10th International Workshop on Bio-Design Automation

Berkeley, California

July 31th - August 3rd, 2018

Foreword

Welcome to IWBDA 2018!

The IWBDA 2018 Executive Committee welcomes you to Berkeley, CA, for the Tenth International Workshop on Bio-Design Automation (IWBDA). IWBDA brings together researchers from the synthetic biology, systems biology, and design automation communities. The focus is on concepts, methodologies, and software tools for the computational analysis and synthesis of biological systems.

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

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

This year, the program consists of 17 contributed talks and 14 poster presentations. Talks are organized into 5 sessions: Design Automation, Machine-Learning, Standards, Applications, and Modeling. In addition, we are very pleased to have two distinguished invited speakers: Dr. Hector Garcia Martin from Berkeley National Lab, and Prof. Hana El-Samad from UCSF.

We thank all the participants for contributing to IWBDA; we thank the Program Committee for reviewing abstracts; and we thank everyone on the Executive Committee for their time and dedication. Finally, we thank MINRES Technologies, Teselagen, DSM, Twist Biosciences, Amyris, BBN Technologies, Digibio, and Cytoscape for their support. We also thank the Brower Center for hosting IWBDA 2018.

The following participants were provided financial support by our sponsors to attend IWBDA 2018

Nick Emery	Boston University (USA)
Timothy Jones	Boston University (USA)
Ali Lashkaripour	Boston University (USA)
Curtis Madsen	Boston University (USA)
Luis Ortiz	Boston University (USA)
Radhakrishna Sanka	Boston University (USA)
Prashant Vaidyanathan	Boston University (USA)
Adil Ali Khan	Habib University, Karachi (Pakistan)
Muhammad Abdullah Siddiqui	Habib University, Karachi (Pakistan)
Evan Appleton	Harvard (USA)
Tristan Daifuku	Harvard (USA)
Michael Moret	EPFL (France), Harvard (USA)
Mona Katharina Tonn	Imperial College London (UK)
Pauline Trébulle	Micalis Institute (France)
Nicholas DeLateur	MIT (USA)
Elisabeth Yaneske	Teesside University (UK)
Guido Zampieri	Teesside University (UK)
Gregory Fonseca	University of California, San Diego (USA)
Jenhan Tao	University of California, San Diego (USA)

IWBDA 2018 Sponsors

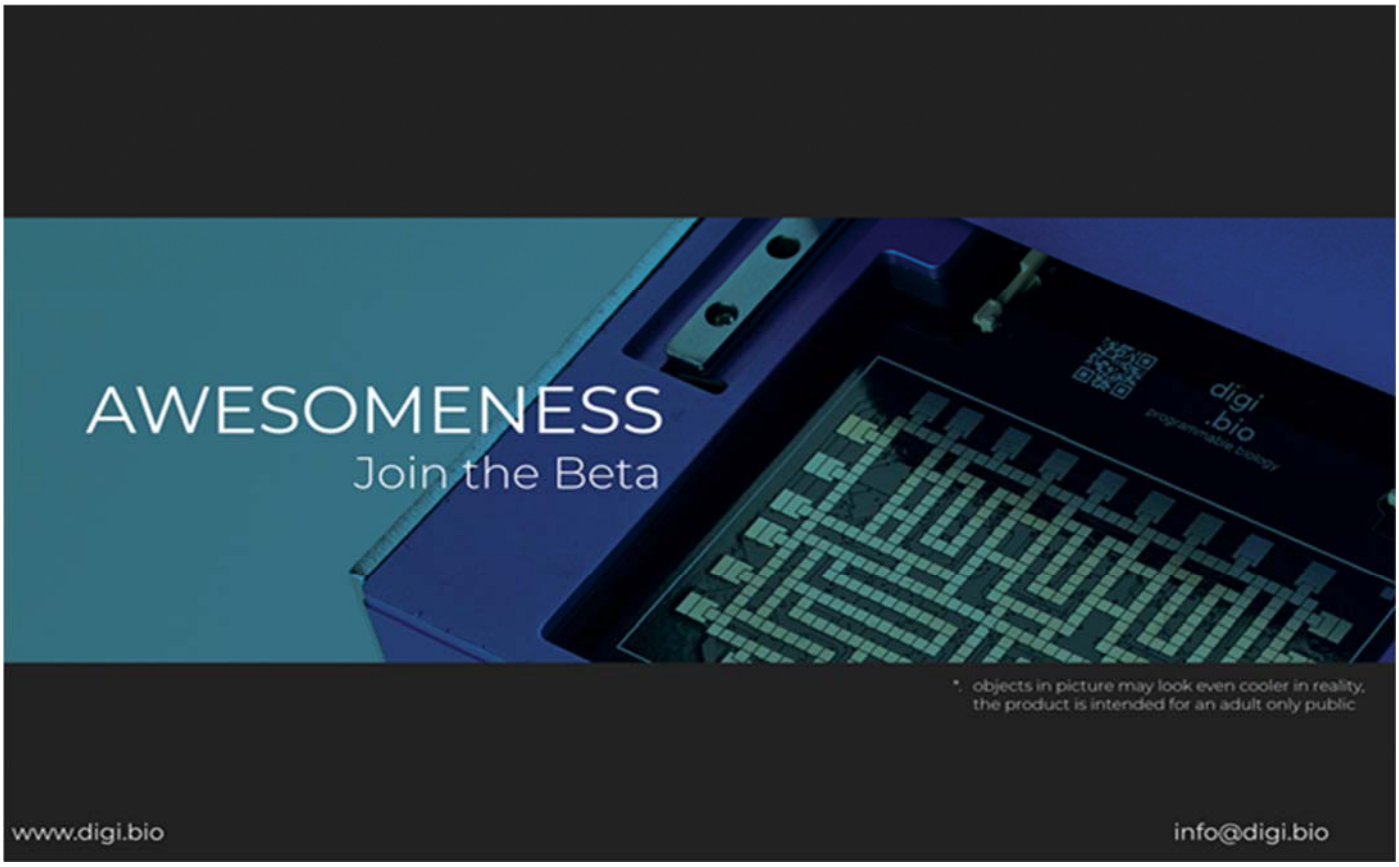
Class

Algorithm + Table

Table

Algorithm + Registration

Algorithm



MENDEL® SBML EDITOR

A graphical editor for models described in **S**ystems **B**iology **M**arkup **L**anguage (SBML).

The SBML Editor is a fast, lightweight, stand-alone graphical SBML editor, helping you to graphically capture and represent your mechanistic bio-chemical models. It is platform- independent and free for personal use, allowing you to first build your SBML model and then save it as an image or SBML file. To simulate your model, just import the SBML file into an SBML-compatible simulation software (e.g. COPASI, PET).

For more information, please visit https://www.sbml-editor.org/

©2018 MINRES® Technologies GmbH. All rights reserved.

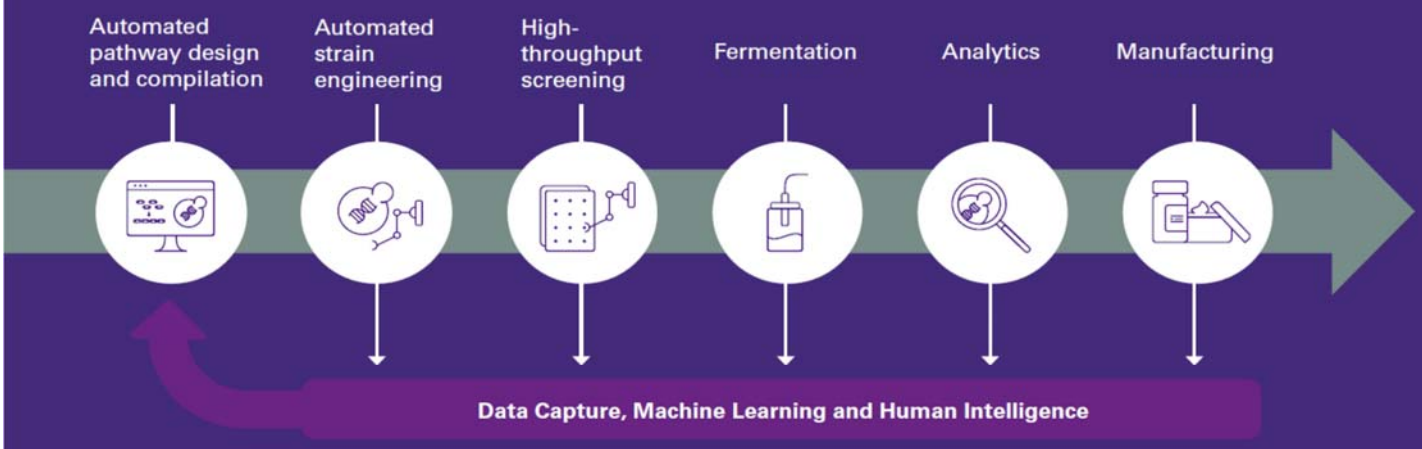
teselagen
BIOTECHNOLOGY

Mind to Molecule™

amyris

Make good.
No compromise.™

Full scale, automatic, metabolic engineering



Follow us

CHALLENGE CONVENTION BE PART OF THE NEXT FIRST.

Raytheon BBN Technologies has been providing advanced technology research and development for more than six decades. From the ARPANET and the first email, to GenBank and the first digital stereo mammography system, through the first network protected by quantum cryptography and our lifesaving Boomerang gunshot detection technology, BBN has consistently transitioned advanced research into innovative, practical solutions. Today, BBN scientists and engineers continue to take risks and challenge convention to create new and fundamentally better solutions.

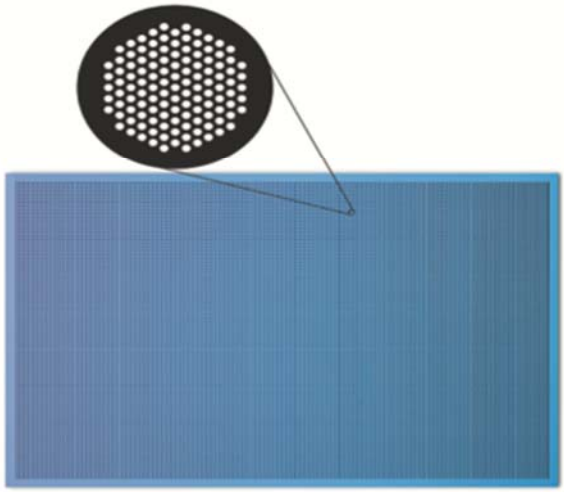
https://jobs.raytheon.com/search-jobs/BBN

Raytheon
BBN Technologies

Think Big, Screen Once

Accuracy and uniformity of oligo synthesis is critical when creating a CRISPR screening library. Twist Bioscience's technology enables massively parallel production of diverse high quality and accuracy oligo pools for generation of CRISPR gRNA libraries, allowing specific targeting and efficient screening.

- Highly Accurate Synthesis For Specific Targeting
- Highly Uniform Synthesis Ensures Guide Representation
- Massively Parallel Guide Synthesis For Efficient Screening



TWIST'S SILICON-BASED PLATE WITH THOUSANDS OF CLUSTERS, EACH WITH 121 UNIQUE OLIGO SEQUENCES.

What can Twist do for you?
sales@twistbioscience.com

TWIST BIOSCIENCE

Are societal challenges driving science?

Or is science driving societal change?

Sometimes the big issues inspire bright ideas, sometimes the big idea already exists and we can adapt it for an entirely new purpose. Whatever the catalyst, we're making bright science happen thanks to a rare and colorful mosaic of diverse competences, connections and collaborations - both inside and outside of DSM.

We're able to tackle some of the very complex problems faced by the world and continue to find the answers to what we don't yet know.

HEALTH · NUTRITION · MATERIALS

DSM
BRIGHT SCIENCE. BRIGHTER LIVING.

Organizing Committee

Executive Committee

General Chair
Nathan Hillson, Berkeley Labs

Program Committee Chair
Ernst Oberortner, Berkeley Labs

Publication Chair
Garima Goyal, Berkeley Labs

Web Chair
Aaron Adler, BBN Technologies

Finance Chair
Traci Haddock-Angelli, iGEM Foundation

Bio-Design Automation Consortium

President
Douglas Densmore, Boston University

Vice-President
Aaron Adler, BBN Technologies

Treasurer
Traci Haddock-Angelli, iGEM Foundation

Clerk
Natasa Miskov-Zivanov, Carnegie Mellon University

Program

Tuesday, July 31st

BDAthlon (http://www.iwbdaconf.org/2018/#bdathlon)

Wednesday, August 1st

SBOL Workshop (http://sbolstandard.org/iwbda-2018/)

Thursday, August 2nd

08:30 - 08:40 **Welcome & Opening Remarks, Nathan Hillson (Berkeley Labs)**

08:40 - 10:00 **Session I: Design Automation, Chair: Prashant Vaidyanathan (BU)**

- A Software Tool for Designing Trans-Differentiation Experiments with Combinations of Transcription Factors
  Evan Appleton, Jenhan Tao, Alex Ng, Christopher Glass, George Church
- A Combined Hierarchical-Combinatorial Design Editor for Large Scale DNA Library Construction
  James Craft, Michael Matena, Ximena Morales, Rodrigo Pavez, Adam Thomas, Nathan Hillson, Michael Fero
- An integrated BUILD system for DNA construction
  Taoh Green, Chris Lamkin, Tiffany Dai, Sam Denicola, Laurel Estes, George McArthur, Ximena Morales, William Moskal, Rodrigo Pavez, Thomas Rich, Adam Thomas, Michael Fero
- Optimal gene circuits for dynamic metabolic engineering
  Irene Otero-Muras, Ahmad Mannan, Julio Banga and Diego Oyarzún

10:00 - 10:30 **Break**

10:30 - 11:30 **Keynote I: Héctor García Martín (Berkeley Labs)**

Towards a predictive synthetic biology enabled by machine learning and automation

Biology has been transformed in the second half of the 20th century from a descriptive to a design science. We can engineer cells faster than ever, enabled by exponentially growing DNA synthesis and revolutionary tools like CRISPR-enabled gene editing. However, while we can make the DNA changes we intend, the end result on cell behavior is usually unpredictable. In this talk, I will explain our efforts to create predictive algorithms that take -omics data and produce actionable items for bioengineering biofuel-producing cells. I will show how machine learning and mechanistic models, enabled by automation capabilities such as microfluidics, can produce predictions accurate enough to drive synthetic biology efforts.

11:30 - 12:00 **Poster Pitches I, Chair: Ernst Oberortner (Berkeley Labs)**

- Coordinating standards: digitalization of the Standard European Vector Architecture with the Synthetic Biology Open Language
  Bryan Bartley, James McLaughlin, Goksel Misirli, Victor de Lorenzo, Anil Wipat and Angel Goni-Moreno
- Damp Lab North: Using Formal Representations of Protocols for Specify-Design-Build-Test Cycle in a Prototypical Software-Driven Laboratory
  Nicholas Emery, Marilene Pavan and Douglas Densmore
- Automating Functional Enzyme Screening & Characterization
  Luis Ortiz, Ali Lashkaripour and Douglas Densmore
- Specifying Combinatorial Designs with the Synthetic Biology Open Language
  Nicholas Roehner, Bryan Bartley, Jacob Beal, James McLaughlin, Matthew Pocock, Michael Zhang, Zach Zundel, Chris Myers and Anil Wipat
- The Desktop Biofoundry: Biodesign Manufacturing Automation in a Cloud-driven Digital Microfluidics Platform with Integrated Temperature Control, Optical Sensing and Purification
  Federico Muffatto, Sabrina Zaini and Frido Emans
- Automating synthetic biology using microfludics
  Steve Shih
- Toward Programming 3D Shape Formation in Mammalian Cells
  Jesse Tordoff, Jacob Beal, Ron Weiss, Bryan Bartley, Gizem Gumuskaya, Katherine Kiwimagi, Matej Krajnc, Kevin Lebo, Stanislav Shvartsman, Allen Tseng and Nicholas Walczak
- Software Projects of the Edinburgh Genome Foundry
  Valentin Zulkower, Isaac Luo, Aitor Bleda and Hille Tekotte

12:00 - 14:30 **Lunch & Poster Session I**

14:30 - 15:30 **Session II: Machine-Learning, Chair: Curtis Madsen (BU)**

- A Reverse Predictive Model Towards Design Automation of Microfluidic Droplet *Generators Ali Lashkaripour, Christopher Rodriguez, Douglas Densmore*
- A Machine Learning Environment for Synthetic Biology Rodrigo Pavez, Felipe Loyola, *Andres Perez, Cesar Pinto, Andres Ramirez, Pablo Vera, Michael Fero, Eduardo Abeliuk*
- Identifying composition rules for transcription factor circuits that control macrophage signal response with deep learning
  Jenhan Tao, Gregory Fonseca, Christopher Glass

15:30 - 16:00 **Break**

16:00 - 16:40 **Session III: Standards, Chair: Cornelia Scheitz (Autodesk)**

- The Synthetic Biology Open Language Supports Integration of the Engineering Life-Cycle for Synthetic Biologists

Bryan Bartley, Christian Atallah, Alasdair Humphries, Vishwesh Kulkarni, Curtis Madsen, Goksel Misirli, Angel Goni-Moreno, Tramy Nguyen, Ernst Oberortner, Nicholas Roehner, Meher Samineni, Zach Zundel, Jacob Beal, Chris Myers, Herbert Sauro, Anil Wipat

- Standardizing Design Performance Comparison in Microfluidic Manufacturing
  Radhakrishna Sanka, Brian Crites, Joshua Lippai, Jeffrey McDaniels, Phillip Brisk, Douglas Densmore

16:40 - 17:00 **Daily Wrap-up and Announcements, Nathan Hillson (Berkeley Labs)**

Social Event

Friday, August 3rd

09:00 - 09:10 **Opening Remarks (Nathan Hillson)**

09:10 - 10:30 **Session IV: Applications, Chair: Jenhan Tao (UCSD)**

- Integrated computational extraction of cross-cancer poly-omic signatures
  Guido Zampieri, Claudio Angione
- Towards Computer-Aided Synthetic Developmental Biology
  Evan Appleton, Michael Moret, Tristan Daifuku, George Church
- Automated design of gene circuits with optimal mushroom-bifurcation behaviour
  Rubén Pérez-Carrasco, Irene Otero-Muras, Julio Banga, Chris Barnes
- Mechanistic effects of influenza in bronchial cells through poly-omic genome-scale modelling
  Elisabeth Yaneske, Claudio Angione

10:30 - 10:50 **Break**

10:50 - 11:00 **Allan Kuchinsky IWBDA Scholarship**

11:00 - 12:00 **Discussion Session, Moderator: Douglas Densmore (BU)**

Bio-Design Automation Design Metrics: What is useful? What is not?

12:00 - 12:30 **Poster Pitches II, Chair: Ernst Oberortner (Berkeley Labs)**

- Context-aware predictive tools for portable genetic circuit engineering
  Pablo Carbonell, Sandra Taylor, Rehana Sung, Adrian J Jervis, Rainer Breitling, Jean-Loup Faulon and Nigel S Scrutton.

- Asynchronous Genetic Circuit Design Automation with Cloud-based Component Libraries
  Timothy Jones, Tramy Nguyen, Zach Zundel, Chris Myers and Douglas Densmore.
- Tracking the provenance of synthetic biological system construction at the DOE Joint Genome Institute (JGI)
  Xianwei Meng, Ernst Oberortner, Nathan Hillson and Samuel Deutsch.

- Open Vector Editor - DNA Viewing and Annotation
  Thomas Rich, Tiffany Dai, Sam Denicola, Ximena Morales, Nathan Hillson and Michael Fero.
- GeneTech 2.0: Improved Genetic Circuit Synthesis and Technology Mapping
  Muhammad Abdullah Siddiqui, Adil Ali Khan, Hasan Baig and Jan Madsen.
- CoRegCAD: a framework from regulatory network to metabolic engineering
  Pauline Trébulle, Jean-Marc Nicaud and Mohamed Elati.

12:30 - 14:00 **Lunch & Poster Session II**

14:00 - 15:00 **Keynote II: Hana El-Samad (UCSF)**

Biological control: The versatile ways in which cells use feedback loops

In 1939, Walter Cannon wrote in his book The wisdom of the Body: “The living being is an agency of such sort that each disturbing influence induces by itself the calling forth of compensatory activity to neutralize or repair the disturbance”. Since this remarkable statement that postulates the use of feedback control to support life, we have come to appreciate that the use of feedback loops is ubiquitous at every level of biological organization, from the gene to the ecosystem. In this talk, we introduce a technology to systematically pinpoint and study feedback operation in endogenous biological systems. We also discuss how building synthetic feedback control with modular architecture and predictable operation would be immensely enabling for biotechnology, and present some ideas on how this might be achieved.

15:00 - 15:30 **Break**

15:30 - 16:50 **Session V: Modeling, Chair: Nicholas Roehner (BBN)**

- Temporal Verification of Genetic Circuits
  Curtis Madsen, Prashant Vaidyanathan, Nicholas Delateur, Evan Appleton, Greg Frasco, Calin Belta, Ron Weiss and Douglas Densmore
- An Automated BioModel Selection System (BMSS) for Gene Circuit Design
  Chueh Loo Poh, Jingwui Yeoh and Kai Boon Ivan Ng
- Spatiotemporal principles of genetic circuit design
  Ruud Stoof, Alexander Wood, James McLaughlin, Anil Wipat and Angel Goni-Moreno
- BLiSS: Black-List Sequence Screening
  Lisa Simirenko, Jan-Fang Cheng, Samuel Deutsch, Nathan J. Hillson

16:50 - 17:00 **Closing Remarks, Nathan Hillson (Berkeley Labs)**

Keynote Presentation

Hana El-Samad

Biological control: The versatile ways in which cells use feedback loops

Prof. Hana El-Samad is a Professor at University of California, San Francisco. Her laboratory is focused on tackling fundamental challenges in the precise measurement and analysis of decision making in complex biological networks. They are an exciting mixture of biologists, engineers, physicists and mathematicians who are constantly pushing technological, experimental, and computational boundaries. For example, they use variability between genetically and environmentally identical cells, coupled with quantitative measurement of single cells and rigorous computational modeling, to unravel cellular wiring diagrams and probe their dynamical properties. They develop technologies to monitor dynamic signal propagation across whole networks in single cells with unprecedented precision and scale. They also develop efficient computational tools for the modeling and analysis of spatio-temporal cellular dynamics. They harness the power of these tools to probe coordination and signal integration in cellular stress responses, such as the Environmental Stress Response.

In 1939, Walter Cannon wrote in his book The wisdom of the Body: “The living being is an agency of such sort that each disturbing influence induces by itself the calling forth of compensatory activity to neutralize or repair the disturbance”. Since this remarkable statement that postulates the use of feedback control to support life, we have come to appreciate that the use of feedback loops is ubiquitous at every level of biological organization, from the gene to the ecosystem. In this talk, we introduce a technology to systematically pinpoint and study feedback operation in endogenous biological systems. We also discuss how building synthetic feedback control with modular architecture and predictable operation would be immensely enabling for biotechnology, and present some ideas on how this might be achieved.

Keynote Presentation

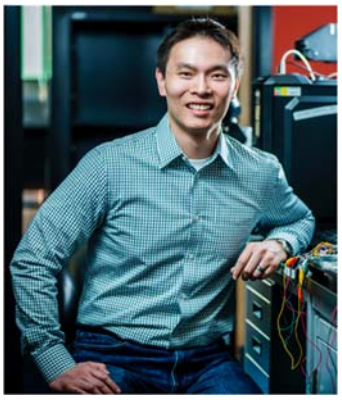
Héctor García Martín

Towards a predictive synthetic biology enabled by machine learning and automation

Dr. Héctor García Martin is a Scientific Lead at Berkeley Lab, participating in both Agile BioFoundry and JBEI projects. His research interests involve mathematical modeling of biological systems, metabolic engineering, systems biology, metabolic flux analysis, data visualization, scientific software development and microbial ecology. In his position at the Joint BioEnergy Institute (JBEI), Dr. Martin is developing predictive quantitative models of microbial metabolism to direct metabolic engineering efforts and improve biofuel yields. His efforts are divided among machine learning and flux modeling approaches, as well as software development for visualization and acquisition of data. He is also using mathematical approaches to try and develop quantitative predictive models for microbial communities.

Biology has been transformed in the second half of the 20th century from a descriptive to a design science. We can engineer cells faster than ever, enabled by exponentially growing DNA synthesis and revolutionary tools like CRISPR-enabled gene editing. However, while we can make the DNA changes we intend, the end result on cell behavior is usually unpredictable. In this talk, I will explain our efforts to create predictive algorithms that take -omics data and produce actionable items for bioengineering biofuel-producing cells. I will show how machine learning and mechanistic models, enabled by automation capabilities such as microfluidics, can produce predictions accurate enough to drive synthetic biology efforts.

Allan Kuchinsky Scholarship



Steve Shih

Dr. Steve Shih completed his B.A.Sc. in Electrical Engineering from the University of Toronto and then went to University of Ottawa to complete his Master's degree in Chemistry. He then returned to Toronto to complete his Ph.D. in Biomedical Engineering with Prof. Aaron Wheeler specializing in microfluidic technologies. His contribution during his Ph.D. include developing automated high-throughput digital microfluidic methods for cell-based assays, point-of-care diagnostics, and biofuel-related applications. After learning microfluidics, he then spent 3 years as a postdoctoral research at the Joint BioEnergy Institute and UC Berkeley working closely with Dr. Nathan Hillson, Dr. Jay Keasling, and Dr. Anup Singh. His main focus was to automate DNA assembly and transformation processes using digital microfluidic platforms. He was also designing novel microfluidic techniques for screening active enzymes used for biofuel producing microbes and integrating it to mass spectrometry. As of January 2016, he is an Assistant Professor at Concordia University in the Department of Electrical and Computer Engineering with a cross-appointment in the Department of Biology. He is also a member of the only Center for Applied Synthetic Biology in Canada. His current interests are to automate processes related to synthetic biology and to improve microfluidic fabrication techniques such that it is translatable to the general public.

The fourth annual Allan Kuchinsky scholarship is generously sponsored by Agilent and Cytoscape.

Previous recipients

2017 Dr. Curtis Madsen

2016 Dr. Nicholas Roehner

2015 Dr. Swapnil Bhatia

Agilent Technologies

Cytoscape

Abstracts – Table of Contents

Oral Presentations

Poster Presentations

A software tool for designing trans-differentiation experiments with combinations of transcription factors

Evan Appleton[1,2,*], Jenhan Tao[3,4], Greg Fonseca[3,4], Alex Ng[1,2], Parastoo Khoshakhlagh[1,2], Christopher Glass[4] and George Church[1,2]

[1]Department of Genetics, Harvard Medical School, Boston, MA
[2]Wyss Institute for Biologically Inspired Design, Boston, MA
[3]Bioinformatics, University of California, San Diego, La Jolla, CA
[4]Department of Medicine, University of California, San Diego, La Jolla, CA
{gfonseca,cglass}@ucsd.edu,{evan_appleton}@hms.harvard.edu,
{gchurch}@genetic.med.harvard.edu, {alexhmng,jenhantao,parastoo.kh}@gmail.com

1. INTRODUCTION

Most canonical synthetic biology projects involve the design, construction, and testing of genetic circuits to perform some function in biological systems. While a recent large focus has been placed on genetic circuits in E. coli, there are other contemporary areas of biology that can use synthetic biology ideas and approaches to solve problems. One of these newer areas is the field of direct cell-type conversions, in particular with human induced pluripotent stem cells (iPSCs). These cells are derived from healthy mature human tissue and are considered functionally equivalent to embryonic stem cells in the sense that they are thought to be able to differentiate into any other cell or tissue type in the human body. This can be accomplished in a variety of ways including surface-condition based methods, media growth factor methods and genetic methods. Many recent successes rely primarily on the use of genetic methods - specifically directed differentiation via transcription factor (TF) over-expression[2]. This last method is one that can benefit from synthetic biology, and especially, bio-design automation (BDA). A key current challenge in the stem cell field is how to use genetics to differentiate iPSCs into other cells types that can be useful for diagnostics or therapies.

