcal{G} \rightarrow \mathcal{E}$. L_G denotes a set of genes, G with a label of L .

Definition 3.3. (Regulation [4]) A regulation, $\mathfrak{R} = \langle \hat{L}_G, \check{L}_G, \hat{L}_{G'} \rangle$ such that:

- (1) (Labels of expression are not same) $\hat{L}_G \cap \check{L}_G = \emptyset$.
- (2) (Regulator and Regulatee are different) $G \cap G' = \emptyset$ and $G \neq \emptyset$.
- (3) (Minimal size of regulator) There is no set, $\hat{L}_{G'}$ such that there is a regulation, $\mathfrak{R}' = \langle \hat{L}_G, \check{L}_G, \hat{L}_{G'} \rangle$.

The labels, \hat{L}_G, \check{L}_G represent labels of genes, G *regulatee* before and after the regulation by the *regulator*, $\hat{L}_{G'}$. The *regulation* is used iteratively to construct the paths for IDTMC model. The construction of feasible intervals of the IDTMC is performed by discretization of the intervals. Reachability queries using PCTL logic are posed on the model to evaluate computational feasibility on a prototype of galactose pathway in yeast. In the current work, partial models are constructed and probabilistic partial model set checking is performed to evaluate computational feasibility of the model on queries represented by PCTL.

4 EVALUATION

The evaluation of the theoretical approach, *probabilistic partial model set checking* is being elucidated by conducting experiments on published for Galactose Pathway in Yeast [6] and ERK signaling biological pathways [5, 7]. The focus is the validation of the theoretical constructs with the published data in terms of accuracy and efficiency. The abstractions are created and fed into the model checkers, PRISM [8], for probabilistic model checking using PCTL. Model checking on large problem sizes are be evaluated for computational feasibility which becomes critical given the increase of the size of probabilistic partial model set.

REFERENCES

- [1] BAIER, C., KATOEN, J.-P., AND LARSEN, K. G. *Principles of model checking*. MIT press, 2008.
- [2] BENEDIKT, M., LENHARDT, R., AND WORRELL, J. Ltl model checking of interval markov chains. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (2013), Springer, pp. 32–46.
- [3] GHOSH, K. Reasoning on stochastic models in systems biology under uncertainty. In *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)* (2020), IEEE, pp. 369–375.
- [4] GHOSH, K., AND SCHLIPF, J. Querying on model abstractions of gene regulation from noisy data. *5th International Conference on Bioinformatics and Computational Biology 2013, BICoB 2013* (01 2013), 107–112.
- [5] HEATH, J., KWIATKOWSKA, M., NORMAN, G., PARKER, D., AND TYMCHYSHYN, O. Probabilistic model checking of complex biological pathways. *Theoretical Computer Science* 391, 3 (2008), 239–257.
- [6] IDEKER, T., THORSSON, V., RANISH, J. A., CHRISTMAS, R., BUHLER, J., ENG, J. K., BUMGARNER, R., GOODLETT, D. R., AEBERSOLD, R., AND HOOD, L. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science* 292, 5518 (2001), 929–934.
- [7] KWANG-HYUN, C., SUNG-YOUNG, S., HYUN-WOO, K., WOLKENHAUER, O., MCFERRAN, B., AND KOLCH, W. Mathematical modeling of the influence of rkip on the erk signaling pathway. In *International Conference on Computational Methods in Systems Biology* (2003), Springer, pp. 127–141.
- [8] KWIATKOWSKA, M., NORMAN, G., AND PARKER, D. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)* (2011), G. Gopalakrishnan and S. Qadeer, Eds., vol. 6806 of *LNCS*, Springer, pp. 585–591.
- [9] SEN, K., VISWANATHAN, M., AND AGHA, G. Model-checking markov chains in the presence of uncertainties. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (2006), Springer, pp. 394–410.