When using TF over-expression based methods, a majority of the work comes down to identifying the subset of TFs in the genome that can successfully convert cell type 'A' to cell type 'B' when overexpressed. Historically this identification process has been rooted in expert knowledge from groups that had extensive expertise in the genetics of one or more cell type. While this approach has resulted in some significant successes, these methods are generally low-throughput and many valuable conversions are still unknown. Furthermore, recent work in our group has resulted in the definition and production of a 'TFome' of all known TFs in the human genome composed of 1768 genes - thus exhaustively screening all possible combinations of TFs of even a relatively small size is essentially impossible with current methods.

Some recent efforts[1, 3] have instead attempted to approach this problem computationally using transcriptomics data to identify TF combinations that are likely candidates to perform a specific conversion, although these methods have a few key limitations - they exclusively rely on sometimes sparse transcriptomics data, they output one final solution as opposed to an experimental design, they do not leverage other types of next-generation sequencing data, and they have no intrinsic feedback loop for evaluating outcomes to automatically design the next round of experiments.

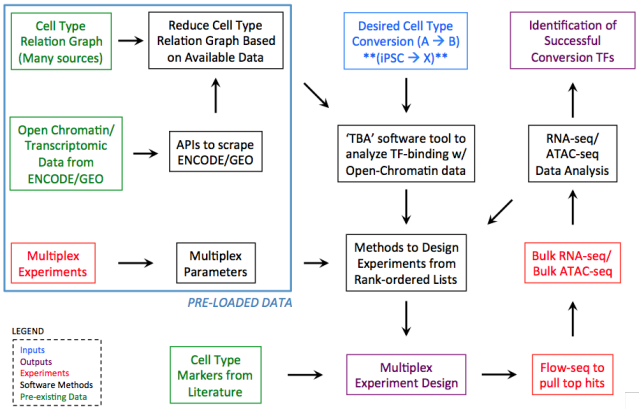


Figure 1: Computational workflow overview. Raw data for predictive analysis is pulled from online sources and markers from literature. Based upon the desired conversion, TBA analysis is run to determine likely lineage-determining factors and then this list is refined by RNA-seq analysis and an experiment can be designed using basic probability

Here we present a software tool that uses open-chromatin data and transcriptomics data to output an experimental design in which a subset of the human TFome is multiplexed to identify cells that seem to be most successful in that screen. The most promising candidates from this screen are then sequenced for RNA expression and open chromatin and the data is used to inform the next multiplex experiment. This tool is first being applied to known TF-based conversions, but is being developed as generally as possible for use as a tool for any desired novel conversion.

BIPOLAR NEURON				DERMIS BLOOD VESSEL ENDOTHELIAL CELL				T-CELL			
GENE	AVG_SIG	AVG_COEF	RPKM	GENE	AVG_SIG	AVG_COEF	RPKM	GENE	AVG_SIG	AVG_COEF	RPKM
SP4	8.17E-123	0.10050333	5.918	JUN	3.17E-280	0.11524513	40.393	RUNX3	4.79E-182	0.12827977	25.618
RFX3	5.13E-73	0.06693288	7.929	JUNB	3.17E-280	0.11524513	28.576	RUNX2	4.79E-182	0.12827977	6.615
RFX5	5.13E-73	0.06693288	20.762	JUND	3.17E-280	0.11524513	44.376	ETS1	2.46E-105	0.10779038	185.06
MXI1	5.13E-73	0.06693288	23.89	FOSL2	3.17E-280	0.11524513	30.511	ETV3	2.46E-105	0.10779038	6.174
BHLHE23	1.51E-62	0.08577534	8.955	NFIA	1.44E-202	0.07090262	10.079	FLI1	2.46E-105	0.10779038	23.391
NEUROG2	1.51E-62	0.08577534	19.792	JUN	6.65E-149	0.07738755	40.393	ELK3	2.46E-105	0.10779038	11.585
OLIG1	1.51E-62	0.08577534	10.903	JUND	6.65E-149	0.07738755	44.376	ELK4	2.46E-105	0.10779038	18.75
OLIG2	1.51E-62	0.08577534	18.801	JUN	5.50E-100	0.06364023	40.393	SP4	1.88E-72	0.0838446	13.181
OLIG3	1.51E-62	0.08577534	53.007	ELF2	1.72E-85	0.04853449	12.502	REST	2.72E-55	0.07782289	17.066
ONECUT1	3.85E-57	0.12887682	4.975	RFX3	1.71E-56	0.04484158	4.122	KLF9	5.13E-50	0.06722925	11.221
ONECUT2	3.85E-57	0.12887682	33.352	RFX5	1.71E-56	0.04484158	17.292	NRF1	8.63E-49	0.0488664	4.622
HES4	6.06E-39	0.11034983	95.105	REST	9.64E-56	0.06037212	22.357	REL	1.97E-40	0.04619416	4.231
ZBTB1	3.29E-33	0.05384625	10.132	HES4	3.38E-52	0.10680297	12.569	RELA	1.97E-40	0.04619416	15.732
NFYB	1.09E-30	0.06963041	6.14	SOX13	2.49E-51	0.06717378	5.488	ZBTB1	8.92E-33	0.06023633	20.731
JUN	1.31E-25	0.05024185	31.973	ZBTB33	1.06E-50	0.06534091	8.592	RFX5	2.40E-28	0.04508091	14.652
JUND	1.31E-25	0.05024185	47.205	TEAD1	1.30E-46	0.03724069	8.012	MXI1	2.40E-28	0.04508091	6.819

Figure 2: Preliminary results for TF combination candidates to convert iPSCs into bipolar neurons, dermis blood vessel endothelial cells, and T-cells. Genes are shown in rank order according to a significance score from analysis of open-chromatin data (DNAse-seq). RNA-sequencing analysis yields a results for gene quantity in units of Reads Per Kilobase of transcript, per Million mapped reads (RPKM)

2. TBA

TBA takes all of the open chromatin sites within a cell type as input. Next, a set of GC-matched background sequences equal in size to the number of open chromatin sites. For each of the open chromatin sites and background sequences TBA calculates the best match to hundreds of DNA binding motifs drawn from the CISBP JASPAR library. The quality of a sequence's match to a motif is quantified as a motif score (aka log likelihood ratio score). A TBA model scores the probability of observing open chromatin at a sequence by computing a weighted sum over all the motif scores computed for that sequence. The weight for each motif is learned by iteratively modifying the weights (initialized from random values) until the model's predictive performance no longer improves. By examining the weight of each motif, we can assess whether the presence of each motif is positively, negatively or not correlated with the binding of a transcription factor. The significance of each motif can be quantified using an *in silico* mutagenesis approach in which the performance of a trained TBA model is compared to a perturbed model which has one motif removed. Given the prominent role lineage determining factors play in establishing open chromatin sites within each cell type, we would expect the motifs assigned a high rank by TBA to correspond to lineage determining factors.

3. RNA-SEQ ANALYSIS

Given a rank ordered list of TFs that are likely active in a given cell type, we further prune this list with RNA-sequencing data. Since many TFs have the same or very similar binding motifs (and therefore the same TBA score), this step significantly prunes the list of candidates and gives extra confidence in the TBA analysis. This analysis is done with a standard pipeline for RNA-seq analysis - raw FASTQ data files are trimmed and aligned to the human genome, HOMER is used to call peaks from alignment regions, which then uses annotations to determine which genes are expressed at which quantities (in RPKM).

4. EXPERIMENTAL DESIGN

Once a rank-ordered list of TFs for a given conversion are produced, we must determine how many TFs to consider in a given experiment. This is currently calculated based upon the number of cells that are transfected, the efficiency of the reaction, and how many TFs can be integrated into the genome in one reaction. In our standard experimental set-up, we can reliably integrate at least 20 genes, we transfect 1M cells in one reaction, and have a >10% efficiency. Since the number of possible combinations of TF integrations is approximated by 2^n, we currently perform screens with the top 16 TFs for each cell type. We perform the transfection and then plate into a 10cm dish. We allow the cells to recover for two days, after which we select and then proliferate for two weeks. Then, using a surface marker from literature, we separate the cells with a positive marker reading using FACS and perform RNA-seq on that population. Based upon these results, we can modify the list of conversion factors from the original prediction.

5. EXPERIMENTAL VALIDATION

We are currently validating our predictions for converting TFs for two cell types we can already reliably differentiate with published factors (bipolar neurons and endothelial cells) and one new conversion (T-Cells) of potentially high value. Based upon the outcomes from this validation, we will modify the prediction results by either adding considerations for network biology, alternative splice forms, or expression.

6. REFERENCES

[1] P. Cahan, H. Li, S. A. Morris, E. L. Da Rocha, G. Q. Daley, and J. J. Collins. Cellnet: network biology applied to stem cell engineering. *Cell*, 158(4):903–915, 2014.

[2] T. Graf and T. Enver. Forcing cells to change lineages. *Nature*, 462(7273):587, 2009.

[3] O. J. Rackham, J. Firas, H. Fang, M. E. Oates, M. L. Holmes, A. S. Knaupp, H. Suzuki, C. M. Nefzger, C. O. Daub, J. W. Shin, et al. A predictive computational framework for direct reprogramming between human cell types. *Nature genetics*, 48(3):331, 2016.

A Combined Hierarchical-Combinatorial Design Editor for Large Scale DNA Library Construction

KYLE CRAFT, TeselaGen Biotechnology, Inc, kcraft@teselagen.com
MICHAEL MATENA, TeselaGen Biotechnology, Inc, mmatena@teselagen.com
XIMENA MORALES, TeselaGen Biotechnology, Inc., ximena@teselagen.com
RODRIGO PAVEZ, TeselaGen Biotechnology, Inc., rpavez@teselagen.com
ADAM THOMAS, TeselaGen Biotechnology, Inc., adam@teselagen.com
NATHAN HILLSON PHD, TeselaGen Biotechnology, Inc, njhillson@teselagen.com
MICHAEL FERO PHD, TeselaGen Biotechnology, Inc., mike.fero@teselagen.com

ABSTRACT

TeselaGen's DESIGN software provides a unified interface and compute infrastructure for the design of a hierarchical and combinatorial system of DNA assembly reactions, and the generation of instructions for how to build those DNA assemblies. The platform provides a standardized system for tracking the relationships between design elements (parts and annotation), designs and assembly protocols. Among features supporting the design process is the ability to create Design Templates that can be reused across designs. A number of assembly reaction types are supported including Type IIs Endonuclease (Golden Gate, MoClo, etc.), Flanking Homology (Gibson, InFusion, etc.). The system optimizes assembly reactions to take advantage of a variety of DNA sourcing options.

CCS CONCEPTS

• **Information systems** → **Information systems applications**; *Enterprise information systems*; Enterprise applications

• **Information systems** → **Information systems applications**; *Decision support systems*; Expert systems

KEYWORDS

Synthetic Biology, DNA Design, Recombinant DNA, Biotechnology, Combinatorial Design, Hierarchical Design, Golden Gate, Gibson

1 INTRODUCTION

The TeselaGen Synthetic Evolution[tm] enterprise platform for synthetic biology consists of four major software modules; DESIGN, BUILD, TEST, and EVOLVE. Within the DESIGN module, the TeselaGen Hierarchical Design Editor (HDE) plays the crucial role of capturing the human designer's intent as a design data structure, subsequently passing it to the protocol design engine *j5* [1] in order to generate both human and machine readable protocols. HDE supports several important synbio workflow requirements:

- Capture of combinatorial + hierarchical designs.
- Scar-less design of large scale DNA libraries.
- Cost optimization including part re-use where warranted.
- Sourcing material from best available options.
- Design Templating to aid the design process.

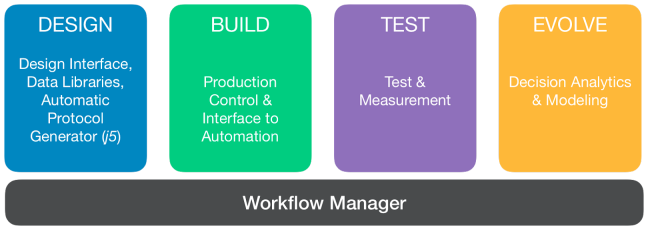


Figure 1: TeselaGen's four part enterprise system for guiding synthetic biology workflows.

2 DESIGN CAPTURE

The fundamental role of the DESIGN module is the accurate capture of the designer's intent. Earlier versions of the DESIGN module allowed for combinatorial designs, but not hierarchical designs. The new HDE editor provides both design modes in a single interface. Target designs are constructed 5' to 3' left to right by moving parts from a parts library to columns in a whiteboard style user interface. Alternatives for any given part are listed as entries within a column. The user has the option of partitioning the target design into a set of sub-designs in a hierarchical fashion. The user can also specify the naming scheme and the preferred DNA assembly chemical reactions.

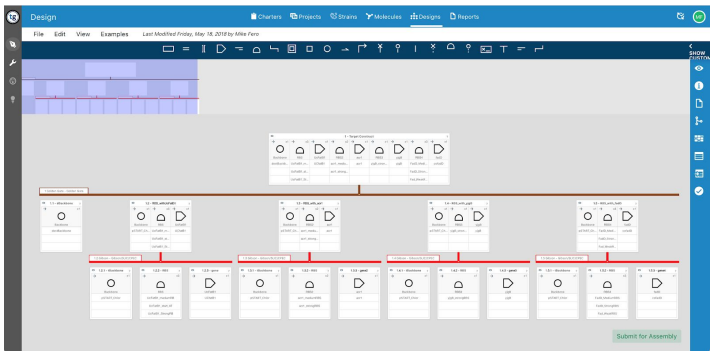


Figure 2: Hierarchical Design Editor. A two level combinatorial design specifies four Gibson assemblies that generate intermediates that will be reactants for a final Golden Gate assembly.

3 PART RE-USE and SOURCING

Part reuse is important when the size of the built library is large and cost is a constraint. Without part reuse the cost of a library can grow with the number of parts ~O(n), with part reuse ~O(log(n)). HDE utilizes the *j5* algorithm when designing a set of combinatorial assembly reactions which automatically maximizes part reuse within a combinatorial design. Hierarchical designs are also be optimized for part reuse across designs with utilities that check if intermediate stretches of DNA already exist in inventory, modifying the build instructions accordingly so that previously built constructs are not assembled again. Users can also use this availability information to automatically break down target constructs into divisions based on available subsections instead of manual divisions. Material availability is also extended to query external vendors for what they can provide. This is done through a series of API integrations with vendor utilities that check DNA segments for their manufacturability, cost, and delivery times.

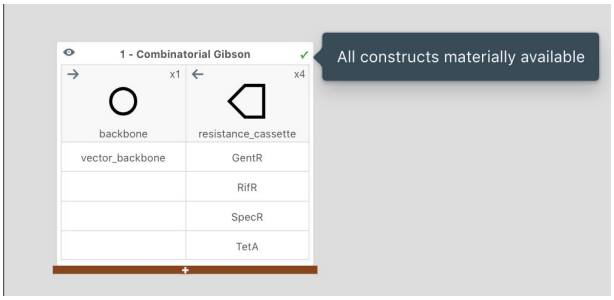


Figure 3: Material availability utilities check to see that DNA is actually available, either in inventory or from external vendors.

HDE also provides the user with greater control over the sourcing of the DNA parts used in their designs, especially with Type IIS restriction enzyme digest/ligation assemblies. When performing such an assembly, the interface automatically adds validation to the input parts to ensure that they are sourced with the appropriate flanking digest sites. Additionally, HDE provides support for custom validation through the use of Design Rules, with validation logic based on either part tags (e.g. all parts in the first column need to have the “backbone” tag) or a part’s base pairs (e.g. all parts in the cds column need to begin with “ATG”).

4 DESIGN TEMPLATING

The templating system allows users to automate building out portions of designs for complicated hierarchical workflows. Users can capture the commonalities of related designs in a template and then apply them across new designs. Any aspect of the design editor can be stored in a template for reuse, including specifications for DNA parts, overhang validation and assembly reactions. This simplifies the design process when creating multiple designs that share characteristics, providing a streamlined interface that minimizes redundant input from the user.

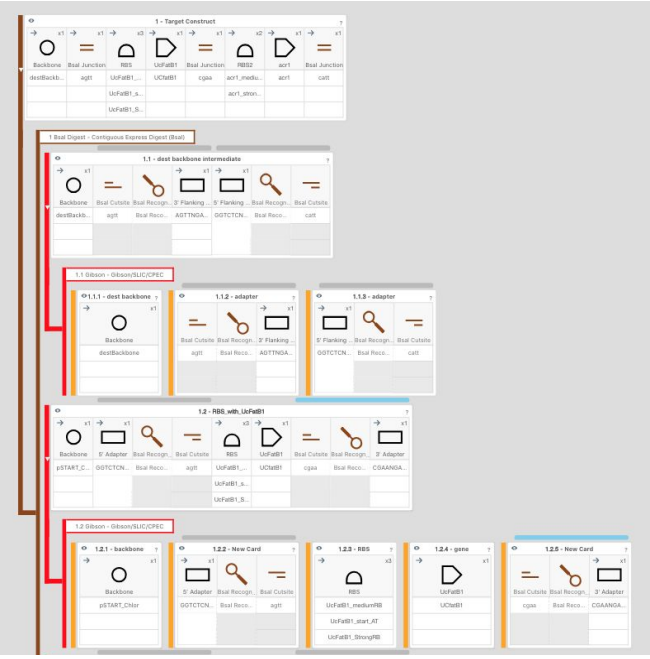


Figure 4: Design Templates allow for automating the creation of complex builds.

5 CONCLUSIONS

TeselaGen’s DESIGN module has added support for hierarchical design workflows. This enhancement adds value for two major use cases. 1) Users who choose to adopt inherently hierarchical assembly methods such as MoClo are now able to rapidly design complex builds using the editor while maximizing part reuse and minimizing re-work and cost. 2) Users who would like to build very long pieces of scarless DNA will find the streamlined interface a convenient and reproducible way to break down very long target designs into buildable submodules.

ACKNOWLEDGMENTS

This work was supported in part by NSF SBIR Phase IIB 1430986.

REFERENCES

[1] Nathan J. Hillson, Rafael D. Rosengarten, and Jay D. Keasling, 2012. j5 DNA Assembly Design Automation Software *ACS Synth. Biol.* 1 (1), pp 14–21, DOI: 10.1021/sb2000116

An integrated BUILD system for DNA construction

TAOH GREEN, TeselaGen Biotechnology, Inc., tgreen@teselagen.com
CHRIS LAMKIN, TeselaGen Biotechnology, Inc., chris.lamkin@teselagen.com
TIFFANY DAI PHD TeselaGen Biotechnology, Inc., tiffany.dai@teselagen.com
SAM DENICOLA, TeselaGen Biotechnology, Inc., sam.g.denicola@gmail.com
LAUREL ESTES, TeselaGen Biotechnology, Inc., laurel.estes@teselagen.com
GEORGE MCARTHUR Arzeda Corp. george.mcarthur@arzeda.com
XIMENA MORALES, TeselaGen Biotechnology, Inc.,, ximena@teselagen.com
WILLIAM MOSKAL, Dow Agrosciences, bill.moskal@gmail.com
RODRIGO PAVEZ, TeselaGen Biotechnology, Inc., rpavez@teselagen.com
THOMAS RICH, TeselaGen Biotechnology, Inc., tnrich@teselagen.com
ADAM THOMAS, TeselaGen Biotechnology, Inc., adam@teselagen.com
MICHAEL FERO PHD, TeselaGen Biotechnology, Inc., mike.fero@teselagen.com

ABSTRACT

TeselaGen's BUILD software provides an interface and compute infrastructure for translating computed DNA Library design information into actionable instructions for automation. In addition, the software provides many of the functions of a traditional LIMS system, allowing users to chain common laboratory actions into workflows that can be tracked and monitored. Other traditional features, such as inventory and bio sample management are also provided.

CCS CONCEPTS

• **Information systems** → **Information systems applications**; *Enterprise information systems*; Enterprise applications

• **Information systems** → **Information systems applications**; *Decision support systems*; Expert systems

KEYWORDS

Synthetic Biology, DNA Design, Recombinant DNA, Biotechnology, Combinatorial Design, Hierarchical Design, Golden Gate, Gibson

1 INTRODUCTION

The TeselaGen Synthetic Evolution[tm] enterprise platform for synthetic biology consists of four major software modules; DESIGN, BUILD, TEST, and EVOLVE.

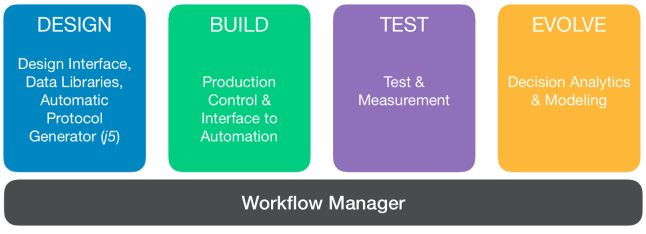


Figure 1: TeselaGen's four part enterprise system for guiding synthetic biology workflows.

The BUILD module plays the crucial role of capturing the computed design from the DESIGN module and turning into actionable instructions for subsequent automation. BUILD supports several important synbio workflow requirements:

- Design to build information translation
- Complex workflow design, execution, and monitoring
- Biological material management and sourcing
- Interfacing with suppliers of reagents and services

2 DESIGN to BUILD

The fundamental role of the BUILD module is the seamless transformation of computed design information and automatically generated protocols into actionable instructions for automation (and/or laboratory workers). When first developed, the DESIGN platform took on the task of generating worklists, the essential data needed by automation to execute liquid handling tasks. As the DESIGN platform developed, and as users continued to add functional requirements, it has become clear that a new BUILD module should take over the practical tasks of associating DNA fragment data with physical locations, plates, wells, volumes and concentrations, etc. The BUILD module has many of the same functional features as a traditional LIMS (Laboratory Information Management System) but is unique in its ability to translate computed designs to worklists.

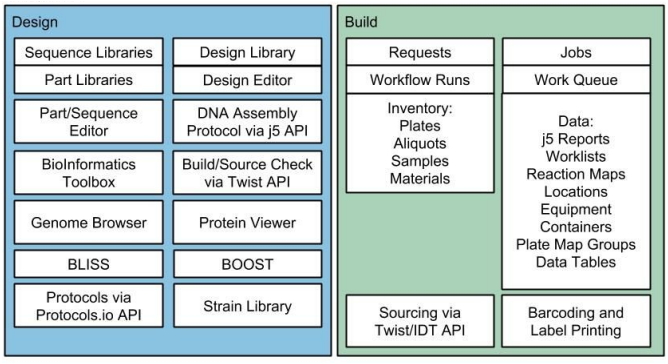


Figure 2: DESIGN and BUILD components

3 WORKFLOW DESIGN and EXECUTION

Experience has shown that, although the fundamental sciences is common to all workflows, users can be very

heterogeneous in their workflow goals and approaches. Rather than locking the BUILD platform into a particular approach or methodology, we have chosen to make the platform modular, following a dataflow execution model. Currently, acyclic directed workflow graphs are accommodated. A number of pre-programmed macros have been built including:

1. PCR Prep and Execution
2. Assembly Rxn Prep and Execution
3. Clonal Transformation
4. DNA Ordering
5. Plate Registration, Barcoding, Reformatting, Rehydration, Replication, Consolidation, Combination
6. Colony Plating
7. Reaction Mapping

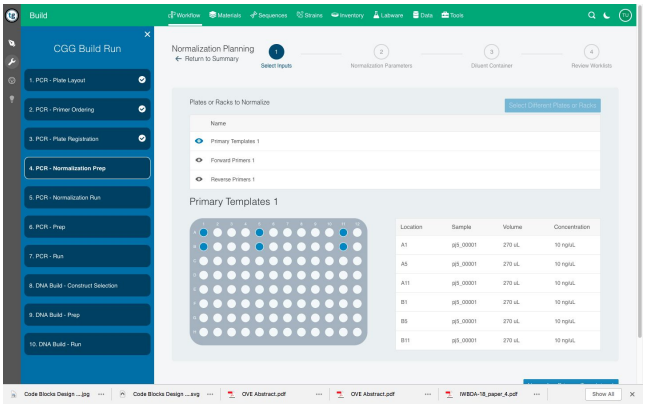


Figure 3: The Build module uses a macro style workflow execution paradigm. Distinct macro blocks are listed from top to bottom at left, their associated data and internal logic at right.

4 BIO MATERIAL MANAGEMENT

We have extended the idea of strain management to accommodate bio materials in general; DNA, Oligos, Strains, Proteins, Enzymes and Reagents and other configurable domain specific bio material types. In the case of a microbial material, the notion of a strain has been enhanced to include information about the individual plasmids the strain may have been modified to carry, information about physical aliquotes such as location, volume and concentration. Physical strains resulting from a synthetic biology workflow have built in provenance for the parent strain and the DNA design used to generate them. The system is also accommodates inventory management tasks such as recording and displaying locations and contents for samples contained in plates/wells, tubes and trays as well as laboratory equipment.

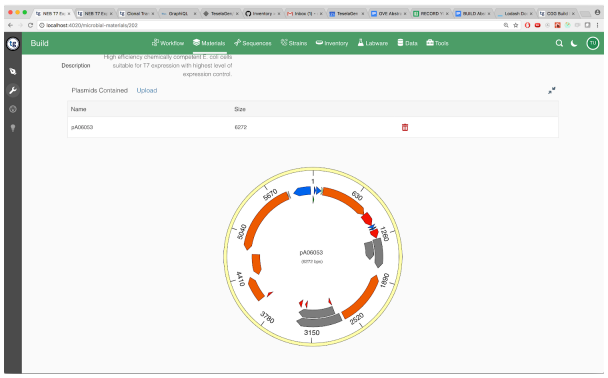


Figure 4: Microbial material entries are linked to physical aliquotes as well as their canonical strain information.

5 INTERFACING with SUPPLIERS

Automatic protocol generation raises the possibility of optimizing protocols against just-in-time availability of reagents and services. The DESIGN module performs this function, but actual fulfillment tasks are handed off to the BUILD module. In the BUILD module, actual orders to our early partners such as Twist and IDT are possible.

5 CONCLUSIONS

Experience with synthetic biology workflows has shown that creating seamless handoffs between the design of complex combinatorial/hierarchical libraries and their “in-practice” construction speeds research and benefits reproducibility. With our BUILD module we have started from the premise that protocols are generated automatically and are (for the most part) handed off from a DESIGN tool. This preserves the continuity of the DESIGN/BUILD connection and helps preserve provenencance and process tracking, as well as enforcing discipline about design of experiments and replicates. Together these attributes ensure better success at subsequent TEST and EVOLVE stages of a typical R&D workflow.

ACKNOWLEDGMENTS

This work was supported in part by NSF SBIR Phase IIB 1430986.

REFERENCES

[1] Nathan J. Hillson, Rafael D. Rosengarten, and Jay D. Keasling, 2012. j5 DNA Assembly Design Automation Software *ACS Synth. Biol.* 1 (1), pp 14–21, DOI: 10.1021/sb2000116

Optimal gene circuits for dynamic metabolic engineering*

Extended Abstract

Irene Otero-Muras
BioProcess Engineering Group
Spanish National Research Council (CSIC)
Vigo, Spain
ireneotero@iim.csic.es

Ahmad A. Mannan
Department of Mathematics
Imperial College London
London SW72AZ, UK
a.mannan@imperial.ac.uk

Julio R. Banga
BioProcess Engineering Group
Spanish National Research Council (CSIC)
Vigo, Spain
julio@iim.csic.es

Diego A. Oyarzún
Department of Mathematics
Imperial College London
London SW72AZ, UK
d.oyarzun@imperial.ac.uk

ABSTRACT

Metabolic engineering has led to the production of a wealth of chemicals with engineered microbes. A traditional strategy is to express foreign enzymes in a cellular host to convert metabolic intermediates into target products. Thanks to recent progress in gene circuit engineering, it is now possible to build gene circuits to dynamically control the activity of synthetic pathways. This strategy offers multiple benefits, including self-regulation of enzyme expression and adaptive pathway activity in response to changes in bioreactor conditions. Here we present a circuit design framework based on multiobjective optimization of metabolic production. We show that combinations of positive and negative feedback loops produce a range of dynamics on a Pareto-optimal front. These loops define connectivities between pathway intermediates and gene expression that achieve optimal tradeoffs between production performance and metabolic burden to the host. Our results lay the computational groundwork for the systematic, model-based, design of complex gene circuitry at the interface of synthetic biology and metabolic engineering.

KEYWORDS

Metabolic engineering; Dynamic control; Synthetic Biology; MINLP Optimization; Pareto optimality

ACM Reference Format:
Irene Otero-Muras, Ahmad A. Mannan, Julio R. Banga, and Diego A. Oyarzún. 2018. Optimal gene circuits for dynamic metabolic engineering: Extended Abstract. In *Proceedings of ACM conference (IWBDA'18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.475/123_4

*Produces the permission block, and copyright information

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
IWBDA'18, July 31–August 3, 2017, Berkeley, CA, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION

A central challenge in metabolic engineering is to determine the optimal enzyme expression levels for maximal production and reduced metabolic footprint on the host. In traditional pathway engineering, enzymes are expressed at constant levels, which leads to "open loop" pathways that lack robustness and cannot adapt to perturbations in bioreactor conditions [2, 6]. Moreover, metabolic imbalances often impair growth and limit the performance of engineered pathways. These imbalances arise from e.g. the accumulation of toxic intermediates, the depletion of key metabolites for survival, or the onset of native regulatory mechanisms that counteract pathway activity.

As a result of such limitations, the last decade has witnessed the emergence of *dynamic metabolic engineering*, a new technology that aims to embed gene regulatory systems into the design of engineered pathways. The core principle is to use synthetic gene circuits that adapt pathway expression in response to the metabolic state of the host. Such circuits can cause a pathway to self-adapt its expression levels to match production goals and dynamically allocate metabolic flux between production and growth. Recent implementations have showcased how gene circuits can improve yield in various pathways [1, 7, 8], yet so far we do not have quantitative procedures for the rational design of circuit architectures or their individual components [3, 5]. In this work we present a circuit design strategy based on multiobjective optimization of circuit parameters and architectures.

2 METHODS, RESULTS AND DISCUSSION

We consider a heterologous pathway that branches from a native pathway that is essential for growth (see inset of Fig. 1A). The dynamic model is a set of ODEs for the mass balance of metabolites and enzymes. The enzymes follow standard Michaelis-Menten kinetics (not shown), and the enzyme synthesis rates (u_i) are parameterized by:

$$u_i = \eta_i^0 k_i^c + \eta_i^{1+}\sigma_i^+(x_1) + \eta_i^{1+}\sigma_i^-(x_1) \quad (1)$$

with $\sigma_i^+(x) = a_i \frac{x^2}{\theta_i^2+x^2}$ and $\sigma_i^-(x) = a_i \frac{\theta_i^2}{\theta_i^2+x^2}$, such that a vector p_b of binary parameters $\eta_i \in \{0,1\}$ defines the architecture of the regulation, and a vector p_c of real tunable parameters contains the strengths a_i and activity thresholds θ_i of the promoters coding for the enzymes.

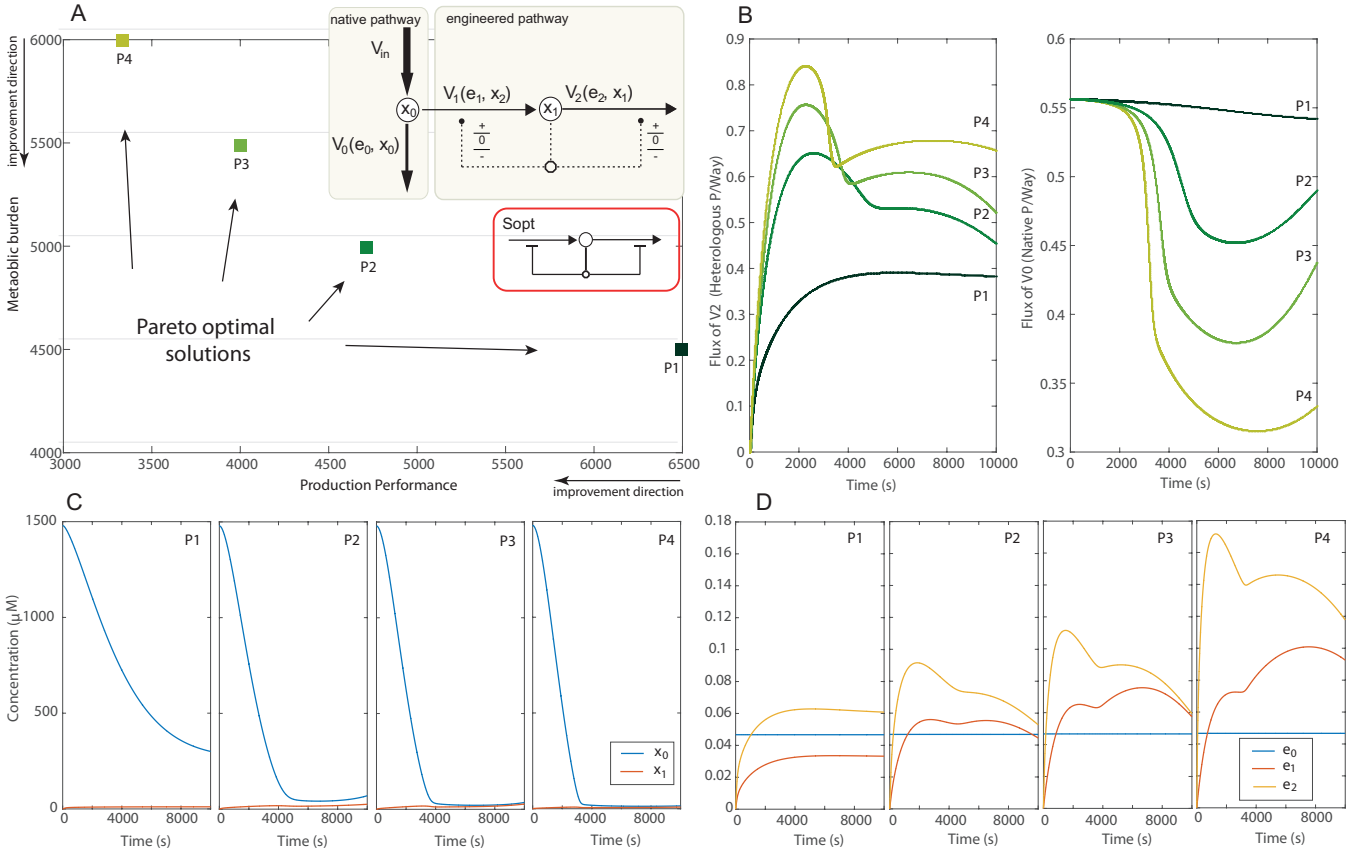


Figure 1: A) Pareto front of solutions for the multiobjective optimization problem. B) Dynamics of fluxes V2 (heterologous pathway) and V0 (native pathway) for each of the Pareto solutions C) metabolite dynamics and D) enzyme dynamics.

We formulate a biobjective optimization problem to find (among all feedback architectures and parameters encoded by the decision vector $[p_b\ p_c]$) those leading to optimal dynamic response in terms of production performance and metabolic burden. The resulting formulation is a Multiobjective Mixed Integer Nonlinear Programming Problem (MO-MINLP), that we solve combining an ε-constraint strategy with state- of-the-art MINLP solvers [4] to obtain the Pareto front of optimal solutions. Unlike exhaustive exploration, our method scales well to large search spaces. As an illustrative proof of concept we solve the multiobjective problem with one controlling metabolite (x_1), obtaining a set of Pareto optimal gene circuits. The solutions are depicted in Fig. 1A in the objective space, all of them correspond to the architecture indicated in the figure. This suggests that the combination of positive and negative feedback provides an optimal trade-off between production performance and metabolic burden to the host cell. The optimal time courses of the production flux (V_2) and growth flux (V_0) are shown in Fig. 1B, and the dynamics of the metabolites and enzymes are shown in Figs. 1C and D respectively. Note that solution $P1$ accounts for one objective only, resulting in a slow, 1st-order like response. In contrast, solutions that account for for the second objective (P2,P3 and P4) display increasingly faster and nonlinear responses as we move along the Pareto front.

Acknowledgments. AAM and DAO acknowledge support from the Human Frontier Science Program through Young Investigator Grant RGY0076-2015. IOM and JRB acknowledge funding from MINECO projects SYNBIOFACTORY (DPI2014-55276-C5-2-R) and SYNBIOCONTROL (DPI2017-82896-C2-2-R).

REFERENCES

[1] S. J. Doong, A. Gupta, and K. L. J. Prather. 2018. Layered dynamic regulation for improving metabolic pathway productivity in *Escherichia coli*. *Proceedings of the National Academy of Sciences* 115, 12 (2018), 2964–2969.

[2] D. Liu, A.A. Mannan, Y. Han, D.A. Oyarzún, and F. Zhang. 2018. Dynamic metabolic control: towards precision engineering of metabolism. *Journal of Industrial Microbiology and Biotechnology* (2018).

[3] A. A Mannan, D. Liu, F. Zhang, and D. A. Oyarzún. 2017. *ACS synthetic biology* 6(10) (2017), 1851–1859.

[4] I. Otero-Muras and J. R. Banga. 2014. Multicriteria optimization for biocircuit design. *BMC Systems Biology* 8 (2014), 113.

[5] J. T. Stevens and J. M. Carothers. 2015. Designing RNA-based genetic control systems for efficient production from engineered metabolic pathways. *ACS Synthetic Biology* 4, 2 (2015), 107–115.

[6] N. Venayak, N. Anesiadis, W. R. Cluett, and R. Mahadevan. 2015. Engineering metabolism through dynamic control. *Curr. Opin. Biotechnol.* 34 (2015), 142–152.

[7] P. Xu, L. Li, F. Zhang, G. Stephanopoulos, and M. Koffas. 2014. Improving fatty acids production by engineering dynamic pathway regulation and metabolic control. *Proceedings of the National Academy of Sciences* 111, 31 (aug 2014), 11299–11304.

[8] F. Zhang, J. M Carothers, and J. D. Keasling. 2012. Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids. *Nature Biotechnology* 30, 4 (2012), 354.

A Reverse Predictive Model Towards Design Automation of Microfluidic Droplet Generators

Ali Lashkaripour[1], Christopher Rodriguez[2] and Douglas Densmore[1,3]

[1]Department of Biomedical Engineering, Boston University, Boston, MA
[2]Department of Cyber Engineering, Louisiana Tech University, Ruston, LA
[3]Department of Electrical & Computer Engineering, Boston University, Boston, MA
lashkari@bu.edu, cwr023@latech.edu , and dougd@bu.edu

1. INTRODUCTION

Droplet-based microfluidic devices in comparison to test tubes can reduce reaction volumes 10^9 times and more due to the encapsulation of reactions in micro-scale droplets [4]. This volume reduction, alongside higher accuracy, higher sensitivity and faster reaction time made droplet microfluidics a superior platform particularly in biology, biomedical, and chemical engineering. However, a high barrier of entry prevents most of life science laboratories to exploit the advantages of microfluidics. There are two main obstacles to the widespread adoption of microfluidics, high fabrication costs, and lack of design automation tools. Recently, low-cost fabrication methods have reduced the cost of fabrication significantly [7]. Still, even with a low-cost fabrication method, due to lack of automation tools, life science research groups are still reliant on a microfluidic expert to develop any new microfluidic device [3, 5]. In this work, we report a framework to develop reverse predictive models that can accurately automate the design process of microfluidic droplet generators. This model takes prescribed performance metrics of droplet generators as the input and provides the geometry of the microfluidic device and the fluid and flow settings that result in the desired performance. We hope this automation tool makes droplet-based microfluidics more accessible, by reducing the time, cost, and knowledge needed for developing a microfluidic droplet generator that meets certain performance requirement.

2. DROPLET GENERATION

As shown in Fig. 1, by flowing an aqueous and a non-aqueous phase through a narrow opening, called orifice, microfluidic droplets are generated. The two major performance metrics of a droplet generator are droplet size and generation rate, which we call "dependent variables". These parameters are dictated by device geometry (i.e., orifice size, aspect ratio, oil width ratio, water with ratio, orifice length, and expansion ratio) and flow rates of oil and water (for a given geometry, these flow rates are determined by Capillary number and flow rate ratio), which we call "independent variables". To have a design automation tool for droplet generation, for a prescribed droplet size and generation rate, we need to provide geometry and flow conditions. Therefore, the goal is to take the dependent variables as input, and output the independent variables, that would result in the given dependent variables.

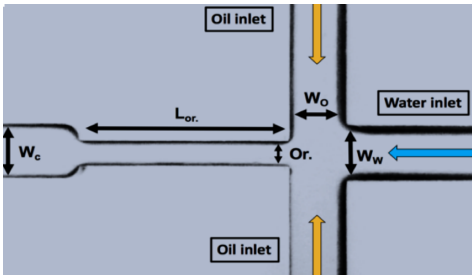


Figure 1: Droplet generation is achieved by flowing oil and water through a flow-focusing geometry. This process has eight inputs (six geometry, and two flow variables) and two outputs (droplet size and generation rate).

3. REVERSE PREDICTIVE MODELS

The first step in building a reverse predictive model is to construct a dataset of inputs-outputs over the range of expected values. in lieu of experimental data, we generated data points based on a formulaic relationship between the independent and dependent variables roughly derived from real world observations [6]. We scaled these equations to remain in reasonable ranges. These formulas are shown in Eqs. (1) & (2).

$$Generation\,rate = \frac{(OW + AR + ER + OL + WW + FR) * Ca * 5000}{Or} \quad (1)$$

$$Droplet\,size = \frac{Or * AR * ER * OL * WW}{OW * Ca * FR * 10} \quad (2)$$

where the parameters and their ranges are given in Table 1. Our models relied on min-max normalization to reduce biases in input magnitudes. For each column in our dataset (representing a type of parameter), we scaled each entry to be in the range of zero to one according to the minimum and maximum of that parameter's range.

We explored three methods for our reverse predictive models: nearest data point, M5P trees, and radial basis function (RBF) interpolation. Nearest data point is one of the simplest strategies, requiring no model to be fit to the data [1]. For any input of desired dependent variables, we simply search our data set for a point with dependent variable values that are the closest to the input. Then, we return the independent variables associated with that data point.

Table 1: The range of inputs (independent variables) used to build the input-output dataset using Eq. (1) and Eq. (2).

Symbol	Parameter	Range
OW	Normalized oil input width*	2 - 4
AR	Aspect ratio	1 - 3
OL	Normalized orifice length*	1 - 9
WW	Normalized water input width*	2 - 4
Or	Orifice width	50 - 300 μm
ER	Expansion ratio*	2 - 6
Ca	Capillary number	0.02 - 0.2
FR	Flow rate ratio	2 - 20

*The normalized values are divided by orifice width.

M5P trees are a more advanced version of linear regression where model trees branch out based on the value of the independent variables and data points that are close are put together in a same leaf. Each leaf contains an equation that represents a linear regression on the grouped data points [8]. We grow two M5P trees (one to optimize on each dependent variable) from our data set. Next, we search our training data set for a data point P which is closest to our desired input, much in the same way as nearest data point. We input the independent variable values from P into both M5P trees to obtain two linear equations Eq. (3) and Eq. (4). We require our solutions to satisfy both of these equations with no error. Therefore $f(x)$ must equal our desired droplet generation rate and $g(x)$ must equal our desired droplet size. There are an infinite number of points which we can accept, as we have only 2 constraints and 8 degrees of freedom. Therefore, we attempt to find a solution that deviates the least amount possible from our original closest data point P.

$$f(x) = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n \tag{3}$$

$$g(x) = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{4}$$

RBF interpolation is a fast way to form regression models in high dimensions. In this study, we used a multiquadric function to build RBF regression models [2]. Much in the same way as the M5P trees, we fit two models to the data, one for each dependent variable. In order to generate suggestions using RBF interpolation, a nearest data point P is found (again, using the same method as nearest data point). P is used as the starting point for our optimization algorithm. We seek to find a point S that minimizes the error for all M models against all Y desired dependent variable values (performance metrics) as shown in Eq. (5). We use a form of gradient descent called SLSQP (Sequential Least Squares Programming) as our cost-minimization function.

$$\sum_{i=0}^{N} |M_i(S) - Y_i| \tag{5}$$

4. RESULTS

We created a dataset of 2500 points for training and another dataset of 2500 for accuracy verification. These datapoints are produced using Eqs. (1) & (2), while parameter values are taken randomly from the range given in Table 1. We tested the accuracy for both single and combined optimizations. Single optimization attempts to find a perfect solution on a single performance metric. Combined optimization attempts to find the best compromise, considering both performance metrics. The error is calculated as given in Eq. (6). Where x is the desired value, $M(x)$ is the model suggestion to get that desired value, and $f(M(x))$ is the "real" value of that suggestion calculated from Eqs. (1) & (2). The results are shown in Fig. 2., a) for single and b) for combined optimization.

$$Error = \frac{|f(M(x)) - x|}{x} \tag{6}$$

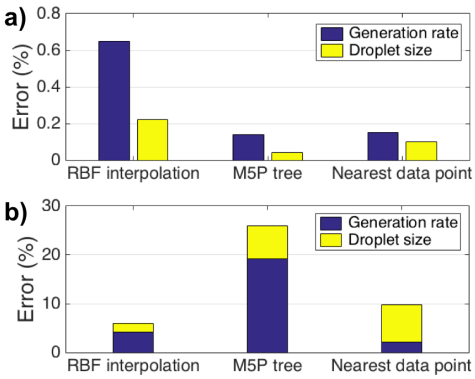


Figure 2: Accuracy comparison of different reverse predictive models. **a)** Only considering one performance metric. **b)** Considering both performance metrics, simultaneously.

5. CONCLUSION AND FUTURE WORK

We proposed a framework for design automation of microfluidic droplet generators, using a reverse predictive model. This model, takes the prescribed performance metrics as the input (droplet size and generation rate). Then, outputs geometry and flow conditions required to achieve this desired performance. The dataset of this study can be replaced by experimental data to accurately capture the real world behavior of microfluidic droplet generators.

6. REFERENCES

[1] D. M. Bates and D. G. Watts. *Nonlinear regression analysis and its applications*, volume 2. Wiley Online Library, 1988.

[2] D. S. Broomhead and D. Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[3] K. Chakrabarty and F. Su. Design automation challenges for microfluidics-based biochips. *DTIP of MEMS & MOEMS, Montreux, Switzerland*, pages 01–03, 2005.

[4] A. Lashkaripour, A. Abouei Mehrizi, M. Rasouli, and M. Goharimanesh. Numerical study of droplet generation process in a microfluidic flow focusing. *Journal of Computational Applied Mechanics*, 46(2):167–175, 2015.

[5] A. Lashkaripour, M. Goharimanesh, A. A. Mehrizi, and D. Densmore. An adaptive neural-fuzzy approach for microfluidic droplet size prediction. *Microelectronics Journal*, 78:73–80, 2018.

[6] A. Lashkaripour, A. A. Mehrizi, M. Goharimanesh, M. Rasouli, and S. R. Bazaz. Size-controlled droplet generation in a microfluidic device for rare dna amplification by optimizing its effective parameters. *Journal of Mechanics in Medicine and Biology*, 18(01):1850002, 2018.

[7] A. Lashkaripour, R. Silva, and D. Densmore. Desktop micromilled microfluidics. *Microfluidics and Nanofluidics*, 22(3):31, 2018.

[8] J. R. Quinlan et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pages 343–348. Singapore, 1992.

A Machine Learning Environment for Synthetic Biology

RODRIGO PAVEZ, TeselaGen Biotechnology, Inc., rpavez@teselagen.com
FELIPE LOYOLA, TeselaGen Biotechnology, Inc., felipe.loyola@teselagen.com
ANDRES PEREZ, TeselaGen Biotechnology, Inc., andres.perez@teselagen.com
CESAR PINTO, TeselaGen Biotechnology, Inc., cesar.pinto@teselagen.com
ANDRES RAMIREZ, TeselaGen Biotechnology, Inc., aramirez@teselagen.com
PABLO VERA, TeselaGen Biotechnology, Inc., pablo.vera@teselagen.com
MICHAEL FERO PHD, TeselaGen Biotechnology, Inc., mike.fero@teselagen.com
EDUARDO ABELIUK PHD, TeselaGen Biotechnology, Inc, eduardo@teselagen.com

ABSTRACT

TeselaGen's EVOLVE module provides an interface and compute infrastructure to create, train and execute Machine Learning algorithms using Teselagen's SDK for fast data loading and processing. The module will facilitate ML solutions created by Teselagen as well as enabling development of custom algorithms in a way that integrates seamlessly with other modules on the Teselagen Suite.

CCS CONCEPTS

• **Information systems** → **Machine Learning**; *Learning Settings*; Learning from demonstrations

KEYWORDS

Synthetic Biology, DNA Design, Machine Learning, Deep Learning

1 INTRODUCTION

The TeselaGen Synthetic Evolution[tm] enterprise platform for synthetic biology consists of four major software modules; DESIGN, BUILD, TEST, and EVOLVE. Advancing synthetic biology relies on the accurate representation of DNA designs and experimental outcomes in order to understand and improve designs to better accomplish design goals. We have developed a platform for testing methods used to improve large scale experiments to optimize biosynthetic pathways for enzyme and chemical production. The module speaks directly to our Test module, which is based on the Experiment Data Depot (EDD) knowledge base [1], developed at LBNL as a foundation for additional analytic and machine learning.

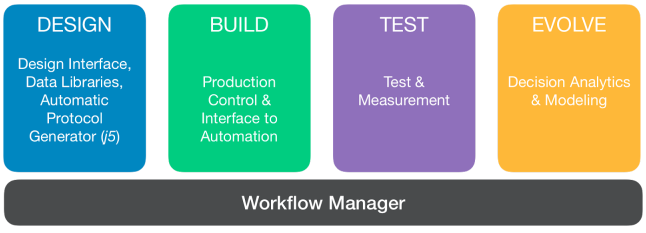


Figure 1: TeselaGen's four part enterprise system for guiding synthetic biology workflows.

2 ARCHITECTURE

The application of machine learning to genetics and synthetic biology raises a number of challenges that need to be addressed. Training machine learning models might require a large amount of data, which can be difficult to acquire. In addition, some machine learning techniques require extensive computational resources, without which training becomes too time-consuming. The EVOLVE module is being built to help biotech companies design, deploy, train and test state-of-the art machine learning algorithms that run on cloud-based hardware, optimized for running compute-intensive ML applications. The EVOLVE module contains a frontend interface, developed in React/Redux, that allows the scientist to easily deploy a custom ML script or chose among a selection of proprietary algorithms. The backend includes a ML engine that can run algorithms dependent on Tensorflow and Scikit Learn.

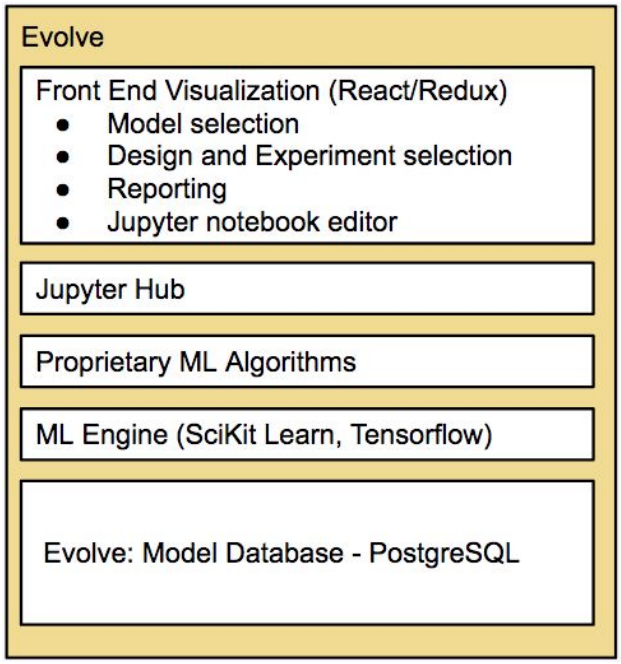


Figure 2: EVOLVE's front-end and back-end components

3 INTEGRATION WITH TEST

The EVOLVE module communicates directly with our DESIGN and TEST modules. The open source TEST module is based on the open source Experiment Data Depot (EDD) knowledge base [1] and provides the necessary data to train supervised machine learning algorithms. The output of the EVOLVE module can guide the next iteration of libraries to be designed and built.

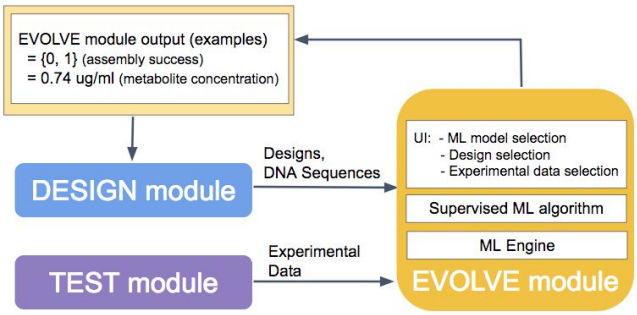


Figure 3: DESIGN, TEST and EVOLVE integration and feedback loop.

4 EVOLVE SDK

The TeselaGen EVOLVE SDK will provide powerful libraries written in Python, for designing and deploying machine and deep learning applications. It includes libraries for communicating with the DESIGN and TEST APIs, as well as our ML task messaging queue and our EVOLVE database.

5 OPTIMIZING DNA ASSEMBLIES

With TeselaGen's platform, researchers can design combinatorial libraries that include thousands of different variants. For many applications, it is not enough to design and build DNA constructs. Researchers need to be confident that their synthesized combinatorial or hierarchical libraries meet stringent quality assurance criteria. High throughput, high content screens common at many biotech and biopharma companies depend crucially on the efficient generation of screening candidates with high probability of success. TeselaGen's *j5* algorithm, which creates detailed instructions to assemble DNA, relies on the specification of assembly strategies, parameters, and rules that can be tuned to achieve optimal results. The tuning of this choices can be guided based on the output of bioinformatic tools such as j5 itself, as well as experimental DNA sequence validation results. As an example, our platform allows our users to align DNA sequencing runs with their reference designs, in order to validate the quality of their synthesized constructs. As we collect these DNA sequence validation datasets, the EVOLVE module can train machine learning models (like RNNs) that can help the biologist further refine their designs and assembly simulations.

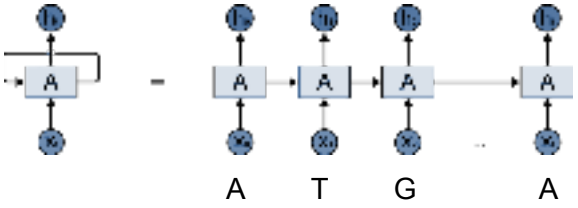


Figure 4: A recurrent neural network (RNN) is a network with memory, ideal for modeling sequences such as DNA.

6 CONCLUSIONS

TeselaGen has developed a powerful, cloud-based, computer aided design and build platform for accelerating synthetic biology. Our customers are already using our flexible informatics backbone to guide the construction of synthetic DNA to further the production of immunotherapy biologics, virus like particles (VLPs), sustainable chemicals, natural products, and plant modifications for enhanced agricultural traits.

TeselaGen Biotechnology has recently set the goal of developing a next-generation software platform that will harness state-of-the-art machine learning to assist customers with the design, build and Synthetic Evolution™ of their biological constructs. As companies seek to scale their synthetic biology efforts, they will benefit from TeselaGen's EVOLVE module to optimize their Synthetic Biology processes. The success will depend ultimately on how well scientists can collect and store experimental data in the TEST module. As our customers work closely with our platform, we will empower them with enabling design decisions that accelerate product development.

ACKNOWLEDGMENTS

This work was supported in part by NSF SBIR Phase IIB 1430986 and CORFO Grant 17IEAT-73382.

REFERENCES

[1] William C. Morrell, Garrett W. Birkel, Mark Forre, Teresa Lopez, Tyler W. H. Backman, Michael Dussault, Christopher J. Petzold, Edward E. K. Baidoo, Zak Costello, David Ando, Jorge Alonso-Gutierrez, Kevin W. George, Aindrila Mukhopadhyay, Ian Vaino, Jay D. Keasling, Paul D. Adams, Nathan J. Hillson, and Hector Garcia Martin, 2017. The Experiment Data Depot: A Web-Based Software Tool for Biological Experimental Data Storage, Sharing, and Visualization *ACS Synth. Biol.* 6 (12), pp 2248–2259, DOI: 10.1021/acssynbio.7b00204

[2] Nathan J. Hillson, Rafael D. Rosengarten, and Jay D. Keasling, 2012. j5 DNA Assembly Design Automation Software *ACS Synth. Biol.* 1 (1), pp 14–21, DOI: 10.1021/sb2000116

Identifying composition rules for TF circuits that control macrophage signal response with deep learning

Jenhan Tao, Gregory Fonseca, Christopher K Glass
Department of Cellular and Molecular Medicine
University of California
San Diego, CA, USA
jenhantao@gmail.com

Christopher Benner
Department of Medicine
University of California
San Diego, CA, USA

KEYWORDS

genetic circuits; transcription factors; regulation of transcription; cell signaling; machine learning; deep learning

1 INTRODUCTION

Regulation of gene expression in eukaryotic cells is mediated in part by hundreds of sequence specific transcription factors (TFs) that bind to their individual binding motifs at genomic sequences proximal to a gene (promoters) as well as at distal elements (enhancers). Promoters and enhancers can interact by looping together in three dimensional space. The binding of TFs at promoters and enhancers mediates the recruitment of cellular machinery necessary for transcription. Prior studies have suggested two classes of TFs: 1) lineage determining TFs (LDTFs) and and 2) signal dependent TFs (SDTFs). LDTFs play important roles in establishing cell type specific patterns of open chromatin (accessible regions of the genome) [5] whereas SDTFs bind in response to a cellular stimuli, resulting in cell-specific responses to signals [6] (Figure 1). These studies, and others, suggest that the context specific gene expression in a cell type is genetically encoded by combinations of TF binding motifs at millions of enhancers scattered throughout the genome [3].

Given the evidence that TFs act collaboratively, it naturally follows that individual TF motifs have been observed to be poor predictors of activation of an enhancer. The biological activity of an enhancer may depend on the specific composition of TF motifs - arrangement and spacing between TF motifs, as well as the sequence degeneracy of each motif [4], and evidence that the arrangement of motifs help to determine transcriptional activity, we endeavored to teach an artificial neural network (ANN) to predict signal dependent activation of enhancers by reading arrangements of motifs present at open chromatin regions. We hypothesize that different arrangements of motifs can be used to predict the response to different cellular stimuli.

2 EXPERIMENTAL DESIGN

Using ATAC-seq, and ChIP-seq targeting H3K27Ac, an enhancer mark associated with active chromatin, we defined active enhancers in mouse macrophage cells stimulated with an array of cytokines (IFN-g, IL-1b, IL-4, IL-5, IL-6, IL-13, IL-23, LPS, TNF-a, TGF-b). This experimental model provides several key advantages: 1) The macrophage is a well characterized immune cell with robust responses to signals such as cytokine. 2) By comparing one signal to another, we can distinguish between SDTFs and general TFs that play a role in many contexts (Figure 2). 3) Enhancers that respond to multiple signals offer an opportunity to study how elements that encode the response to individual signals can be composed together.

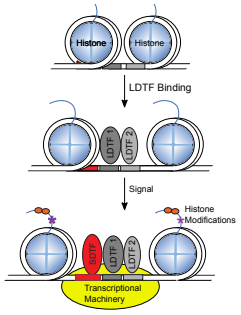


Figure 1: A collaborative hierarchical model for TF binding. Lineage determining TFs (LDTFs) bind collaboratively to make cell type regions of chromatin accessible. In response to a signal, a signal dependent TFs (SDTFs) bind at sites bound by LDTFs.

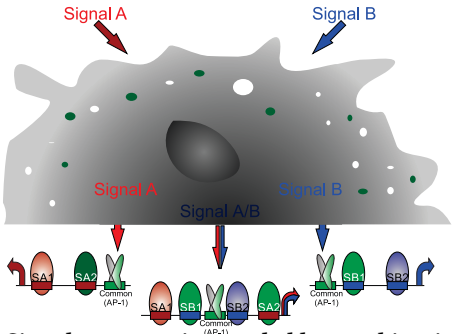


Figure 2: Signal response is encoded by combinations of TF binding sites. Activation of enhancers that respond to signals A and B are mediated by distinct sets of SDTFs ([SA1, SA2] and [SB1,SB2] respectively). Enhancers that respond to both signals should contain TF motifs that mediate both signals.

3 MODEL DESIGN

The sequence of each enhancer as well as the enhancers' response to each signal, is used as input to train an artificial neural network

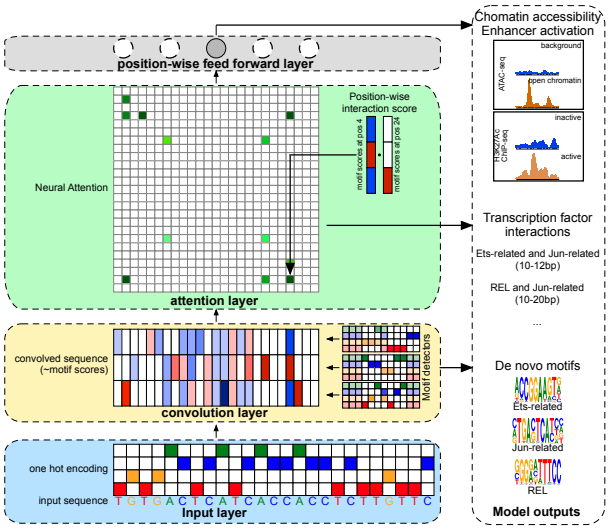


Figure 3: Overview of model

(ANN) with an attention mechanism to predict signal dependent activation of an enhancer. In contrast to traditional ANNs that combines the input data in a cryptic fashion (via a fully connected layer) to predict enhancer activity [1, 7], ANNs with an attention mechanism highlight which regions of the inputs (subsequences of enhancers that presumably are TF binding motifs) the ANN is paying attention to as it makes each prediction, thereby divulging the "reasoning" of the ANN. Here we implement a convolutional neural network that uses a dot product attention mechanism [2] to use genomic sequence alone to predict enhancer activity. The architecture of our neural network is shown in Figure 3.

4 PRELIMINARY RESULTS

To assess the performance of our model architecture, we compared the performance of our model against the current state of the art, a convolutional network. We trained our model and an implementation of DeepBind, a previously described convolutional network,[1], to distinguish accessible enhancers from random genomic sequences. The performance of our model exceeded that of the convolutional model, in terms of model accuracy and precision, at detecting enhancers present in macrophages in 3 separate treatment conditions (Table 1 Att versus Conv). Our model's increase in performance versus the convolutional network can be potentially attributed to the greater number of free parameters used (Table 1). And so, we also trained a large convolutional network (with 54 convolution kernels and 108 dense neurons versus 16 convolution kernels and 32 neurons in the original model). The improved performance of our model suggests that the attention mechanism is capable of extracting useful information.

5 FUTURE WORK

While we are encouraged by the performance of our model, we believe the insights we can extract from the network more important. We are currently extracting TF binding sites highlighted by our

			Model		
			Att	Conv	Large-Conv
		# params	10753	2162	10850
Tx	Veh	Acc.	0.854	0.822	0.846
		Prec.	0.838	0.804	0.830
	KLA-1h	Acc.	0.859	0.807	0.839
		Prec.	0.857	0.791	0.826
	IL4-24h	Acc.	0.862	0.832	0.847
		Prec.	0.858	0.809	0.836

Table 1: Model performance. Mean performance metrics (n=3), accuracy (Acc.) and precision (Prec.), of 3 models: our attentive model (Att.), a convolutional network (Conv), and a large convolutional network (Large-Conv) are shown for macrophages under 3 treatment conditions

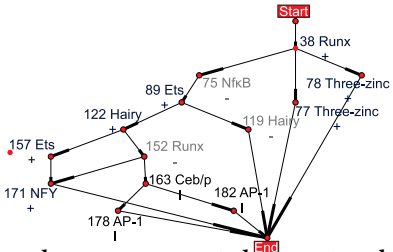


Figure 4: An enhancer represented as a network of TF motifs. Motifs are represented as nodes. Adjacent, motifs are connected with edges. The position are annotated at each node.

model and representing each enhancer as a network of TF motifs (Fig. 4). Next, we will calculate arrangements of motifs that are enriched at enhancers that respond to a specific cytokine. Thus, we can determine a compositions of TF motifs that encodes the transcriptional response to each cytokine, yielding insights into compositional rules for signal specific TF circuits.

REFERENCES

[1] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 33, 8 (2015), 831–838. https://doi.org/10.1038/nbt.3300

[2] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long Short-Term Memory-Networks for Machine Reading. (jan 2016). arXiv:1601.06733 http://arxiv.org/abs/1601.06733

[3] The ENCODE Project Consortium. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 7414 (sep 2012), 57–74. https://doi.org/10.1038/nature11247

[4] Emma K. Farley, Katrina M. Olson, Wei Zhang, Daniel S. Rokhsar, and Michael S. Levine. 2016. Syntax compensates for poor binding sites to encode tissue specificity of developmental enhancers. *Proceedings of the National Academy of Sciences* 113, 23 (jun 2016), 6508–6513. https://doi.org/10.1073/pnas.1605085113

[5] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C. Lin, Peter Laslo, Jason X. Cheng, Cornelis Murre, Harinder Singh, and Christopher K. Glass. 2010. Simple Combinations of Lineage-Determining Transcription Factors Prime cis-Regulatory Elements Required for Macrophage and B Cell Identities. *Molecular Cell* 38, 4 (may 2010), 576–589. https://doi.org/10.1016/j.molcel.2010.05.004

[6] S Heinz, C E Romanoski, C Benner, K A Allison, M U Kaikkonen, L D Orozco, and C K Glass. 2013. Effect of natural genetic variation on enhancer selection and function. *Nature* 503, 7477 (nov 2013), 487–92. https://doi.org/10.1038/nature12615 arXiv:NIHMS150003

[7] Daniel Quang and Xiaohui Xie. 2016. DanQ: A hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Research* 44, 11 (2016), 1–6. https://doi.org/10.1093/nar/gkw226

The Synthetic Biology Open Language Supports Integration of the Engineering Life-Cycle for Synthetic Biologists

Bryan Bartley[1], Christian Atallah[2], Alasdair Humphries[2], Vishwesh Kulkarni[3], Curtis Madsen[4], Goksel Misirli[5], Angel Goni-Moreno[2], Tramy Nguyen[6], Ernst Oberortner[7], Nicholas Roehner[1], Meher Samineni[6], Zach Zundel[6], Jacob Beal[1], Chris Myers[6], Herbert M Sauro[8], Anil Wipat[2]
[1]Raytheon BBN Technologies, [2]Newcastle University, [3]University of Warwick, [4]Boston University, [5]Keele University, [6]University of Utah, [7]DOE Joint Genome Institute, [8]University of Washington
bryan.a.bartley@raytheon.com

MOTIVATION

A critical bottleneck for large-scale engineering collaboration in synthetic biology has been the inability to integrate data through successive stages of the design-build-test-learn (DBTL) engineering life-cycle. These workflows generate large volumes of data and physical artifacts (e.g., DNA samples and cell stocks) that are difficult to organize, track, and manage without systematized, automated tool chains.

The DBTL cycle is a generalized, iterative framework for engineering problem-solving—something like a scientific method for engineers. In the context of synthetic biology, the DBTL cycle may include processes such as pulling data about biological parts from online databases, assembling new genetic programs from DNA sequences, synthesizing and assembling DNA, performing quality control, measurement and model-based characterization of a DNA part's encoded behavior, submitting characterized parts to inventories, and publishing data sheets. Ideally, each cycle generates new knowledge that feeds back into new cycles in the form of alternative approaches, reformulated problems, or forward specifications for future designs.

In this abstract, we describe how the *Synthetic Biology Open Language* (SBOL) data exchange standard has recently been extended to enable documentation, automation, and integration of DBTL pipelines to support large-scale, distributed research and development in synthetic biology.

REPRESENTING WORKFLOWS

Distributed collaboration in synthetic biology requires integrating a diverse network of resources, including software tools, automation, instrumentation, databases, and repositories. In order for these resources and, by extension, the collaborative, scientific communities they support to communicate and operate efficiently, data standards are needed. This abstract reports recent developments in data exchange using the SBOL standard that support large-scale collaboration among diverse communities of experimental and computational synthetic biologists.

SBOL is a data exchange standard intended to support reuse and reproducibility of prior scientific work by synthetic biologists and developed through an open, community-wide process. SBOL defines a high-level data model that represents important conceptual knowledge for human users, while, at a lower level, data is serialized in a machine-readable RDF/XML format that enables semantic interoperability between distributed resources. Since the standard

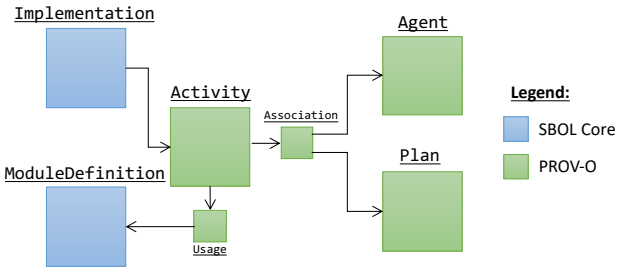


Figure 1: PROV-O and SBOL classes used to represent the design-build-test-learn (DBTL) engineering life-cycle.

was originally released [3], its scope has expanded [8], most recently with SBOL 2.2, which includes support for integration of computational workflows with experimental workflows [2].

In order to describe multi-stage workflows, SBOL leverages the *World Wide Web Consortium* (W3C) *Provenance Ontology* (PROV-O) [7]. Provenance may be defined as a form of structured meta-data that describes the execution of processes in which one artifact is transformed into another. This record is essential for understanding where data comes from, deciding whether it should be trusted, and integrating it with other information sources. In PROV-O, workflows are represented as a directed, acyclic graph linked by **Activities** executed by **Agents** (e.g., persons, robotics, and/or software tools) according to a **Plan** upon **Usages** of other artifacts (Figure 1). The PROV-O data model is deliberately generic. Although it has heretofore been used primarily for describing computational workflows, it has been adapted for use in SBOL to describe experimental workflows as well.

More specifically, SBOL classifies provenance **Activities** according to a simple DBTL ontology. Data objects produced by "design" **Activities** represent an engineer's intended design and are purely conceptual (these are typically **ComponentDefinition** or **ModuleDefinition** objects). Data objects produced by "build" **Activities** represent physical artifacts such as DNA samples or cell lines (these are typically **Implementation** objects). These in turn may be subject to "test" **Activities**, an experimental measurement that results in new data objects (these are typically an **Attachment** or a **Collection** of **Attachment** objects). These data are then subject to reduction and analysis through "learn" **Activities** (learned objects can be any other type of object representing what has been learned). Workflows in SBOL may consist of any number of specialized steps and protocols, but by and large they are expected to fit into the

DBTL abstraction, as this is how workflows are often presented in synthetic biology literature [1, 4].

Figure 2 represents one complete iteration through a hypothetical DBTL cycle. The cycle starts with a model specifying a gene circuit's desired behavior. A parts-based design is created with the iBioSim tool. Subsequently, a DNA construct is implemented in the lab by a technician and its behavior is measured using an automated plate-reader protocol. Finally, the data are fit with a mathematical model in order to characterize the observed behavior, which may or may not match the original specification. This scenario represents a model-based design approach to synthetic biology.

APPLICATIONS AND TOOL SUPPORT

SBOL 2.2 is currently being used for distributed collaboration in several large-scale synthetic biology efforts that are characterizing genetic parts and designs, including the NSF Living Computing Project and the EPSRC-funded Synthetic Portabolomics project. These efforts involve diverse communities of computational and experimental researchers. To support these efforts, software tools have been developed and upgraded to support PROV-O. These include, among others, synthetic biology design tools, such as SBOLDesigner [9], modeling tools, such as iBioSim [5], and data repositories, such as SynBioHub [6]. SynBioHub, in particular, provides a means to browse provenance histories online using its web interface. These tools leverage open source software libraries for Java (libSBOLj) [10], JavaScript (sboljs), C++ (libSBOL), and Python (pySBOL), which have been extended to support the creation and search of provenance histories using PROV-O and networked data exchange with SynBioHub instances.

CONTRIBUTIONS

SBOL 2.2 is an enabling technology that supports large-scale engineering efforts in synthetic biology [2]. Teams of synthetic biologists using experimental and computational methods may collaborate better using a growing company of resources and infrastructure that communicate with SBOL. Genetic designs, laboratory samples, and experimental data can be linked together by provenance histories, making it easier for one synthetic biologist to reuse the work of another. The pace of synthetic biology innovation will likely improve as synthetic biologists proceed through many iterations of DBTL with computer-aided and automated technologies.

ACKNOWLEDGMENTS

The authors of this work are supported by the National Science Foundation under Grant No., 1522074 (J.B., M.S., and C.M.), DBI-1355909 (H.M.S. and B.B.), and DBI-1356041 (Z.Z., T.N., and C.M.). A.W. was funded by Engineering and Physical Sciences Research Council (EPSRC) Grant Nos. EP/N031962/1 and EP/J02175X/1. A.G-M is funded by EPSRC grant EP/R019002/1. C.A. acknowledges studentship funding from the EPSRC and from Prozomix Ltd. E.O. has been funded through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U.S. Department of Energy. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

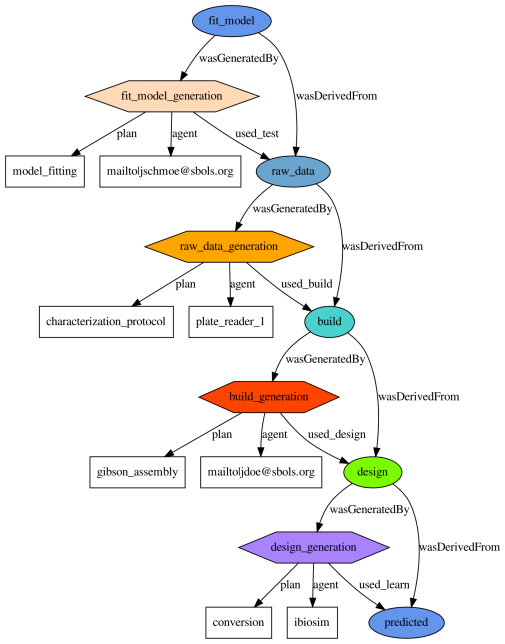


Figure 2: A hypothetical DBTL workflow representing model-based design. This figure is rendered using a Python tool for rendering SBOL 2.2 workflows (https://github.com/chrisAta/sbol-provo-viz).

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

REFERENCES

[1] E. D. Carlson. Creating ribo-t:(design, build, test) n, 2015.
[2] R. S. Cox, C. Madsen, J. A. McLaughlin, T. Nguyen, N. Roehner, B. Bartley, J. Beal, M. Bissell, K. Choi, K. Clancy, et al. Synthetic biology open language (SBOL) version 2.2.0. *Journal of integrative bioinformatics*, 2018.
[3] M. Galdzicki, K. P. Clancy, E. Oberortner, M. Pocock, J. Y. Quinn, C. A. Rodriguez, N. Roehner, M. L. Wilson, L. Adam, J. C. Anderson, et al. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology*, 32(6):545, 2014.
[4] C. A. Hutchison, R.-Y. Chuang, V. N. Noskov, N. Assad-Garcia, T. J. Deerinck, M. H. Ellisman, J. Gill, K. Kannan, B. J. Karas, L. Ma, et al. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.
[5] C. Madsen, C. J. Myers, T. Patterson, N. Roehner, J. T. Stevens, and C. Winstead. Design and test of genetic circuits using iBioSim. *IEEE Design Test of Computers*, 29(3):32–39, June 2012.
[6] J. A. McLaughlin, C. J. Myers, Z. Zundel, G. Misirli, M. Zhang, I. D. Ofiteru, A. Goñi Moreno, and A. Wipat. SynBioHub: A standards-enabled design repository for synthetic biology. *ACS synthetic biology*, 7(2):682–fi?!688, 2018.
[7] P. Missier, K. Belhajjame, and J. Cheney. The W3C PROV family of specifications for modeling provenance metadata. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 773–776. ACM, 2013.
[8] N. Roehner, J. Beal, K. Clancy, B. Bartley, G. Misirli, R. Grnberg, E. Oberortner, M. Pocock, M. Bissell, C. Madsen, et al. Sharing structure and function in biological design with SBOL 2.0. *ACS synthetic biology*, 5(6):498–506, 2016.
[9] M. Zhang, J. A. McLaughlin, A. Wipat, and C. J. Myers. SBOLDesigner 2: an intuitive tool for structural genetic design. *ACS synthetic biology*, 6(7):1150–1160, 2017.
[10] Z. Zhang, T. Nguyen, N. Roehner, G. Misirli, M. Pocock, E. Oberortner, M. Samineni, Z. Zundel, J. Beal, K. Clancy, et al. libSBOLj 2.0: a java library to support SBOL 2.0. *IEEE life sciences letters*, 1(4):34–37, 2015.

Standardizing Design Performance Comparison in Microfluidic Manufacturing

Methods and means for microfluidic physical design tools

Radhakrishna Sanka
Boston University
Boston, MA, USA
sanka@bu.edu

Brian Crites
University of California, Riverside
Riverside, CA, USA
bcrit001@ucr.edu

Joshua Lippai
Boston University
Boston, MA, USA
jlippai@bu.edu

Jeffrey McDaniel
University of California, Riverside
Riverside, CA, USA
jeffrey.mcdaniel@ucr.edu

Philip Brisk
University of California, Riverside
Riverside, CA, USA
philip@cs.ucr.edu

Douglas Densmore
Boston University
Boston, MA, USA
dougd@bu.edu

1 INTRODUCTION

The automation of rote laboratory experiments and the transformation of ultrahigh throughput, controlled in-vitro testing environments have burgeoned in the space of microfluidic design automation, attracting researchers from biology, electronic design automation (EDA) and computer engineering alike over the last decade. Today a large section of the microfluidic devices represented in the literature are not published with sufficient information for automating the physical design process. With the advent of component level design tools like 3DuF[1] [Lippai et al. 2018], the problem of microfluidic physical design automation is no longer just a computational problem to solved in a sandbox but also a necessity to proliferate the technology into research labs.

However, the lack of detailed design information accompanying published microfluidic designs has severely limited researchers' ability to work with test cases that are representative of the latest class of devices that are being used in research labs. The work done by CIDAR[2] at BU and CARES[3] at UC Riverside has resulted in the development of methodologies and standards that allow researchers to gauge the efficacy of developed algorithms.

2 DEPENDENCE ON MANUFACTURING

Evolving manufacturing technologies and protocols and the emergence of low cost manufacturing tools [Lashkaripour et al. 2018; Walsh et al. 2017] have lowered the entry barrier for manufacturing microfluidic devices. Since microfluidic device architectures have to date been primarily dictated by the capabilities of the manufacturing technologies used, the emergence of low cost manufacturing techniques has the potential to upend the assumptions and constraints that are factored into the physical design algorithms.

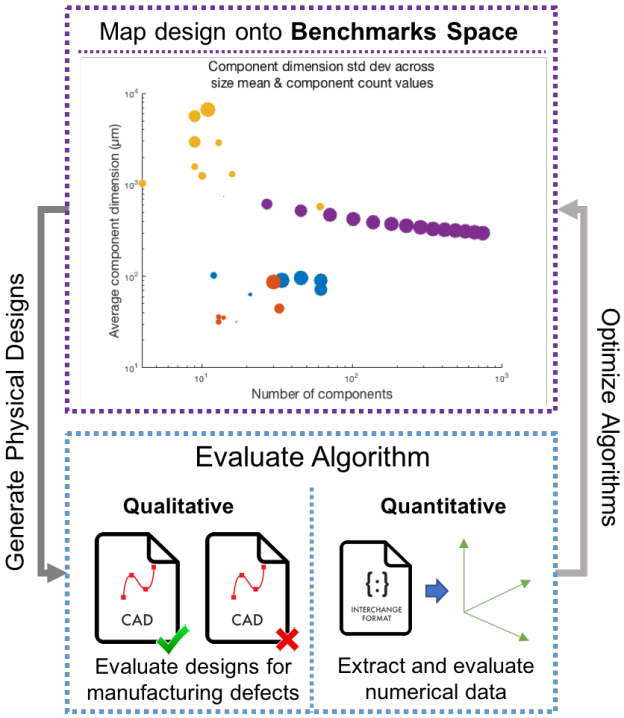


Figure 1: By generating and examining the designs of architectures that occupy various regions in the benchmark space, researchers can optimize/modify their physical design algorithms.

3 BENCHMARK SPACES

Any abstract architecture of a microfluidic device has potentially infinite ways in which it can be realized as a design. Moreover any solution for a design of a microfluidic chip

[1] http://3DuF.org
[2] http://cidarlab.org/
[3] http://www1.cs.ucr.edu/faculty/philip/

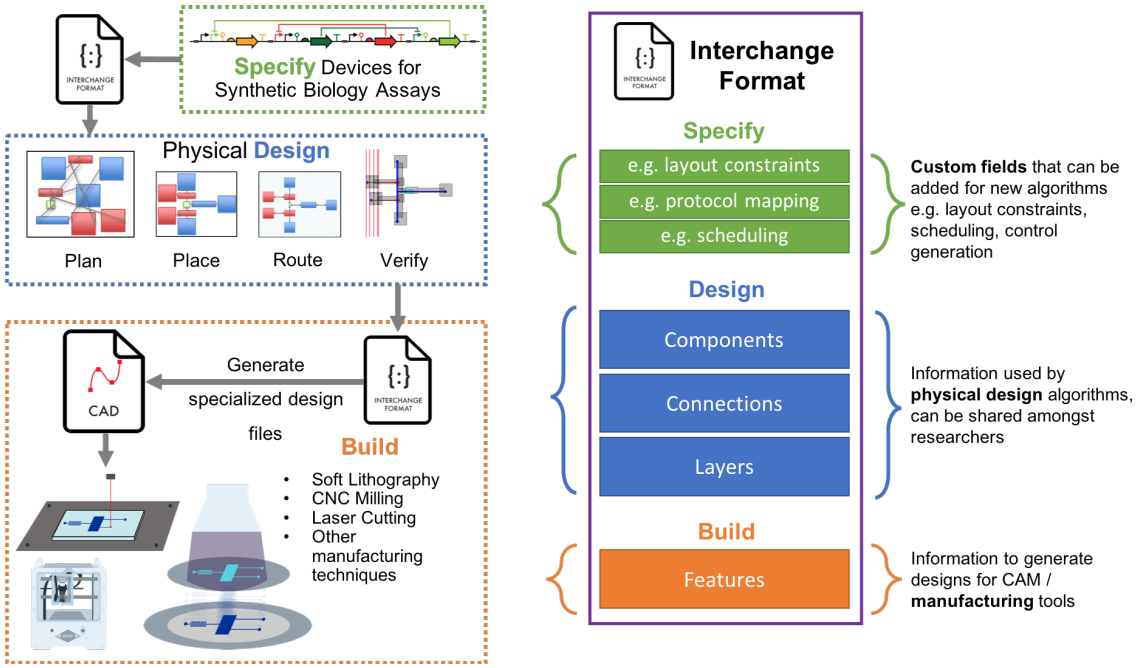


Figure 2: An interchange format allows for the capture and exchange of information that is essential for the physical design automation algorithms to work. In addition to capturing the design information, it is also allows the format to be extensible to include custom objects required by different tools allowing researchers to share the information across research domains.

could have numerous valid layouts based on the application space.

Hence to compare different devices and different algorithms we introduce *Benchmark Spaces* to understand the performance of algorithms on different devices compared against useful performance metrics. Each *benchmark space* is a 2D/3D visualization of the various microfluidic devices where each of the axes is a unique characteristic of the device. The graph in Figure 1 is an example of benchmark space characterizing the statistics of microfluidic components that constitute the device design. We believe that this visualization method allows the researchers to compare the quantitative and the qualitative results of their layout algorithms against different devices that occupy the same region in a *benchmark space*.

Since literature in the microfluidic physical design typically only characterize microfluidic devices by the number of components, connections. We believe that the work done towards formalizing and refining the parameters used in visualizing *benchmark spaces* will prove to be an invaluable resource to effectively monitor the efficacy of physical design algorithms against different classes of devices.

4 STANDARDS

While benchmark spaces can help address the problem of comparing vastly different microfluidic designs from different application spaces for the purposes of physical design, it is still necessary to create standards that not only ensures that the data can be shared efficiently between research groups that engage in algorithm research but also encourages device designers and manufacturers to adopt the standards. This is achieved by allowing the interchange format to include custom fields at the top level which can be used for application/algorithm-specific constraints. Figure 2 shows how the interchange format used for describing microfluidic device designs can capture the Specify, Design and Build work flow for microfluidic devices.

REFERENCES

Ali Lashkaripour, Ryan Silva, and Douglas Densmore. 2018. Desktop micromilled microfluidics. *Microfluidics and Nanofluidics* 22, 3 (26 Feb 2018), 31. https://doi.org/10.1007/s10404-018-2048-2

Joshua Lippai, Radhakrishna Sanka, Dinithi Samarasekera, Dylan Samperi, Sarah Nemsick, and Douglas Densmore. 2018. 3DuF - Interactive Design Environment for Continuous Flow Microfluidic Devices. *In preparation for Lab on a Chip* (2018).

David I. Walsh, David S. Kong, Shashi K. Murthy, and Peter A. Carr. 2017. Enabling Microfluidics: from Clean Rooms to Makerspaces. *Trends in Biotechnology* 35, 5 (May 2017), 383–392. https://doi.org/10.1016/j.tibtech.2017.01.001

Integrated computational extraction of cross-cancer poly-omic signatures

Extended Abstract*

Guido Zampieri
Department of Computer Science and Information Systems, Teesside University
Middlesbrough, United Kingdom
g.zampieri@tees.ac.uk

Claudio Angione
Department of Computer Science and Information Systems, Teesside University
Middlesbrough, United Kingdom
c.angione@tees.ac.uk

ABSTRACT

Understanding the interplay between metabolism and genetic regulation is considered key to shed light on the mechanisms underlying cancer onset and progression. In this work, we reconstruct a number of tumor-specific genome-scale metabolic models and inspect estimated flux profiles via statistical analysis, characterizing the detailed metabolic response associated to altered regulation in various tissues. We thus demonstrate that combining complementary computational techniques it is possible to identify poly-omic differences and commonalities across cancer types.

KEYWORDS

Genome-scale modeling, flux balance analysis, statistical data analysis, cancer metabolism.

ACM Reference Format:
Guido Zampieri and Claudio Angione. 2018. Integrated computational extraction of cross-cancer poly-omic signatures: Extended Abstract. In *Proceedings of 10th International Workshop on Bio-Design Automation (10th IWBDA).* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

Several recent studies have shown how cancer cells present distinct metabolic hallmarks, such as deregulated uptake of glucose and amino acids. Even the gene theory of cancer has been recently object of revision in light of old and new observations [1]. It is therefore clear that alterations on a genomic and a metabolic level do not work in isolation, but rather co-participate in malignant transformation. However, the precise rewiring in the metabolism of transformed cells requires more extensive elucidation. Here, we address this problem through the investigation of the entire metabolic states associated to altered genetic regulation in the NCI60 cancer cell line panel, which covers nine different tissues [2]. By combining genome-scale metabolic models (GSMMs) and statistical analysis we characterize the corresponding cross-cancer poly-omic landscape.

2 METHODS

Experimental data sets here employed are transcriptomic profiles, nutrient uptake rates and proliferation rates for 56 NCI60 cell lines, obtained from previous studies [3, 4]. We used this data to build and evaluate an array of cell line-specific GSMMs, starting from the human cell model Recon 2.2 [5]. In this process, a novel version of METRADE [6] was adopted to (i) transform normalized gene expression profiles by gene set rules (ii) obtain tumor-specific flux bounds taking into account both genetic and metabolic uptake constraints. The estimation of associated flux configurations is conducted by a regularized flux balance analysis (FBA) optimization task, as follows:

$$\begin{aligned} &\max_{\mathbf{v}} \ \mathbf{w}^\top \mathbf{v} - \frac{\sigma}{2}\mathbf{v}^\top \mathbf{v} \\ &\text{subject to } \ \mathbf{S}\,\mathbf{v} = 0\,, \\ &\mathbf{v}_{lb}\,\varphi(\boldsymbol{\Theta}) \le \mathbf{v} \le \mathbf{v}_{ub}\,\varphi(\boldsymbol{\Theta})\,. \end{aligned} \tag{1}$$

Here $\mathbf{w}$ is a real vector expressing the contribution of each reaction to the objective and $\sigma = 10^{-6}$ is a regularization parameter. Vectors $\mathbf{v}_{lb}$ and $\mathbf{v}_{ub}$ represent native flux bounds in Recon, while vector $\varphi(\boldsymbol{\Theta})$ models the reaction-level gene regulation state in any cell line based on the following map:

$$\varphi(\boldsymbol{\Theta}) = \delta\,(1 + \gamma\,|log(\boldsymbol{\Theta})|)^{sgn(\boldsymbol{\Theta}-1)}\,. \tag{2}$$

In this equation, $\boldsymbol{\Theta}$ is obtained from transcript abundances by converting logical gene-protein-reaction rules into max/min operations, as originally implemented in METRADE [6]. Moreover, γ is a parameter representing the magnitude with which gene expression affects reaction rates, while δ is a scaling factor introduced to adjust native flux bounds to experimental uptake rates.

*Oral presentation

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

10th IWBDA, August 2018, Berkeley, California USA

© 2018 Copyright held by the owner/author(s).

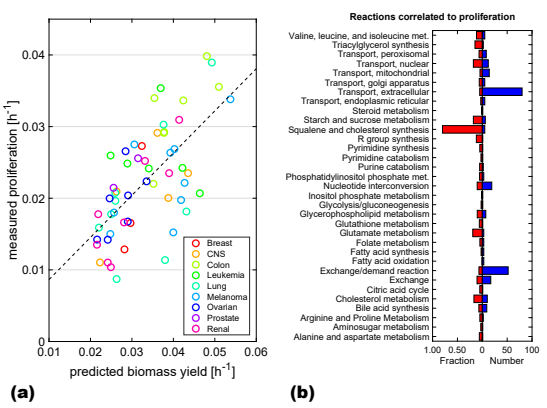


Figure 1: (a) Comparison between biomass yield predicted by each cell line-specific GSMM and the corresponding experimentally measured proliferation rates at the optimal γ and δ values. (b) Overview of metabolic reactions whose predicted fluxes significantly correlate with measured cellular proliferation (1% threshold). For each pathway, number and fraction of significantly correlated reactions are visualized in blue and red, respectively.

We performed a sensitivity analysis on parameters γ and δ in Eq. (2) to evaluate the obtained flux profiles in terms of the Pearson correlation coefficient (PCC) r between predicted cellular growth and experimentally measured proliferation rate. The predicted growth was computed through Eq. (1) assuming biomass accumulation as a proxy for cell proliferation and thus as a meaningful FBA objective to model cancerous metabolism. Repeated PCC estimation allowed identifying optimal γ and δ values across several orders of magnitude. We carried out regularized FBA using the COBRA toolbox in Matlab and the quadratic solver Gurobi [7]. Finally, using the FactoMineR package in R [8] we performed principal component analysis (PCA) to characterize the cross-tumor variation at a genome-scale metabolic flux level.

3 RESULTS

As a result of the sensitivity analysis on parameters γ and δ in Eq. (2), we obtained a PCC peak where $r \simeq 0.66$, $p\text{-}value \simeq 1.5 \cdot 10^{-8}$ (Fig. 1a). We thus inspected the whole flux profiles of tumor cells by studying their PCC with respect to cellular proliferation rates. We observed a significant PCC (threshold 1%) for reactions in a number of cancer-associated pathways, supporting the reliability of our GSMMs, as well as in less obvious pathways (Fig. 1b). These may suggest or corroborate unknown mechanisms for tumor development. In particular, the majority of cholesterol synthesis pathway emerges as correlated to proliferation, supporting its debated involvement in cancer. As another example, the exchange

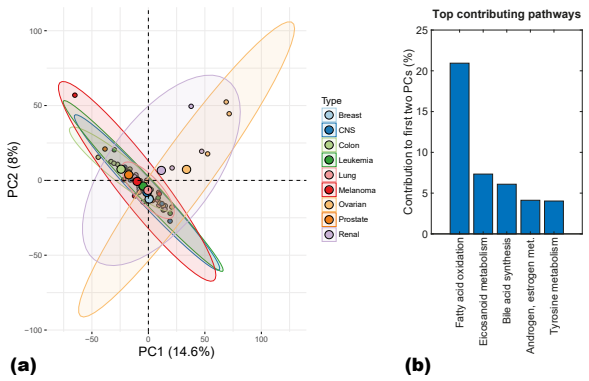


Figure 2: (a) Variability across flux profiles relative to different tumor types in the space of the first two principal components. (b) Contribution to the first two principal components of the most highly contributing pathways, obtained by summing the contributions of their associated reactions.

of dietary compounds such as maltodextrins also results associated to proliferation.

Next, PCA of the flux profiles allowed detecting poly-omic heterogeneities across the cell lines. As Fig. 2a shows, the ovarian and renal cell tumors present a markedly distinct metabolic behavior, almost orthogonal to all other tissues. A closer look at the composition of first principal components allowed identifying key pathways underlying such variation, like fatty acid oxidation or eicosanoid metabolism (Fig. 2b). This analysis thus highlights potential links in the metabolic reprogramming of the two cancer types, suggesting also precise reactions to focus experimental verification on.

4 CONCLUSIONS

In this work, we analyzed the poly-omic configurations of multiple cancer types through an integrated computational pipeline and within a comprehensive cross-tumor framework. Our analysis led to the identification of both variation and common patterns across the tumors, providing novel insights in the general cancer molecular landscape. We thus showed that the joint application of GSMMs and statistical analysis techniques can help elucidate the mechanisms underlying cancer development and progression.

REFERENCES

[1] Thomas N. Seyfried et al. 2014. *Carcinogenesis* 35, 3 (2014), 515–527.
[2] Uwe Scherf et al. 2000. *Nat Genet* 24, 3 (2000), 236–244.
[3] Christian Diener and Osbaldo Resendis-Antonio. 2016. *Front Physiol* 7 (2016), 644.
[4] Daniel C. Zielinski et al. 2017. *Sci Rep* 7 (2017), 41241.
[5] Neil Swainston et al. 2016. *Metabolomics* 12, 7 (2016), 109.
[6] Claudio Angione and Pietro Lió. 2015. *Sci Rep* 5 (2015), 15147.
[7] Jan Schellenberger et al. 2011. *Nat Protoc* 6 (2011), 1290.
[8] Sébastien Lê et al. 2008. *J Stat Softw* 25, 1 (2008), 1–18.

Towards Computer-Aided Synthetic Developmental Biology

Evan Appleton[1,2], Michael Moret[1,2], Tristan Daifuku[1,2], Demarcus Briers[3], Iman Haghighi[3], Noushin Mehdipour[3], Calin Belta[3], and George Church[1,2]

[1]Department of Genetics, Harvard Medical School, Boston, MA
[2]Wyss Institute for Biologically Inspired Design, Boston, MA
[3]Division of Systems Engineering, Boston University, Boston, MA

1. MOTIVATION

Developmental biology concerns the growth and differentiation of a body of cells from a single-cell progenitor with complete genetic information. From this single cell, a complete multi-cellular organisms is developed autonomously - in order for scientists to be able to consistently grow functional tissues or tissue-like substructures, it would be ideal to develop strategies for genetic-level control of development of 3-D shapes.

Current approaches to forming specific cellular structures typically require scaffolding or 3-D printing [4], both of which require considerable experimenter intervention. Moreover, protocols now exist to generate disparate organoids (e.g. human cerebral organoids) from a range of progenitor cell types (e.g. pluripotent stem cells), yet in these approaches, researchers are still a long way from being able to use these processes to create *ex vivo* organs. Furthermore, the capacity to program cells to form novel, specified synthetic structures via these endogenous developmental programs has not been demonstrated [5]. In this work, we present a computational method to facilitate the genetic encoding of biological materials into customizable shapes without any experimenter intervention, by defining developmental biology subproblems and providing some preliminary solutions. Our framework takes as input a computer-aided design (CAD) of a 3D cellular shape and outputs a set of DNA instructions informing a progenitor cell how to develop autonomously into the specified shape.

2. FRAMEWORK OVERVIEW

First, a set of algorithms transform the computer-aided design of the shape into a set of building blocks – alike lego blocks – with a defined connectivity pattern. Orthogonal protein binding pairs will ensure the right pattern. Second, the structural specifications defined in the first step are encoded into synthetic genetic circuits (**Fig 2 & 3**) that will inform the cells on a developmental plan (**Fig 4**) to direct the cells to grow into the designed 3D cellular structure. Finally, a verification step assess the biological experiment in order to quantify its success against the CAD specifications.

3. STRUCTURAL SPECIFICATIONS

Our framework inputs a 3D computer-aided design of a cellular shape and transforms it into a list of blocks. Those blocks, either made of eight cells (rhombuses) or four cells (tetrahedron), have been designed taking inspiration from the early developmental stages of the human embryo. To segment our desired shape into blocks, we rely on meshing algorithms – widely used in the field of finite element analysis to divide a physical object into multiple parts in order to simulate the behavior of the object resulting from applied loads and constraints [7].

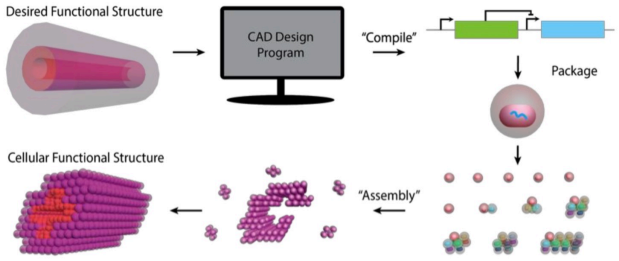


Figure 1: Synthetic developmental biology CAD workflow. From the desired shape, a CAD design is created and an algorithm outputs a genetic circuit necessary for a cell to grow into the desired shape.

To connect the blocks together, orthogonal protein binding pairs (connectors) are used. Given that their number is limited, optimization algorithms are used in order to minimize the number of connectors needed in order to create a given shape. This step allow us, given a set of N available connectors, to increase the space of possible shapes that can be designed. To this end, we build on top of graph theory concepts and algorithms in order to create multiple connectivity schemes having different shape properties. Notably, the lower-bound connectivity is define by the minimum spanning tree (MST) algorithm while the upper-bound connectivity is define by a full connection pattern between all cells of neighboring blocks [6].

4. DEVELOPMENTAL PLAN

From the list of blocks and their connectors, the developmental plan algorithm is used to encode the necessary information into a genetic program. Those information will allow the progenitor cell to autonomously divide into the designed shape. To inform the cell of the program, a mechanism composed of two specific synthetic genetic circuits has been designed, called a counter and a register. The counter circuit uses promoters specific to a part of the cell cycle to count cell divisions. This counter will provide a biological mechanism to activate expression of specific genes in the register circuit at specific cell divisions (e.g. to express a specific surface binding protein). Both circuit rely on recombinases and reversal cofactors (XIS/RDF) to change their states by

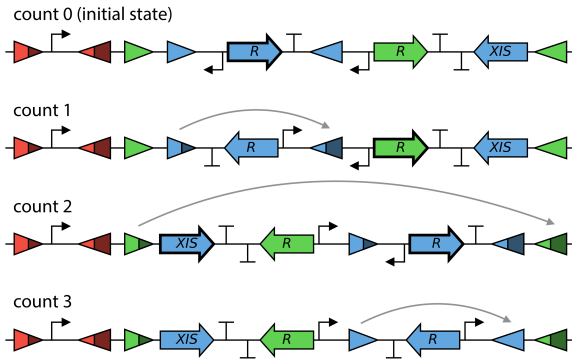


Figure 2: A 'binary' counter to count 4 states using 2 recombinases. Recombinase sites are depicted with triangles. Genes with *R* represent a recombinase, and *XIS*, a reversal cofactor. Bolded elements are expressed at the given state and arrows indicate sequences inversed to increment the count.

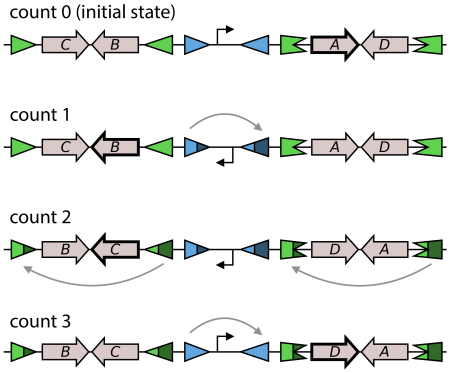


Figure 3: A 'binary' register to express specific genes according to the program given by the counter. Cut triangles represent orthogonal binding sites.

inverting sub-sequence of the circuit lying between recombinase sites [2].

Finally, an algorithm using a binary tree structure is used to compute the needed information on the register circuit. Starting with the leafs of the tree, where each cell is defined by the connectors it needs to express, information is propagated back to all parent nodes until the root node is reached. At each level of the tree, information is translated into genetic instructions on the register to be expressed at given count by specific cells. To ensure that cells can expressed different connectors despite coming from the same progenitor cell, a mechanism to do asymmetrical division is used. Namely, the *Numb* protein, which has been shown to segregate in only one of two daughter cells during division in Drosophila melanogaster Neuroblasts, can be fused to a recombinase to modify the register asymmetrically [3].

5. VERIFICATION AND SIMULATION

To verify a synthetic gene circuit design can reliably produce a specified 3D geometry, we developed a biophysical model to simulate the directed self-assembly of a single cell into autonomous cell blocks and 3D cell shapes. Our computational model integrates the propulsion of cells in a rotating fluid, selective cell adhesion using single or paired surface binding proteins, and gene circuit-driven logic to direct the self-assembly of individual cells into geometric cell shapes.

After experiments (or simulations *in silico*), the cell mass

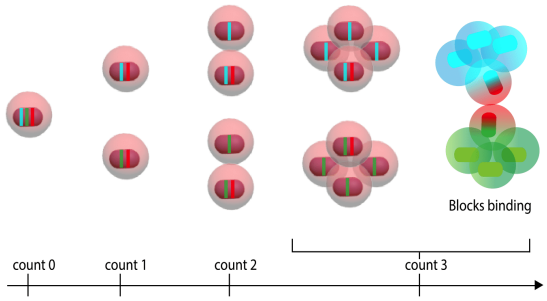


Figure 4: Example of a developmental plan to create a shape made of two building blocks from a single progenitor cell. Cell colors are used to display a type of connector (genotype on cell nucleus, phenotype for the whole cell).

is imaged with confocal microscope. A graph is reconstructed from the resulting z-stack of 2D images with segmentation algorithms, allowing to place cell in 3D space [1]. The reconstruction is then compared with the original design to quantitatively assess the result of the experiment.

6. RESULTS

We can use our computational framework to compute developmental plans to create custom simple 3D cellular shapes designed on an open source CAD software. Moreover, we have developed a comprehensive strategy for shape formation from a single-cell progenitor using only genetic circuits and have developed a recombination circuit verification software to verify that the counter design is extensible and can count to 2^n with (n-1) reversible recombinases and express specific proteins at each individual count with a register.

7. CONCLUSION

Our approach allows to encode a custom 3-D shape into a set of genetic instructions such that a progenitor cell can develop into the desired shape, without extensive experimenter intervention.

8. REFERENCES

[1] S. O. V. V.-M. J. G.-R. Alberto Garcia-Garcia, Sergio Orts-Escolano. A review on deep learning techniques applied to semantic segmentation. *arXiv*, 2017.

[2] N. D. Grindley, K. L. Whiteson, and P. A. Rice. Mechanisms of site-specific recombination. *Annu. Rev. Biochem.*, 75:567–605, 2006.

[3] J. A. Knoblich, Y. N. Jan, et al. Asymmetric segregation of numb and prospero during cell division. *Nature*, 377(6550):624, 1995.

[4] D. B. Kolesky, K. A. Homan, M. A. Skylar-Scott, and J. A. Lewis. Three-dimensional bioprinting of thick vascularized tissues. *Proceedings of the National Academy of Sciences*, 113(12):3179–3184, 2016.

[5] B.-K. Koo, D. E. Stange, T. Sato, W. Karthaus, H. F. Farin, M. Huch, J. H. Van Es, and H. Clevers. Controlled gene expression in primary lgr5 organoid cultures. *Nature methods*, 9(1):81, 2012.

[6] M. V. Oliver Mason. Graph theory and networks in biology. *IET systems biology*, 2006.

[7] O. C. Zienkiewicz, R. L. Taylor, O. C. Zienkiewicz, and R. L. Taylor. *The finite element method*, volume 3. McGraw-hill London, 1977.

Automated design of gene circuits with optimal mushroom-bifurcation behaviour

Extended Abstract

Rubén Pérez-Carrasco
Department of Mathematics
University College London
London, UK
r.carrasco@ucl.ac.uk

Irene Otero-Muras
BioProcess Engineering Group
Spanish National Research Council (CSIC)
Vigo, Spain
ireneotero@iim.csic.es

Julio R. Banga
BioProcess Engineering Group
Spanish National Research Council (CSIC)
Vigo, Spain
julio@iim.csic.es

Chris P. Barnes
Department of Cell and Developmental Biology
University College London
London, UK
christopher.barnes@ucl.ac.uk

ABSTRACT

We present an automated design method to find gene circuits compatible with a target bifurcation diagram optimizing through parameter and topology spaces. We apply the method to the design of gene circuits exhibiting the so called mushroom bifurcation, finding the set of minimal topologies that lead to an improved sensor functionality.

CCS CONCEPTS

• **Applied computing** → **Computational biology**;

KEYWORDS

Synthetic Biology; Bifurcation; Multiobjective Global Optimization;

ACM Reference Format:
Rubén Pérez-Carrasco, Irene Otero-Muras, Julio R. Banga, and Chris P. Barnes. 2017. Automated design of gene circuits with optimal mushroom-bifurcation behaviour: Extended Abstract. In *Proceedings of ACM conference (IWBDA'18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

One of the main challenges of synthetic biology is to design and implement circuits capable of specific tasks optimally while keeping a minimal design [3, 4]. The limited resources of the cell restrict the combination of multiple working circuits in the same organism. This gives leading relevance to the design of multifunctionality: how can different behaviours be integrated in the same circuit? Powerful tools to answer this question are provided by bifurcation theory of dynamical systems, linking the topology of the network (given by a set of ODEs) with the different dynamics available under a controllable input. This is the case of the mushroom bifurcation: combining the behaviour of two toggle switches, the mushroom presents an ON state that is only available for intermediate values of a signal (Fig. 1B) [5, 6], allowing us to build a precise signal detector. In addition, it contains two different bistable zones where the state of the cell will be determined by the signal history, endowing the cell with memory capabilities. Finally, the critical slow-down close to the neck of the mushroom can result in an efficient mechanism to control time, only responding after the signal has been present for a required amount of time.

The efficiency of the possible different behaviours of the mushroom bifurcation will depend on the shape of its bifurcation diagram. Here we develop an automated design method to find (searching trough parameter and topology spaces simultaneously in an efficient manner) gene circuits that not only allow certain behaviours (compatible with a target bifurcation diagram), but are also optimized for specific sets of functions.

2 METHODS, RESULTS AND DISCUSSION

We encode the dynamics of gene regulation for the 2-gene system in Fig. 1A via a mixed-integer framework. Using the Shea-Ackers formalism, the gene circuit is characterized by a vector y of 4 integer variables (y_{uu}, y_{uv}, y_{vu}, y_{vv}) and a vector x of 10 real variables coding for tunable parameters (including promoter strengths, leakiness, degradation rate constants, repression and activation terms). Activation functions by the quorum signal *AHL* (denoted by S) are of the Hill type. Generalizing the condition for fold bifurcation in [2], the target behaviour can be encoded as a function of $[x, y]$ such that the mushroom bifurcation is achieved when the function reaches its minimum. Then, we formulate the search as a global optimization problem aiming to find those circuits $[x, y]$ minimizing the objective. The resulting mixed-integer nonlinear programming problem (MINLP) is solved with efficient hybrid solvers [1]. In order to guarantee efficient performance and a successful implementation in the lab, we seek the structure and parameter ranges leading to an optimal performance in terms of the distances between the limit bifurcation points (see Fig. 1B). We set the mushroom condition as a constraint and solve a multiobjective optimization problem with two objectives: maximize distance 1 and minimize distance 2 as defined in Fig. 1B. The MO-MINLP problem is solved following [1].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
IWBDA'18, July 31–August 3, 2017, Berkeley, CA, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

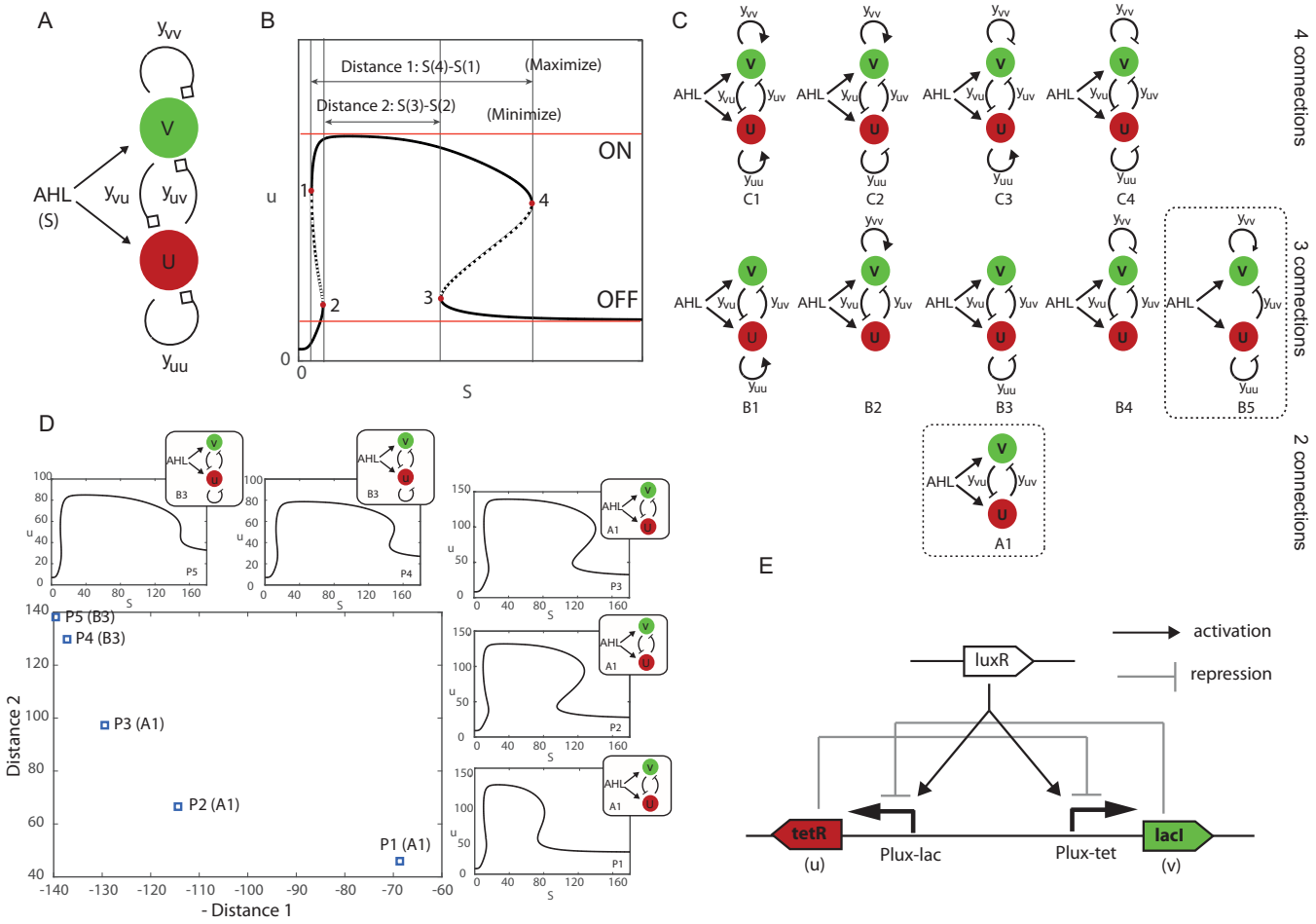


Figure 1: A) Super-structure of the two gene design. B) Bifurcation diagram of a mushroom toggle switch, points 1, 2, 3 and 4 indicate saddle-node bifurcations. C) All the 2-gene structures leading to mushroom bifurcations found by the single objective optimization algorithm. D) Pareto front of optimal solutions, for each solution (P1 to P5) the structure and bifurcation diagram are depicted. Distances 1 and 2 are defined in Fig 1B. E) Implementation of the design (with A1 topology).

Results. By solving the single objective optimization problem (in a multistart strategy) we find 10 different topologies leading to mushroom bifurcation behaviour. These structures are represented in Fig. 1C and classified attending to the number of active connections. There are two core topologies (for which no connection can be removed without losing the mushroom functionality), corresponding to structures A1 and B5. Unlike exhaustive exploration strategies, our optimization-based method can find structures fulfilling the target behaviour very efficiently (in the order of seconds).

Applying the multiobjective optimization approach, we obtained the Pareto front of optimal solutions in Fig. 1D (P1, . . ., P5). Circuits corresponding to point P2 and P3 provide a good compromise between optimization objectives and therefore both are good candidates for implementation. In Fig. 1E we depict one possible implementation of the design. Importantly, the same optimization strategy can be used to find circuits compatible with other target bifurcation behaviours and also starting from a library of standard components.

Acknowledgements. RPC acknowledges financial support by the UCL Mathematics Clifford Fellowship. CPB acknowledges financial support by Wellcome Trust Research Career Development Fellowship (097319/Z/11/Z). IOM, JRB acknowledge funding from MINECO projects SYNBIOFACTORY (DPI2014-55276-C5-2-R) and SYNBIOCONTROL (DPI2017-82896-C2-2-R)

REFERENCES

[1] I. Otero-Muras and J. R. Banga. 2017. Automated Design Framework for Synthetic Biology Exploiting Pareto Optimality. *ACS Synt. Biol.* 1180–1193 (2017), 6(7).
[2] I. Otero-Muras, P. Yordanov, and J. Stelling. 2017. Chemical Reaction Network Theory elucidates sources of multistability in interferon signaling. *PLoS Comp. Biol.* 13(4) (2017), e1005454.
[3] Ruben Perez-Carrasco, Chris P. Barnes, Yolanda Schaerli et al, et al. 2018. Combining a Toggle Switch and a Repressilator within the AC-DC Circuit Generates Distinct Dynamical Behaviors. *Cell Syst.* (mar 2018), 1–10.
[4] Yolanda Schaerli, Andreea Munteanu, Magüi Gili, et al. 2014. A unified design space of synthetic stripe-forming networks. *Nat. Commun.* 5, May (2014), 4905.
[5] Dola Sengupta and Sandip Kar. 2018. Deciphering the Dynamical Origin of Mixed Population during Neural Stem Cell Development. *Biophys. J.* 114, 4 (2018), 992.
[6] Hao Song, Paul Smolen, Evyatar Av-Ron, et al. 2006. Bifurcation and singularity analysis of a molecular network for the induction of long-term memory. *Biophys. J.* 90, 7 (2006), 2309–2325.

Mechanistic effects of influenza in bronchial cells through poly-omic genome-scale modelling

Elisabeth Yaneske
Department of Computer Science and Information Systems, Teesside University
Middlesbrough, United Kingdom
e.yaneske@tees.ac.uk

Claudio Angione
Department of Computer Science and Information Systems, Teesside University
Middlesbrough, United Kingdom
c.angione@tees.ac.uk

ABSTRACT

In this work we propose regularised bi-level constraint-based modelling to determine the fluxomic profiles for four different influenza viruses, H7N9, H7M7, H3N2 and H5N1. We report here the first step of the analysis of the flux data using AutoSOME clustering, where we identify novel biomarkers of infection. This is a work in progress that can directly lead to novel therapeutic targets.

CCS CONCEPTS

• **Applied computing** → **Computational biology**; **Systems biology**;

KEYWORDS

genome-scale models; regularisation; bi-level optimisation.

ACM Reference Format:
Elisabeth Yaneske and Claudio Angione. 2018. Mechanistic effects of influenza in bronchial cells through poly-omic genome-scale modelling. In *Proceedings of 10th International Workshop on Bio-Design Automation (IWBDA).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Previous work [6] analysed transcriptomic data to identify FDA-approved antiviral drugs that would be effective against the H7N9 Anhui01 influenza virus. This was done by infecting human bronchial epithelial cells cells with H7N9 and comparing the transcriptomic profile of these with cells infected with H3N2, H5N1 and H7N7. Four replicate samples were taken at 3, 7, 12, and 24 hours. A control batch of uninfected cells was also sampled at the same time. Here we extend on this work by applying genome-scale modelling to the transcriptomic profiles of the four strains of influenza virus H3N2, H5N1, H7N7, H7N9 in order to determine their metabolic phenotypes.

Standard metabolic models created using FBA (Flux Balance Analysis) and constraint-based modelling have no unique solution for the optimal flux vector. The Cobra 3.0 toolbox [4] introduced a regularisation function so that the optimisation problem has a single unique solution. We here adapt the regularisation function to create a novel bi-level linear program with FBA and regularisation. To our knowledge, this is the first time this has been reported in the literature. This modelling procedure enables us to predict how the distribution of flux rates within the cell responds to infection with different influenza viruses. The transcriptomic data from each individual virus is used to constrain the model to generate a virus-specific metabolic model for each of the four influenza strains at each of the four time points sampled.

2 METHODS

Data processing and metabolic modelling

After retrieving the transcripomic data was from GEO (GSE49840), the probe data was matched to HGNC IDs. Where multiple probes were associated with a single HGNC ID, the gene expression values were averaged. The replicate samples were averaged to give a single transcriptomic profile for each time point. The transcriptomic data was normalised by taking the ratio of the influenza data to the control data to obtain the fold change. The normalised transcriptomic profiles of the influenza viruses were then used to create virus specific bronchial epithilel cell metabolic models. The metabolic models were created using constraint based modelling and flux balance analysis (FBA) of the human epithelial cell augmented with transcriptomics [6] through GEMsplice [1].

Constraint-based modelling with regularisation

In FBA the cell is assumed to be in steady state, $Sv = 0$, where S is a stoichiometric matrix of all known metabolic reactions (metabolites by reactions) and v is the vector of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IWBDA, July 31 - August 3 2018, Berkley, CA USA
© 2018 Association for Computing Machinery.
ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

reaction by flux rates. Additionally, every reaction flux is constrained by lower- and upper- bounds ($v^{\min}$ and $v^{\max}$). Here we constrain the strain-specific metabolic models generated from the transcriptomics data with upper- and lower-bounds on reactions set as a function of the expression level of the genes involved in the reactions using GEMsplice [1]. We set the primary objective as maximisation of hexokinase [7] and the secondary objective as maximisation of UDP-N-acetylglucosamine diphosphorylase [10]. We additionally apply regularisation to the secondary objective function such that it is maximised subject to the primary objective being maximised with a penalty term defined as a multiple of $v^T v$. This is achieved by adding a function that drives minimisation of the squared flux rates. This state reflects the most efficient metabolic network. We use the following bi-level program with regularisation:

$$\begin{aligned} &\max\ g^\intercal v - \frac{\sigma}{2} v^T v \\ &\text{such that} \qquad \max\ f^\intercal v, \quad Sv = 0, \qquad (1) \\ &\qquad\qquad v^{\min}\varphi(\Theta) \le v \le v^{\max}\varphi(\Theta). \end{aligned}$$

The Boolean vectors f and g are weights to select the first and second objectives respectively to be maximised from the vector v i.e. hexokinase and UDP-N-acetylglucosamine diphosphorylase. The vectors $v^{\min}$ and $v^{\max}$ represent the lower- and upper-bounds for flux rates. The regularisation function ($\frac{\sigma}{2} v^T v$) requires that the sum of the square of the fluxes is minimised for the maximisation of the second objective to be obtained. To maintain the optimal value of the original linear objective whilst minimising the square of the fluxes, the coefficient, σ, is set to 10^{-6}.

The vector Θ represents the set of gene expression values for the enzymes catalysing the biochemical reactions associated with the vector of fluxes v. The upper- and lower-bounds are constrained depending on the expression levels of the enzymes and a rule based on the type of enzyme (single enzyme, isozyme, or enzymatic complex) using the function φ [2]. Simulations were performed in Matlab R2016b.

Clustering

To cross-compare the fluxomics of the four viruses, flux distributions were clustered using AutoSOME [8], an unsupervised SOM-based method for high-dimensional data that uses a combination of density equalisation, minimum spanning tree clustering and ensemble averaging strategies. AutoSOME has the advantage that it does not require prior knowledge of the number of clusters and is not skewed by outliers in the data.

3 RESULTS AND CONCLUSIONS

Clustering the influenza sample subsystems according to their flux profile using AutoSOME resulted in four clusters.

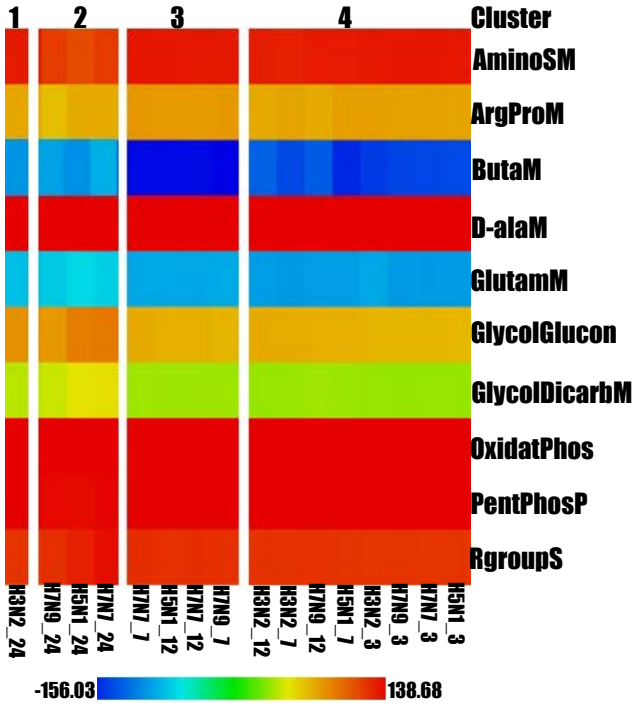


Figure 1: Heatmap of AutoSOME clustering. A subset of the subsystems is shown illustrating the variability between the four clusters.

In agreement with [6], H7N9 shows its own unique metabolic profile. Among the avian viruses at 24 hours of influenza infection, the metabolic profile of H7N9 is closest to H5N1 though it shares similarities with the H3N2 virus at both 12 and 24 hours [6]. Of the pathways showing strongest perturbations, the pentose phosphate pathway [9], oxidative phosphorylation [3] and r-group (de novo viral protein) synthesis [11] have previously been identified as important in viral replication. The importance of amino sugar metabolism may be due to its links with glycolysis [9] and glycoprotein production [10]. D-alanine metabolism has not previously been reported but may be important in the production of pyruvate [9] for viral replication. Butanoate metabolism shows a different profile across the four clusters. Butanoate metabolism has also not previously been reported but may highlight differences in viral cAMP signalling [5]. These results identify novel biomarkers of infection, suggesting that further analysis of the data using machine learning techniques focussed on these metabolic features could contribute to the identification of novel therapeutic targets.

REFERENCES

[1] C Angione. 2018. Integrating splice-isoform expression into genome-scale models characterizes breast cancer metabolism. *Bioinformatics* 34, 3 (2018), 494–501.

[2] C Angione and P Lió. 2015. Predictive analytics of environmental adaptability in multi-omic network models. *Scientific reports* 5 (2015), 15147.

[3] Adi Bercovich-Kinori, Julie Tai, Idit Anna Gelbart, Alina Shitrit, Shani Ben-Moshe, Yaron Drori, Shalev Itzkovitz, Michal Mandelboim, and Noam Stern-Ginossar. 2016. A systematic view on influenza induced host shutoff. *Elife* 5 (2016), e18311.

[4] L Heirendt, S Arreckx, T Pfau, S N Mendoza, A Richelle, A Heinken, H S Haraldsdottir, S M Keating, V Vlasov, J Wachowiak, et al. 2017. Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3. 0. *arXiv preprint arXiv:1710.04038* (2017).

[5] Noriyuki Hirata, Futoshi Suizu, Mami Matsuda-Lennikov, Tatsuma Edamura, Jyoti Bala, and Masayuki Noguchi. 2014. Inhibition of Akt kinase activity suppresses entry and replication of influenza virus. *Biochemical and biophysical research communications* 450, 1 (2014), 891–898.

[6] L Josset, H Zeng, S M Kelly, T M Tumpey, and M G Katze. 2014. Transcriptomic characterization of the novel avian-origin influenza A (H7N9) virus: specific host response and responses intermediate between avian (H5N1 and H7N7) and human (H3N2) viruses and implications for treatment options. *MBio* 5, 1 (2014), e01102–13.

[7] Hinissan P Kohio and Amy L Adamson. 2013. Glycolytic control of vacuolar-type ATPase activity: a mechanism to regulate influenza viral infection. *Virology* 444, 1-2 (2013), 301–309.

[8] A M Newman and J B Cooper. 2010. AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number. *BMC bioinformatics* 11, 1 (2010), 117.

[9] Joachim B Ritter, Aljoscha S Wahl, Susann Freund, Yvonne Genzel, and Udo Reichl. 2010. Metabolic effects of influenza virus infection in cultured animal cells: Intra-and extracellular metabolite profiling. *BMC systems biology* 4, 1 (2010), 61.

[10] David J Vigerust and Virginia L Shepherd. 2007. Virus glycosylation: role in virulence and immune interactions. *Trends in microbiology* 15, 5 (2007), 211–218.

[11] Tokiko Watanabe, Shinji Watanabe, and Yoshihiro Kawaoka. 2010. Cellular networks involved in the influenza virus life cycle. *Cell host & microbe* 7, 6 (2010), 427–439.

Temporal Verification of Genetic Circuits

Curtis Madsen[1], Prashant Vaidyanathan[1], Nicholas A. DeLateur[2], Evan Appleton[3], Greg Frasco[1], Calin Belta[1], Ron Weiss[2], Douglas Densmore[1]
[1]Boston University, [2]Massachusetts Institute of Technology, [3]Harvard Medical School
{ckmadsen,prash,frascog,cbelta,dougd}@bu.edu,{delateur,rweiss}@mit.edu,evan_appleton@hms.harvard.edu

1 INTRODUCTION

Recent advances in synthetic biology have led to the development of software tools capable of computationally designing functional genetic circuits using *Boolean logic* [7]. However, biologists are often concerned with how their systems behave over time instead of how they behave in the steady-state. This concern has led researchers to turn towards the use of *temporal logics* such as *Signal Temporal Logic* (STL) [5] for their specifications. To further address this interest in temporal behavior, we have developed Phoenix, a *bio-design automation* (BDA) workflow that utilizes STL to specify functionally rich desired behaviors of signals in synthetic genetic circuits. The workflow can then design genetic circuits from a well characterized library of DNA parts and verify the simulations of the models of these circuits against the desired specification. With this approach, biologists can create formal, performance-bound specifications for complex genetic circuits and run finite-time simulations for modular designs to identify genetic circuits with high likelihoods of satisfying the desired specification. Figure 2 illustrates the Phoenix workflow being applied to the design of a simple inverter circuit, and the rest of this abstract describes the steps of the workflow in more detail.

2 SPECIFICATION

The first step in Phoenix is the specification step. The following subsections describe each of the inputs to this step.

2.1 Performance

The temporal behavior of a desired circuit is specified as an STL formula. This formal language allows for the creation of specifications that include parameters intrinsic to genetic components, interactions with complex environments and other components, and timing of interactions and events. The STL formula is used in the verification step to determine which circuit design has the highest likelihood of realizing the desired behavior.

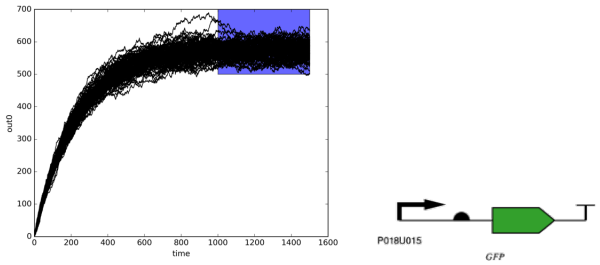
2.2 Structure

Functioning biological constructs adhere to specific structural constraints and biological rules [8]. A structural specification which includes constraints like counting (number of occurrences of a part), position (juxtaposition or index of parts in a design), orientation (forward or reverse), and functional interaction (interaction between a *coding sequence* (CDS) and promoter) helps narrow the design space of all possible circuits that can be built using the available parts in the library. To specify biological rules and constraints, Phoenix takes as input a structural constraint file of the desired genetic circuit written in the expressive specification language, Eugene [8].

2.3 Module Library

Our workflow connects to the SynBioHub [6] to download *Synthetic Biology Open Language* (SBOL) [1] descriptions of biological parts for composition into genetic circuit designs. These SBOL descriptions can be annotated with structural information as well as characterization data gathered while performing experiments using the parts. The SynBioHub also allows for the inclusion of mathematical models specified in languages such as the *Systems Biology Markup Language* (SBML) [3] that describe the functional behavior of the parts in the repository.

This work has been funded by the Office of Naval Research under Grant No. N00014-11-1-0725 and the National Science Foundation under grant CPS Frontier 1446607.



(a) Stochastic Simulations (b) Constitutive Expression

Figure 1: (a) shows 100 stochastic simulations of the constitutive expression of GFP circuit shown in (b). The area shaded in blue represents the region of satisfaction of the desired performance specification written in STL as $G_{[1000,1500]}(GFP >= 500 \;\&\&\; GFP < 700)$, which specifies that GFP must be between 500 and 700 MEFL (Molecules of Equivalent Fluorescein) from time 1000 mins to 1500 mins. In this example, this circuit satisfies the specification with a satisfaction rate of 1.

3 ASSIGNMENT AND SIMULATION

Using the structural specification, Eugene generates a design space of all rule-compliant circuit designs. These circuit designs are then decomposed into each transcriptional unit present (both forward and reverse orientations). A transcriptional unit (of a specific orientation) starts with a promoter, followed by a *ribosome binding site* (RBS), followed by a CDS, and ends with a terminator. A transcriptional unit could have multiple CDSs as long as each CDS is immediately preceded by an RBS of the same orientation. Each transcriptional unit is broken down into the 'promoter unit' (where expression is characterized) and the CDS (where protein degradation and dilution is characterized).

Models for circuit designs are composed using the models for expression and loss of proteins, which are attached to the components in the library. For CDSs, a single loss term representing degradation and dilution is included in the model. For promoters, the models include reactions with rates utilizing standard Hill-function equations. The part models utilized by our approach are carefully constructed so that when a promoter model is combined with a CDS model, the resulting model represents the behavior of a classic genetic module. In this current framework, we only consider the biological functions of a promoter and CDS. It should also be noted that these mathematical models are derived using parameter estimation techniques to fit the Hill-functions and loss equations

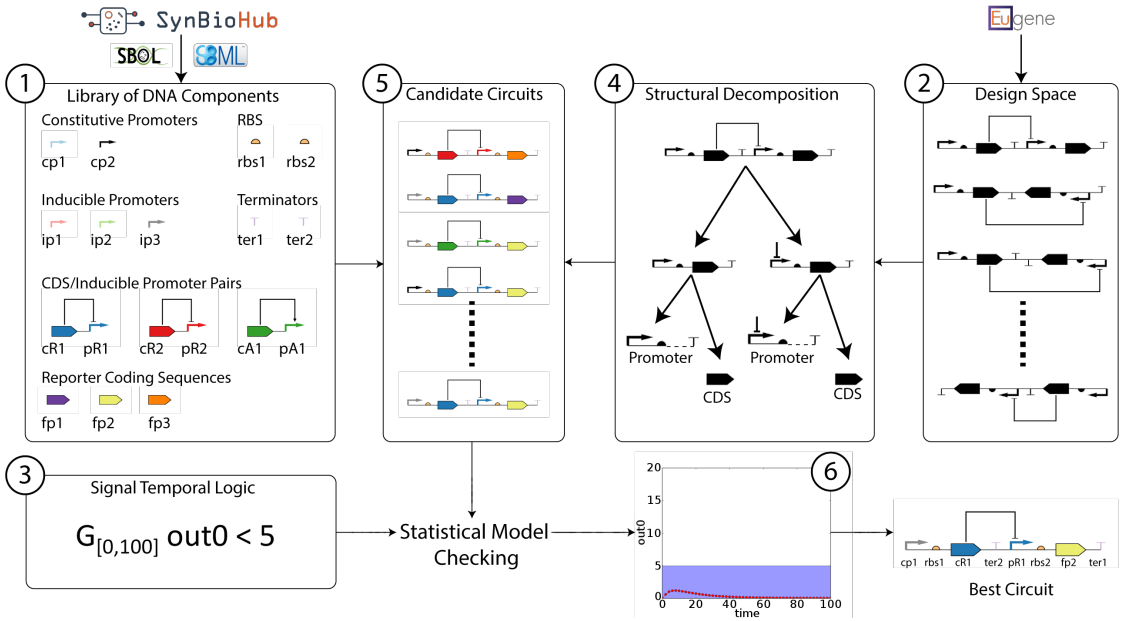


Figure 2: An example of using Phoenix to create a simple inverter circuit. (1) In the Specify step, the workflow connects to a SynBioHub repository of biological parts including several promoters, RBSs, CDSs, and terminators. (2) A structural specification, written in the Eugene language, is provided that specifies that one CDS is driven by a constitutive reporter and another is driven by a promoter that is repressed by the output protein of the first transcriptional unit. (3) A performance specification written in STL is also supplied that requires the output of the circuit to stay below 5 MEFL for 100 time units. (4) In the Design step, the biological parts are queried from the repository and composed together to create candidate circuits based on the design space constraints from the structural specification. (5) Models for each of these designs are simultaneously constructed by composing the models for each part from the repository. (6) In the Verify step, the composed models associated with the candidate circuits are simulated and checked against the performance specification using statistical model checking. The circuit that best satisfies the specification is returned as the result.

to data produced from wet-lab experiments for each part and that the parameter estimation step is a preprocessing step to Phoenix.

Once models of the genetic circuits are constructed, they are simulated to produce traces representing their behavior. Due to the stochastic nature of genetic circuits, our workflow utilizes stochastic analysis methods in order to better capture the possible behaviors of the circuit models. Simulations are performed using iBioSim [4] and the resulting traces are statistically verified.

4 STATISTICAL MODEL CHECKING

Utilizing well-known model checking techniques such as those applied to applications in electronic circuit design and robotics [2], our approach can determine which circuit designs best realize the desired performance specification. *Statistical model checking* is applied to each circuit design's simulation traces to determine the likelihood that each circuit will satisfy the specification. The circuit designs are then ordered based on the satisfying probability for each circuit. If a threshold value has been specified for the lowest possible acceptable satisfaction probability, this list will be truncated to only those circuits that are guaranteed to have at least the threshold satisfaction rate before being returned. Figure 1 shows an example of performing stochastic simulation and statistical model checking on a circuit for constitutive expression of GFP. Once Phoenix produces the circuit model that best satisfies a desired STL specification, the circuit is synthesized in the wet-lab and the resulting empirical data is checked against this same STL formula to determine how well Phoenix predicted the circuit's behavior. This information can then be fed back into the workflow to improve future predictions.

5 EXPERIMENTAL RESULTS

Kinetic parameters for expression and degradation/dilution in exponential phase *E. coli* have been obtained and are being utilized by the Phoenix workflow. We are currently building a library of empirically characterized genetic circuit modules for inducible and repressible expression. The models for these modules take transcription factor and small molecule concentrations as inputs to the interaction with their cognate promoter part.

REFERENCES

[1] Cox III, R. S., et al. Synthetic biology open language (SBOL) version 2.2.0. *J. Integrative Bioinformatics* (2018).

[2] Fainekos, G. E., et al. Temporal logic motion planning for mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (April 2005), pp. 2020–2025.

[3] Hucka, M., et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics 19*, 4 (2003), 524–531.

[4] Madsen, C., et al. Design and test of genetic circuits using iBioSim. *IEEE Design & Test of Computers 29*, 3 (2012), 32–39.

[5] Maler, O., and Nickovic, D. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 2004, pp. 152–166.

[6] McLaughlin, J. A., et al. SynBioHub: A standards-enabled design repository for synthetic biology. *ACS Synthetic Biology 7 2* (2018), 682–688.

[7] Nielsen, A. A., et al. Genetic circuit design automation. *Science 352*, 6281 (2016), aac7341.

[8] Oberortner, E., et al. A rule-based design specification language for synthetic biology. *ACM Journal on Emerging Technologies in Computing Systems (JETC) 11*, 3 (2014), 25.

Phoenix is currently in closed beta testing and can be accessed by contacting the authors.

An Automated BioModel Selection System (BMSS) for Gene Circuit Design

Kai Boon Ivan Ng
Department of Biomedical Engineering
National University of Singapore
Synthetic Biology for Clinical & Technological Innovation
Singapore
ivanng23@gmail.com

Jing Wui Yeoh
Department of Biomedical Engineering
National University of Singapore
Synthetic Biology for Clinical & Technological Innovation
Singapore
jingwui_yeoh@nus.edu.sg

Chueh Loo Poh
Department of Biomedical Engineering
National University of Singapore
Synthetic Biology for Clinical & Technological Innovation
Singapore
poh.chuehloo@nus.edu.sg

ABSTRACT

Constructing a complex working gene circuit composed of different modular standardized biological parts to achieve the desired performance could be challenging without a proper understanding of how the individual modules behave. Mathematical models play an important role towards better quantifying and optimizing the performance of the overall gene circuit, providing insights and guiding the design of experiments. As different gene circuits might require exclusively different mathematical representations, one of the key challenges in model development is the selection of the appropriate model. To address this, we developed an automated biomodel selection system, based on a framework which includes a library of pre-established models. As a proof of concept, we showed the system worked successfully using commonly used chemical inducible systems. Future work includes extending the BMSS to handle more complex gene circuits and to be compatible with SBOL/SBML for ease of use. Our intent is to assist the users to derive the best candidate mathematical model in a fast, efficient and automated way using characterization data.

KEYWORDS

Automation; biological part; model fitting; model selection; characterization data

1 INTRODUCTION

The use of mathematical models in synthetic biology allows a representation of the essential aspects of the constructed system, capturing the system behaviors in a quantitative manner useful for analysis and rational design optimization (e.g. design of experiments). The model development process involves structure identifications, formulation derivations, parameter inferences, model verifications and validations. Abstracting the experimental data in this way is a tedious yet complicated process that requires extensive experience and knowledge of the respective system of interests. This often requires many iterative trial-and-error learning and testing cycles which can essentially take months. Automating this process could drastically reduce the time consumed with minimal manual interventions from users.

To date, a plethora of computer-aided software or modeling tools is available to facilitate the synthetic biology design and modeling processes [2,3]. However, while the tools provide useful functions and interactive GUIs, there is still a lack of automated features that could expedite the process of selecting the most appropriate candidate model. Most of them demand moderate to intensive manual efforts from users. Hence, we aim to develop a system to automate the biomodel development and selection processes, providing a means to efficiently derive the best candidate model using characterization data.

2 METHODOLOGY

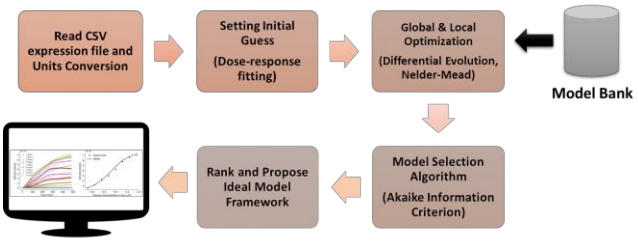


Figure 1: Workflow of system algorithm

Fig. 1 illustrates the schema of the system algorithm. The system was written in Python 3 as the scripting language which is open source and freely accessible. Numerical integration, optimization, and plotting packages were used to solve the ordinary differential equations (ODEs), iteratively fit models to experimental data through minimizing the sum squared residuals and plotting graphical results for data visualization. The model selection algorithm was based on the Akaike Information Criterion [1].

3 SYSTEM DEVELOPMENT

As a proof of concept, three widely used promoter-inducer circuits (i.e. pBAD/Arabinose, pTet/aTc, pLac/IPTG), which exhibit distinct gene expression behaviors, were chosen to create three different model representations. These promoter systems with the red fluorescent protein (RFP) as a reporter were characterized in *E. coli* and the expression output was measured using a microplate reader (BioTek Synergy H1). The characterization data of the pBAD promoter induced by arabinose exhibited an expression profile which could be recapitulated with model assuming a constant inducer concentration/inducer

activation (Fig. 2a, 3a-b). The pTet promoter characterization induced by aTc displayed expressions that slowly reduce over time which could be attributed to the fast inducer degradation or deactivation of the transcription process caused by the inducer binding/unbinding mechanism. A simpler model with fewer introduced parameters was adopted to describe this trend (Fig. 2b, 3c-d). Lastly, the IPTG-induced pLac promoter system that manifests a large initial delay in gene expressions was well described by an active transport mechanism which agrees with the nature characteristics of the system (Fig. 2c, 3e-f).

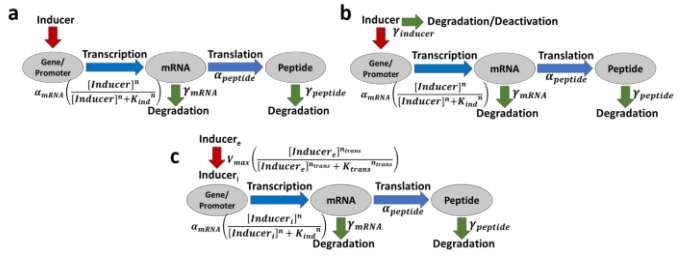


Figure 2: Schematic diagrams of mass action kinetics for three distinct model representations (a) The constant inducer activation model (b) The inducer degradation or deactivation model (c) The delayed inducer activation model.

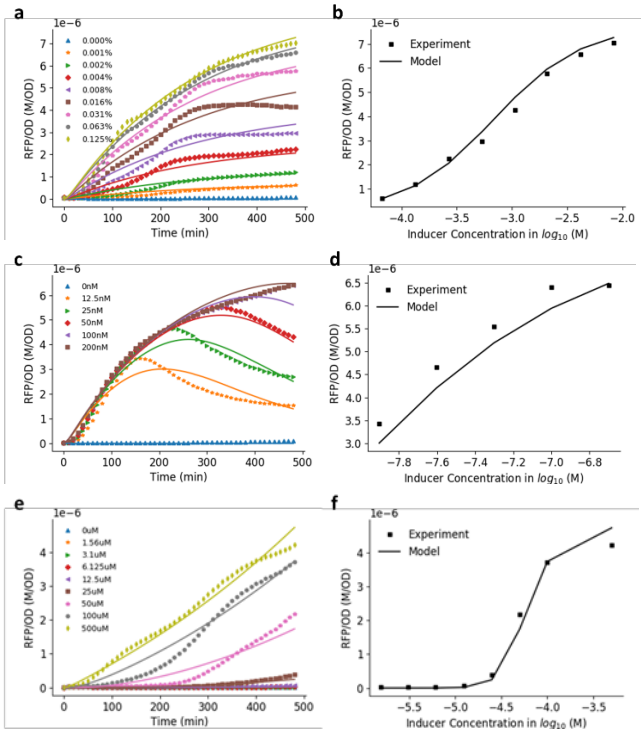


Figure 3: Temporal expressions and their corresponding dose-response profiles for experimental characterization data (filled symbols) and the best fitted models (solid lines) (a-b) pBAD/Arabinose (c-d) pTet/aTc (e-f) pLac/IPTG

4 SYSTEM VALIDATION

Two independent sets of characterization data from different promoter systems were used to verify the effectiveness of the BMSS (Fig. 4). The first data is from a LasR promoter system controlled by the AHL inducer, whereas the second data is from an arabinose inducible promoter under rbs34, which has a weaker strength compared to the default RBS used in the training sets. The model representation with constant inducer (Fig. 2a) was ranked the highest by the system for both the data (Fig. 4). The identified model corroborates the fast diffusion mechanism of the small molecules, AHLs, and the fast activation of arabinose inducers.

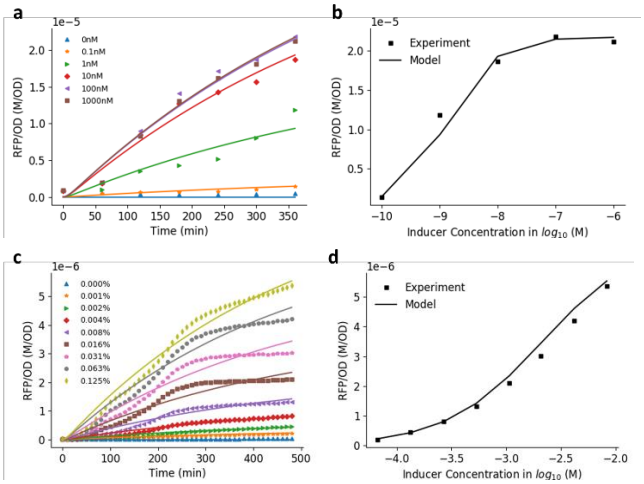


Figure 4: Temporal expressions and dose-response profiles from models (solid lines) and experiments (filled symbols) based on two independent characterization data (a-b) pLasI/AHL (c-d) pBAD/Arabinose (rbs34).

5 DISCUSSIONS

The preliminary results showed that the system was able to rank different model representations based on characterization data and estimate the best-fitted parameters in an automated manner. The selection algorithm ranks the models based on the trade-off between their goodness-of-fits and the model complexities to prevent overfitting. Future work includes expanding the library of model representations to capture the different and more complex gene circuits so as to extend its usefulness and implementing the BMSS to be compatible with SBOL/SBML. Nonetheless, this system could eventually serve as a pre-screening platform to be coupled with human interpretations to expedite the model development process.

REFERENCES

[1] Kenneth P. Burnham and R. Anderson. David. 2003. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media.

[2] Joao Cardoso, Kristian Jensen, Christian Lieven, Anne Sofie Lærke Hansen, Svetlana Galkina, Moritz Emanuel Beber, Emre Özdemir, Markus Herrgard, Henning Redestig, and Nikolaus Sonnenschein. 2018. Cameo: A Python Library for Computer Aided Metabolic Engineering and Optimization of Cell Factories. *ACS Synthetic Biology* 7: 1163–1166. https://doi.org/10.1021/acssynbio.7b00423

[3] Adrian L. Slusarczyk, Allen Lin, and Ron Weiss. 2012. Foundations for the design and implementation of synthetic genetic circuits. *Nature Reviews Genetics* 13, 6: 406–420. https://doi.org/10.1038/nrg3227

Spatiotemporal principles of genetic circuit design

Ruud Stoof
Newcastle University
United Kingdom
r.stoof2@ncl.ac.uk

Alexander Wood
Newcastle University
United Kingdom
Alexander.Wood@ncl.ac.uk

James A. McLaughlin
Newcastle University
United Kingdom
j.a.mclaughlin@ncl.ac.uk

Anil Wipat
Newcastle University
United Kingdom
anil.wipat@ncl.ac.uk

Ángel Goñi-Moreno
Newcastle University
United Kingdom
angel.goni-moreno@ncl.ac.uk

MOTIVATION

A critical bottleneck for the engineering of (more) robust and reliable synthetic gene circuits has been the disregard for the intracellular spatial organization of bacterial cells. In recent years, it has become increasingly clear that bacteria are highly ordered organisms, localizing and coordinating their vital functions in both time and space. Our recent experimental results [1] highlight the role of distance between circuit components on final performance. This suggests that each gene of a given genetic network may need a specific location within the cell-volume for optimal performance.

This abstract describes the computational framework developed to design genetic circuits considering spatial constraints e.g. the diffusion of molecules and the localization of DNA components. Spatiotemporal effects are showcased by a circuit composed of two connected genetic inverters (NOT logic gates). While the inverters were incompatible when localized in proximity, they became compatible when the distance between them was enlarged. This suggests that in order to fine-tune circuit performance, it is not only the nature of the components that matters, but also their location. A comprehensive analysis of how circuit dynamics can be influenced by intracellular space will impact both the understanding of biological processes and the ability to program living cells.

SPATIOTEMPORAL GENE REGULATION

Genetic circuits are bio-molecular devices able to perform functions that mimic those observed in electronic circuits. In these, specific combinations of genetic logic gates determine the way genes are regulated across the circuit. It is through these regulation events that genetic information is processed from input signals to output responses. Unlike electronic circuits, in which signals are unequivocally carried from one gate to another through dedicated wires (i.e. one wire per signal), genetic circuits share the same intracellular space for all inter-gate communications – molecular signals share the same *"wire"* . The physical distance between genetic components (Figure 1 top), for instance between the source of transcription factors and their target promoter, is a critical feature that can drastically change the performance of such communication [1] and, consequently, the function of the whole circuit. Despite the small volume of a cell, molecules are not always where they are needed; they must *travel* the distance from where they are expressed to the component they regulate.

Transcription factors (TFs) are not homogeneously distributed [2] and are thus more likely to meet their target promoter if both components are located nearby [3]. An indication of this is that natural regulatory networks have been observed to be to some extent clustered in the chromosome - related genes are more likely to be found in proximity [4]. Moreover, it has also been observed that gene expression noise patterns and strength change according to this spatial effect [1, 5]. An intriguing question is whether inter-genetic distance has been exploited as a tool by evolution to fine tune systems to deploy beneficial phenotypes. The corresponding challenge is then to embed spatial constraints as engineering principles to improve the design-build-test-learn lifecycle.

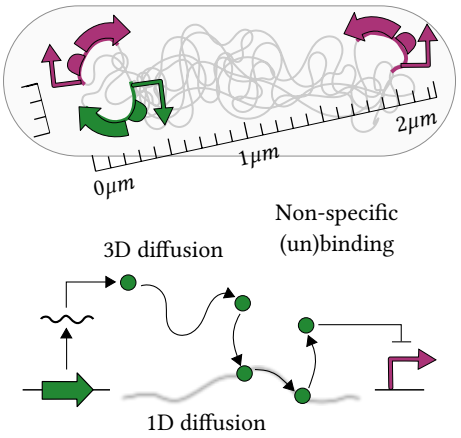


Figure 1: Spatial constrains of gene regulation. Top: sketch of the physical distance between the DNA components of a genetic circuit inserted into the chromosome. This distance is then correlated against phenotypic measurements to elucidate spatiotemporal regulation. Bottom: dynamics considered when modelling transcription factor-promoter interplay – 3D diffusion in cytoplasm, non-specific (un)binding to DNA regions other than the target promoter, and 1D diffusion along the chromosome.

MATHEMATICAL FRAMEWORK

In order to include spatial constraints in mathematical modelling, we decomposed the TF *binding rate* (which would define the promoter-TF reaction rate in a homogeneous cell) into two components. Firstly, the promoter-TF molecular affinity [6] (the ability of the

two molecular parts to interact physically). Secondly, the ability of a TF to reach the location of the promoter. While the definition of the former is relatively straightforward, the latter involves several dynamics (Figure 1 bottom): [i] the three-dimensional diffusion of a TF through the cytoplasm once it has been transcribed and translated, [ii] the (un)binding to non-specific DNA regions and [iii] the one-dimensional diffusion along the chromosome (i.e. *sliding*). TFs have been reported to spend up to 90% of their lifetime bound to the DNA (both target and non-specific regions) [7]. Although that is most of the TF's lifetime, 3D diffusion through the cytoplasm is much faster than 1D diffusion along the DNA [8].

We formalized all of these dynamics in a model for *facilitated diffusion* [9] and use it to analyse and improve the compatibility and modularity of genetic inverters (NOT logic gates).

SPATIOTEMPORAL CIRCUIT DESIGN

Spatiotemporal constraints have the potential to play an important role within the set of principles for genetic circuit design [10]. The software Cello [11] was developed to exploit such principles by exploring the combination of genetic logic gates to find compatible solutions that assembled larger circuits. Using a library of well characterized gates, Cello searches for the best ones to achieve a predefined Boolean function. If a gate does not have the appropriate dynamic range, for instance, it is discarded and another one takes its place. Our results suggest that there is another possibility: do not discard the gate, but just change its location. This way the number of combinations among a finite list of gates increases and more optimal solutions could be found.

Figure 2 shows how the compatibility of two genetic inverters is space-dependent. In fact, the characterization of response functions (concentration of input vs. output) does not only depend on the DNA part itself – it is also relative to its positioning. The first plot shows the incompatibility of the two inverters when placed at short distance. For the two inverters to be connected in a row, output levels of the first inverter need to map to valid input levels in the next inverter. When placed in short distance this condition is not met - it would be virtually impossible to get two different Boolean states (0/1) out of the circuit. However, in the second plot, by moving the second inverter further apart, the interaction weakens and the repression becomes less efficient. In this new scenario, both parts do have aligned response functions and are now compatible.

This example was modelled *in silico* using the Synthetic Biology Open Language (SBOL) [12]. In SBOL, the two components of the system can be represented as sub-modules of a larger module representing the host context. However, this representation is limited to describing high-level topology. The authors therefore plan to extend the SBOL data model with the ability to capture spatial properties of genetic circuits. The SBOL Visual specification has also been updated with a set of best practices and new glyphs for representing the genomic context of a genetic construct.

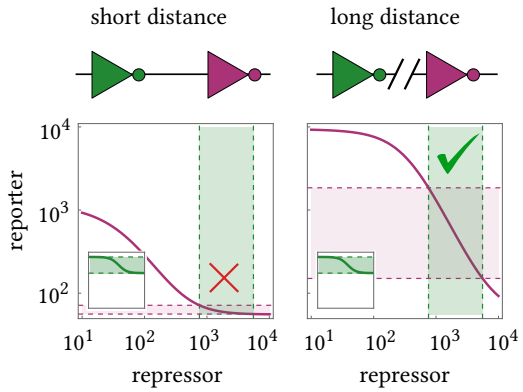


Figure 2: Simulated results of response function matching for different inter-genetic distances between two genetic NOT gates. At short distance (left plot) the output levels of the first gate (green) do not map to valid input levels of the next gate (purple). As a result, this combination will not perform properly. At long distance (right plot), however, the input levels of the second gate (purple) become valid. This combination performs as expected. There was no need to change circuit components, but only their location.

ACKNOWLEDGMENTS

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC) grants EP/J02175X/1 and EP/N031962/1 (James A. McLaughlin and Anil Wipat) and EP/R019002/1 (Alexander Wood and Angel Goni-Moreno).

REFERENCES

[1] Ángel Goñi-Moreno, Ilaria Benedetti, Juhyun Kim, and Víctor de Lorenzo. Deconvolution of Gene Expression Noise into Spatial Dynamics of Transcription Factor–Promoter Interplay. *ACS Synthetic Biology*, 8:acssynbio.6b00397, apr 2017.

[2] Akira Ishihama, Ayako Kori, Etsuko Koshio, Kayoko Yamada, Hiroto Maeda, Tomohiro Shimada, Hideki Makinoshima, Akira Iwata, and Nobuyuki Fujita. Intracellular concentrations of transcription factors in escherichia coli: 65 species with known regulatory functions. *Journal of Bacteriology*, pages JB–01579, 2014.

[3] Peter Dröge and Benno Müller-Hill. High local protein concentrations at promoters: strategies in prokaryotic and eukaryotic cells. *Bioessays*, 23(2):179–183, 2001.

[4] P B Warren and Pieter Rein ten Wolde. Statistical analysis of the spatial distribution of operons in the transcriptional regulation network of Escherichia coli. *Journal of Molecular Biology*, 342(5):1379–1390, 2004.

[5] Jacob A Englaender, J Andrew Jones, Brady F Cress, Thomas E Kuhlman, Robert J Linhardt, and Mattheos AG Koffas. Effect of genomic integration location on heterologous protein expression and metabolic engineering in e. coli. *ACS synthetic biology*, 6(4):710–720, 2017.

[6] Chaitanya Rastogi, H Tomas Rube, Judith F Kribelbauer, Justin Crocker, Ryan E Loker, Gabriella D Martini, Oleg Laptenko, William A Freed-Pastor, Carol Prives, David L Stern, Richard S Mann, and Harmen J Bussemaker. Accurate and sensitive quantification of protein-DNA binding affinity. *Proceedings of the National Academy of Sciences*, page 201714376, apr 2018.

[7] Johan Elf, Gene-Wei Li, and X Sunney Xie. Probing transcription factor dynamics at the single-molecule level in a living cell. *Science (New York, N.Y.)*, 316(5828):1191–4, may 2007.

[8] G W Li and X S Xie. Central dogma at the single-molecule level in living cells. *Nature*, 475(7356):308–315, 2011.

[9] Zeba Wunderlich and Leonid A Mirny. Spatial effects on the speed and reliability of protein-DNA search. *Nucleic Acids Research*, 36(11):3570–3578, jun 2008.

[10] Jennifer AN Brophy and Christopher A Voigt. Principles of genetic circuit design. *Nature methods*, 11(5):508, 2014.

[11] Alec A K Nielsen, Bryan S Der, Jonghyeon Shin, Prashant Vaidyanathan, Vanya Paralanov, Elizabeth A Strychalski, David Ross, Douglas Densmore, and Christopher A Voigt. Genetic circuit design automation. *Science (New York, N.Y.)*, 352(6281):aac7341, 2016.

[12] Michal Galdzicki, Kevin P Clancy, Ernst Oberortner, Matthew Pocock, Jacqueline Y Quinn, Cesar A Rodriguez, Nicholas Roehner, Mandy L Wilson, Laura Adam, J Christopher Anderson, et al. The synthetic biology open language (sbol) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology*, 32(6):545, 2014.

BLiSS: Black List Sequence Screening

Lisa Simirenko, Jan-Fang Cheng, Samuel Deutsch, and Nathan J. Hillson
DOE Joint Genome Institute, 2800 Mitchell Dr., Walnut Creek, CA 94598
Lawrence Berkeley National Lab, 1 Cyclotron Road, Berkeley CA 94720
{lsimirenko, jfcheng, sdeutsch, njhillson}@lbl.gov

1 INTRODUCTION

Concerns have been raised that individuals with malicious intent could exploit DNA synthesis technologies to acquire genetic elements from organisms and toxins that would otherwise be difficult to obtain [1–4]. In response to these concerns, in 2010 the U.S. Department of Health and Human Services (HHS) issued the *Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA* [5], which outlines recommendations for screening double-stranded DNA to ensure that existing regulations[1] and best practices are followed in addressing biosecurity concerns.

In 2015, the J. Craig Venter Institute (JCVI) released a report titled *DNA Synthesis and Biosecurity: Lessons Learned and Options for the Future* [6] describing the current status of biosecurity screening in the U.S. In this report, the authors identify the administrative costs of sequence screening, specifically the time taken by staff to review sequences that may be of concern, to be the costliest aspect of the process, and may be a barrier to adoption for smaller companies.

In accordance with the HHS guidance, the U.S. Department of Energy (DOE) Joint Genome Institute's (JGI) DNA Synthesis Science program has developed a DNA screening pipeline (BLiSS – Black List Sequence Screening) to screen all sequences that it synthesizes. As specified in the guidance, BLiSS detects "*sequences of concern*"[2] of at least 200 nucleotides in length on either DNA strand, and the resultant polypeptides from translations using the three alternative reading frames on each DNA strand (or six-frame translation). The construct sequences are aligned to the sequences in GenBank's non-redundant nucleotide and protein databases rather than a curated database, to ensure that it automatically adapts as new sequences are added to GenBank. A "Best Match" approach is used to determine whether a query sequence is unique to Select Agents or Toxins, or CCL-listed agents, toxins or genetic elements, and to minimize false positives from closely related organisms or highly conserved "house-keeping genes" which do not pose a biosecurity threat.

In order to save staff time and to facilitate the analysis of the screening results, we have added post-processing steps to detect false positives and a screen for viral sequences that may not be on any of the blacklists.

2 COMPUTATIONAL DETAILS

BLiSS consists of two components: 1) the analysis pipeline, written in Python with a MySQL database backend, and optimized to run on clusters managed by the DOE's National Energy Research Scientific Computing Center (NERSC) and 2) a web-based User Interface (UI).

In the first step, the analysis pipeline aligns all the given constructs to be synthesized to the GenBank *nt* and *nr* databases using *blastn* and *blastx* respectively. The meta-data describing the alignments is compared to a database of terms associated with the entities on the Select Agents and Toxins list and the CCL. If "hits" are found, the span of the construct that aligns to those "hits" is considered a putative sequence of concern.

If putative sequences of concern are detected, they are further analyzed using sliding 200bp windows. The best matches for each window are determined by individually aligning the window sequence and each of the sequences of the alignments that overlap the window. Best matches are assigned to the local alignments to the window with the highest product of the percent identity and length. For any given window there may be more than one "*best match*". The windows are then scored with a *Status* (Table 1).

Table 1: Status

Status	Definition
Passed	Not a best match to a sequence of concern
Failed	A best match to a sequence of concern on the list of Select Agents and Toxins
Controlled	A best match to a sequence of concern on the Commerce Control List (CCL)

When sequences are not "passed", they undergo post-processing steps to identify possible false positives. These are presented in the UI as "suggestions" and require user input to accept the suggestion. Currently, we have two criteria for suggesting that a "hit" may be a false positive:

1) The best matches of a sequence of concern are all in protein space and have <70% identity[3], and therefore is unlikely to be functionally the same protein.

[1] Select Agent Regulations (SAR) and, for international orders, the Export Administration Regulations (EAR)

[2] A "*sequence of concern*" is defined in the guidance as sequences "derived from or encoding" entities on the HHS/CDC's Select Agents or Toxins list, or agents on the Bureau of Industry and Security's Commerce Control List (CCL).

[3] The JCVI report [6] reported that International Gene Synthesis Consortium (IGSC) companies only consider those with >80% homology to be a "hit".

2) The best matches of a sequence of concern are found to be orthologs of those in the Benchmarking Universal Single-Copy Orthologs (BUSCO) [7] set, a collection of orthologous groups with near-universally distributed single copy genes in all species, and therefore unlikely to be involved in the pathogenicity of a particular organism.

If failed or controlled sequences are found, we ask the end user to verify the legitimacy of the end-use and consult with our local FBI weapons of mass destruction agents or LBNL's export control office.

The visualization of the screening results consists of a list of all the sequences in a given project. For each sequence there is a color-coded cartoon of the window spans and alignments that allows the user to drill down to see the best matches for each window, including the sequences, % identity, top hit rankings and other information (Figure 1).

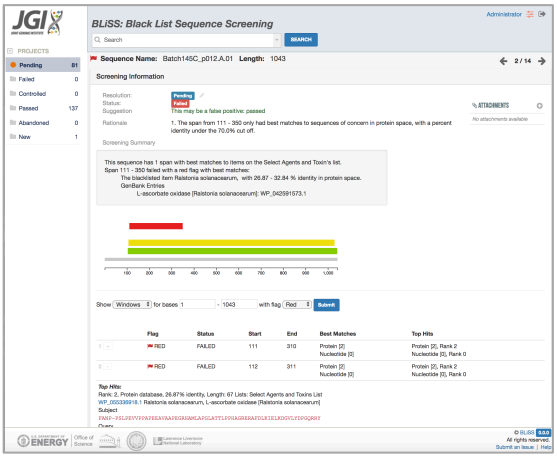


Figure 1: Visualization of screening results

3 RESULTS AND DISCUSSION

To date the JGI has screened 16,259 sequences (24.7 Million bp). Prior to post-processing, 0.8% Failed, 1.5% were Controlled, and 97.7% Passed (Figure 2). Of the 377 sequences that were not passed, 252 were found to be false positives by our post-processing. Of these, 17 were cleared using the BUSCO orthologs (Figure 3).

In the future, we plan to implement more screens for both false positives and false negatives, in an effort to give the user as much information as possible to save time and administrative costs. In this context, false negatives are hard to quantify. They would include genes that are from an organism on one of the blacklists, but not related to pathogenicity, or they could be miss-annotated sequences. For the later, we plan to flag sequences that have a high similarity to sequences of concern, even if they are not the "best match". The software will be made available to other credentialed researchers and synthetic DNA providers.

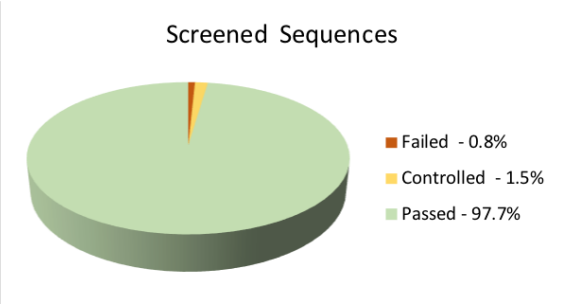


Figure 2: Results of JGI sequence screening to date

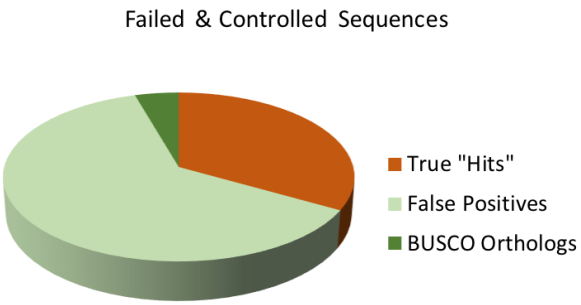


Figure 3: False positive detection

REFERENCES

[1] Jones R. 2005. Sequence Screening: In Working Papers for Synthetic Genomics: Risks and Benefits for Science and Society, pp. 1-16. Garfinkle MS, Endy D, Epstein GL, Friedman RM, editors, 2007

[2] Garfinkle MS, Endy D, Epstein GL, Friedman RM. 2007. Biodefense strategy, practice and science. Biosecurity and bioterrorism. 2007-12—01; 5.4: 359-62

[3] Tucker JB. 2010. Double-Edged DNA: Preventing the Misuse of Gene Synthesis. Issues in Science and technology 26, no.3 (Spring 2010)

[4] Tucker JB and Zilinskas RA. 2006. The Promise and Perils of Synthetic Biology. The New Atlantis, No 12 (Spring 2006), pp.25-45

[5] Department of Health and Human Services 2010. Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA https://www.phe.gov/Preparedness/legal/guidance/syndna/Documents/syndna-guidance.pdf

[6] Carter SR, Friedman RM. 2015. DNA Synthesis and BiSecurity: Lessons Learned and Options for The Future. J. Craig Venter Institute, La Jolla, California October 2015

[7] Simão FA, Waterhouse RM, Ioannidis P, Evgenia VK, and Evgeny MZ. 2105. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. Bioinformatics, published online June 9, 2015. doi: 10.1093/bioinformatics/btv351

Coordinating standards: digitalization of the Standard European Vector Architecture with the Synthetic Biology Open Language

Bryan Bartley
Raytheon BBN Technologies
United States
bryan.a.bartley@rytheon.com

James A. McLaughlin
Newcastle University
United Kingdom
j.a.mclaughlin@ncl.ac.uk

Göksel Mısırlı
Keele University
United Kingdom
g.misirli@keele.ac.uk

Victor de Lorenzo
National Center for Biotechnology
Spain
vdlorenzo@cnb.csic.es

Anil Wipat
Newcastle University
United Kingdom
anil.wipat@ncl.ac.uk

Angel Goñi-Moreno
Newcastle University
United Kingdom
angel.goni-moreno@ncl.ac.uk

MOTIVATION

The development of standards is a priority to the synthetic biology community. Standards facilitate the design-build-test-learn engineering lifecycle, since they enable the integration of inherently different tools and methods into coherent workflows. While existing standards target specific stages of the lifecycle, the coordination of inter-stage standardization efforts has received relatively little attention. This abstract describes how to use the Synthetic Biology Open Language (SBOL), a data standard for the design of genetic circuits , to represent the Standard European Vector Architecture (SEVA), a plasmid vector standard for de-/re-construction of bacterial function. The resulting data is stored in SEVAhub (sevahub.es), an instance of the SynBioHub repository. SEVAhub allows data specific to the build stage (i.e. SEVA) to be stored and related back to the appropriate designs. Information about both genetic circuits and their carrier vectors can therefore be fully standardized. We advocate the coordination of standards as a way to integrate and automate the design-build-test-learn lifecycle.

STANDARDIZATION OF PLASMIDS

To date, standardization efforts have focused on the genetic circuit itself and its parts and interactions. Important structural information, such as the plasmid vector carrying the circuit, is often sparse. The choice of the plasmid vector can have a major impact, not only in the building strategy, but also in circuit performance. The standardization of plasmid vectors and their components has a number of advantages. Firstly, the community benefits from sharing information using data formats that are well defined. This standardization fosters reusability and reproducibility. Secondly, sharing characterization data informs potential users about the performance of the plasmid, which will ultimately impact circuit behavior. Features like the copy number of a vector, for instance, are most relevant if the circuit at stake can be easily unbalanced against other components in different plasmids (or in the chromosome). It is therefore important to be able to select and optimize the most appropriate vector for a given synthetic genetic system.

The Standard European Vector Architecture (SEVA[1] [1]) establishes an unambiguous format for the structure of plasmid vectors in a modular fashion. The main functional (sub)components of each plasmid, namely antibiotic resistance, origin of replication, and cargo (where the circuit of interest is cloned into), are unequivocally located between specific rare restriction sites (Figure 1). These modules can be selected from a suite of DNA parts that have been minimized and edited following the SEVA standard. A fixed backbone, common to all SEVA vectors, enables the modules to easily be exchanged for specific vector functionality. To date, around 1500 SEVA vectors have been shipped to laboratories of >30 countries.

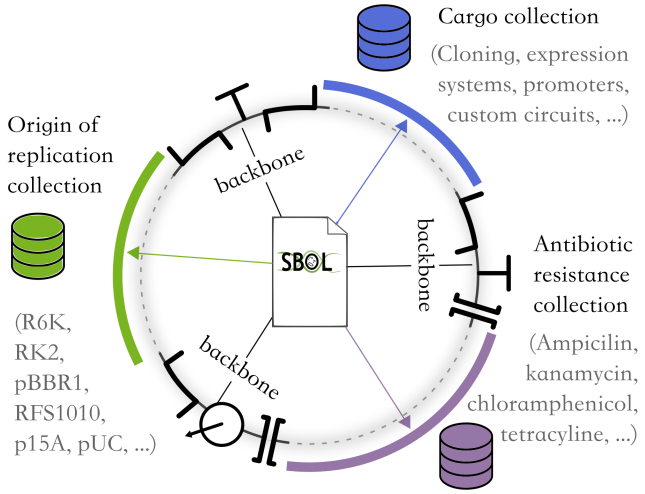


Figure 1: SBOL description of the SEVA standard. All SEVA plasmids share a backbone, which is hard-coded in SBOL as a template. Three main functional modules are located in unequivocal positions: antibiotic resistance, origin of replication and cargo. There is a suite of parts available for such modules, which allow users to choose specific plasmid vectors. These modules are organized as SBOL collections and are used to complete the backbone template and to generate SEVA plasmids in SBOL format.

SEVA-TO-SBOL CONVERSION

The goal of this work is to use the Synthetic Biology Open Language (SBOL [2]) to digitally formalize the molecular architecture defined by the SEVA format. This is accomplished using a modular approach

[1] http://seva.cnb.csic.es

following a combinatorial design strategy, so that all possible SEVA combinations are generated automatically.

Figure 1 summarizes this conversion. Three separate SBOL collections were generated, one for each of the main functional modules of the SEVA standard. A Python script was written that defines a **SEVAVector** class that extends an SBOL **ComponentDefinition**. The **SEVAVector** constructs a plasmid backbone using components from the SEVA collections and the custom cargo **ComponentDefinition** provided by the user to return a unique SEVA vector. For instance, we can fill in the template with the origin of replication R6K, ampicillin as the antibiotic marker and the multiple cloning site (MCS) as the selected cargo.

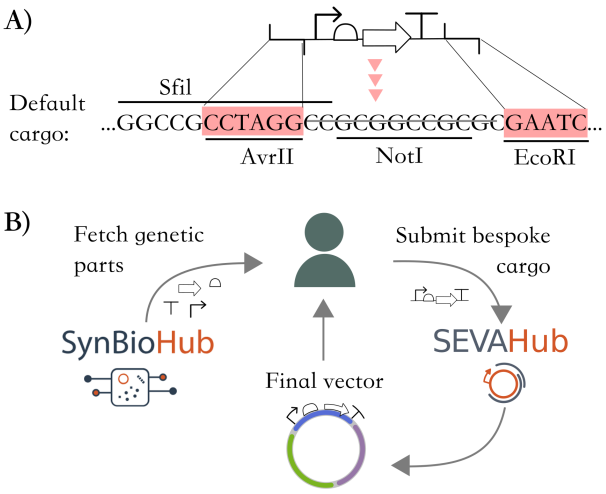


Figure 2: Usage of digitalized SEVA vectors. A) Detail of embedding a genetic circuit into the SEVA default cargo section. Restriction sites that flank the circuit denote position; after merging components, the final SEVA sequence is recalculated. B) Example workflow: a user fetches genetic parts from SynBioHub and builds a final construct using SEVAhub for plasmid description.

The major challenge from the SBOL standpoint was not handling the collections which represent modular plasmid parts, but representing the modification of the part of a plasmid that receives the cloned cargo, the multiple cloning site (MCS, termed the default cargo). The MCS is represented as a list of fourteen restriction sites that enables the user to select whatever combination is necessary to clone the genetic circuit of interest in between. The input to this method is two SBOL designs: one for the genetic circuit of interest and one for the MCS (Figure 2A). The output is a customized cargo. After replacing the DNA sequence between the two selected restriction sites by the genetic circuit, the positions within the MCS must be recalculated. Furthermore, it is possible that a restriction site stops being usable (as SfiI in the figure). This outcome requires the SBOL design to be re-defined in terms of its (sub)components.

SEVAHUB: A DESIGN REPOSITORY

In order to store the SBOL data generated from the SEVA-to-SBOL conversion, an instance of SynBioHub [3] (synbiohub.org) was installed under the domain sevahub.es – the SEVAhub repository. SynBioHub was recently developed as an open-source software project to facilitate the sharing of information about genetic designs, delivered in a standardized format using SBOL.

SEVAhub provides a graphical user interface that enables users to interact with SEVA data. Currently, there are four major collections in SEVAhub: a collection of all origins of replication, a collection of all antibiotic resistance genes, a collection of possible cargoes and a collection with all possible SEVA combinations. Users can download the SBOL data for single components (e.g. a kanamycin marker) or full plasmid descriptions. The plasmid of choice can be then assigned to a physical SEVA storage location (seva.cnb.csic.es).

A more advanced used of SEVAhub is shown in Figure 1, where a user can fetch genetic parts from another repository (in this case, from synbiohub.org), then build a genetic circuit with CAD tools and upload it to SEVAhub. The user is then able to generate a bespoke cargo (that was not in the collection before) and select this along with other functional modules to complete a SEVA vector, which can then be returned to the user. The information about the final genetic circuit embeds a detailed, hierarchical specification of the vector, not only the circuit itself. Currently, the construction of the SEVA vector requires the use of a programmatic software library [2]. In the future, this functionality will be integrated with the SEVAhub website to enable use by non-programmers.

Recently there have been a number of reports detailing the development of standards-enabled workflows [4] for synthetic biology. The coordination of the SEVA and SBOL standards will make an impact in next-generation pipelines, facilitating the automatic construction of vectors for a range of tasks. Furthermore, the adoption of SynBioHub and its instances, including SEVAhub, has the potential to address pressing issues in synthetic biology, such as the lack of sharing of design information – which plays a major part in the reproducibility crisis in the field [5].

ACKNOWLEDGMENTS

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC) grants EP/J02175X/1 and EP/N031962/1 (J.A.M. and AW) and EP/R019002/1 (AGM).

REFERENCES

[1] Esteban Martínez-García, Tomás Aparicio, Angel Goñi-Moreno, Sofía Fraile, and Víctor de Lorenzo. Seva 2.0: an update of the standard european vector architecture for de-/re-construction of bacterial functionalities. Nucleic acids research, 43(D1):D1183–D1189, 2014.

[2] Michal Galdzicki, Kevin P Clancy, Ernst Oberortner, Matthew Pocock, Jacqueline Y Quinn, Cesar A Rodriguez, Nicholas Roehner, Mandy L Wilson, Laura Adam, J Christopher Anderson, et al. The synthetic biology open language (sbol) provides a community standard for communicating designs in synthetic biology. Nature biotechnology, 32(6):545, 2014.

[3] James Alastair McLaughlin, Chris J Myers, Zach Zundel, Goksel Misirli, Michael Zhang, Irina Dana Ofiteru, Angel Goñi Moreno, and Anil Wipat. Synbiohub: A standards-enabled design repository for synthetic biology. ACS synthetic biology, 2018.

[4] Angel Goñi-Moreno, Marta Carcajona, Juhyun Kim, Esteban Martínez-García, Martyn Amos, and Víctor de Lorenzo. An implementation-focused bio/algorithmic workflow for synthetic biology. ACS synthetic biology, 5(10):1127–1135, 2016.

[5] Jean Peccoud, J Christopher Anderson, Deepak Chandran, Douglas Densmore, Michal Galdzicki, Matthew W Lux, Cesar A Rodriguez, Guy-Bart Stan, and Herbert M Sauro. Essential information for synthetic dna sequences. Nature biotechnology, 29(1):22, 2011.

[2] https://github.com/SynBioDex/SEVA

DAMP Lab North: Using Formal Representations of Protocols for the Specify-Design-Build-Test Cycle in a Prototypical Software-Driven Laboratory

Nicholas Emery
Boston University
Boston, MA, USA
emernic@bu.edu

Marilene Pavan
Boston University
Boston, MA, USA
mapavan@bu.edu

Douglas Densmore
Boston University
Boston, MA, USA
dougd@bu.edu

ABSTRACT

The mission of the DAMP Lab North[1] is to execute small to medium scale projects, constructing novel biological systems using formal representations of protocols and experiments for the specify-design-build-test cycle. The DAMP Lab North uses a hybrid execution model where jobs are completed using both low-cost automation systems and technicians following highly-standardized manual protocols specified through code. We demonstrate this approach by developing and testing a system of such protocols for molecular cloning. These protocols are implemented in Aquarium[2], an open source LIMS system. The molecular cloning workflow is currently available as a service to internal and external researchers, with additional services planned for implementation soon. Data on various aspects of laboratory operations are collected and organized automatically to improve processes.

KEYWORDS

Synthetic Biology, Automation, LIMS, Reproducibility

INTRODUCTION

Advances in software and robotics will help to free researchers from repetitive manual tasks, make their data easier to store and query, and ultimately lead to cheaper, more efficient, and more reproducible science [1]. However, the parameters affecting the success or failure of even extremely common laboratory protocols[3] are still poorly characterized, and the protocols themselves are often insufficiently-specified or vary drastically between researchers. Therefore, successfully drafting and carrying out an optimal experimental plan still relies heavily on experimental expertise and intuition. This ultimately leads to increased costs and a lack of reproducibility, a problem that has gained increasing recognition in life sciences [2]. The DAMP Lab North (DLN) uses formal representations of protocols to complete experimental plans in a prototypical software-driven laboratory using a fully open-source software stack and off-the-shelf hardware. The ultimate goal of the DLN is to collect and organize data generated during experimental execution and use this data to improve protocols and laboratory processes, closing the loop to enable continuous improvements in the cost, efficacy, and reproducibility of biological research.

DAMP LAB NORTH PROCESS

The primary focus of the DLN is on process development and the integration of existing tools rather than the invention of specific novel hardware or software. Therefore, the DLN makes use of off-the-shelf hardware and open-source software wherever possible, allowing a wide audience to use or adapt the processes we characterize.

The DLN uses Aquarium, an open-source LIMS system, as a general framework for organizing protocols, plans[4], and inventory. Plans can be submitted via the DLN web GUI or API. Both interfaces are standardized across all laboratories using Aquarium LIMS, which will allow for the development of generalized upstream planning software, such as Puppeteer[5], that can submit experimental plans to either the DLN or other laboratories. Because Aquarium plans reference existing protocols, end-users are abstracted away from low-level actions such as liquid transfers; this is in contrast to other approaches that require the user to specify all atomic actions[6][7]. After plans are received by the DLN, the operations[8] within each plan are batched into jobs[9] by the lab manager, subject to scheduling rules specified for each protocol and precedence constraints of the plan.

[1] https://www.damplab.org/laboratories
[2] https://github.com/klavinslab/aquarium
[3] We define a protocol as a description of a laboratory process (e.g. PCR) with defined inputs and outputs that contains all necessary execution logic and actionable commands (at any level of abstraction). For example, protocols in the DLN North capture execution logic using Ruby and contain commands in Krill (Aquarium's language for human-readable commands) and a Ruby library of OT-2 robotic commands.
[4] A plan is a user-submitted collection of connected "operations" that reference existing protocols. Precedence constraints are inferred by input-output connections between operations.
[5] http://cidarlab.org/puppeteer/
[6] https://www.transcriptic.com/
[7] https://docs.antha.com/
[8] Each operation in a plan contains user-specified inputs/outputs and reference an existing protocol.
[9] A job is a group of operations of the same type, though not necessarily from the same plan, which are executed together by a robot or technician.

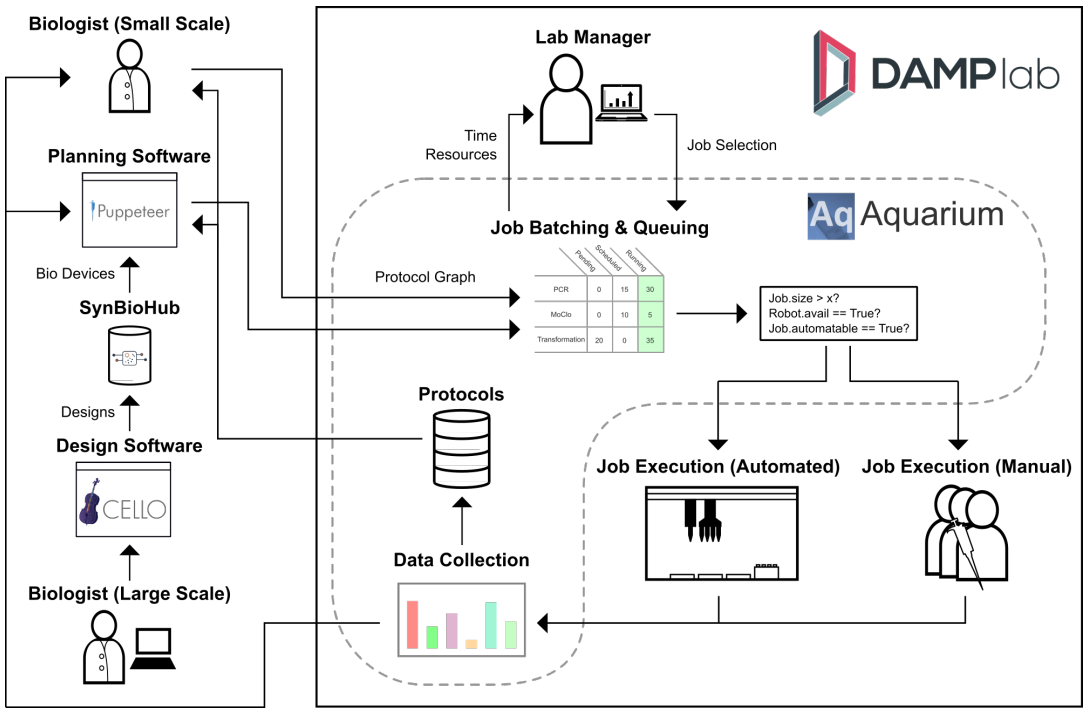


Figure 1: Biologists submit plans (protocol graph) via design and planning software or directly via the Aquarium web GUI. Operations within plans are batched into jobs and executed. Data is used to improve protocols and to inform researchers and planning software.

Jobs are executed manually, as in existing labs[10] that use Aquarium, or robotically, using the OpenTrons OT1 and OT2 liquid handling platforms. Both approaches have novel aspects in their implementation: 1.) manual protocols developed in the DLN capture all calculations and protocol execution decisions at the software level, displaying only unambiguous atomic commands to technicians, 2.) protocols for the OpenTrons robots are generated dynamically to account for differences in job size and other parameters, 3.) manual instructions for loading, unloading, and configuring the OpenTrons robots are generated automatically and displayed to technicians, 4.) both manual and automated execution methods are characterized for many protocols, allowing the lab manager to decide on-the-fly between these approaches. The coexistence of both execution methods makes the system robust to hardware failures and allows difficult-to-automate protocols to be rapidly integrated while still allowing for automation of the most heavily used protocols.

Data collected through Aquarium is automatically analyzed via tools currently under development at the DLN, allowing for the identification of key areas for process improvement. Both the data and the analysis tools will be made available for the benefit of the research community.

[10]http://www.uwbiofab.org

CONCLUSION AND FUTURE DIRECTIONS

The DLN uses formalized protocols and a combination of automated and manual execution to characterize key experimental processes for synthetic biology. Currently, 30 protocols related to molecular cloning are available to researchers, with additional services planned for implementation in the near future. Through this effort, we will: 1) gain insights into parameters affecting the performance of various experimental processes, which will be useful in scaling up synthetic biology, 2) develop best practices for the establishment and operation of a highly-standardized, data-driven research lab, which will serve as a useful template across scientific disciplines, 3) provide valuable experimental services to collaborating synthetic biology laboratories and institutions.

REFERENCES

[1] Erika Check Hayden. 2014. The automated lab. *Nature* 516, 7529 (dec 2014), 131–132. DOI:http://dx.doi.org/10.1038/516131a

[2] Leonard P. Freedman, Iain M. Cockburn, and Timothy S. Simcoe. 2015. The Economics of Reproducibility in Preclinical Research. *PLOS Biology* 13, 6 (jun 2015), e1002165. DOI:http://dx.doi.org/10.1371/journal.pbio.1002165

Automating Functional Enzyme Screening & Characterization

Luis Ortiz
Molecular Biology, Cell Biology & Biochemistry
Boston University
610 Commonwealth Ave
Boston, MA 02215
lortiz15@bu.edu

Ali Lashkaripour
Biomedical Engineering
Boston University
610 Commonwealth Ave
Boston, MA 02215
lashkari@bu.edu

Douglas Densmore
Electrical & Computer Engineering
Boston University
610 Commonwealth Ave
Boston, MA 02215
dougd@bu.edu

ABSTRACT
Microfluidics continue to gain traction as an inexpensive alternative to standard multi-well plate-based, and flow cytometry-based, assay platforms. These devices are especially useful for the types of ultra-high throughput screens needed for enzyme discovery applications where large numbers (>10^6) of unique samples must be screened rapidly[1]. Coupled with cell-free protein synthesis[2], microfluidics are being used to identify novel enzymes useful for a variety of applications with unprecedented speed. However, these devices are typically produced using PDMS, and require considerable infrastructure and artisanal skill to fabricate, limiting their accessibility. Likewise, enzyme hits obtained from a screen are often validated manually and would benefit from automation of downstream validation processes. To address these limitations, we propose a workflow which leverages software tools to automate the rapid design and fabrication of low-cost polycarbonate microfluidic devices for use as high-throughput screening platforms for enzyme discovery, as well as an automated DNA assembly tool to streamline validation of screening candidates. Using this workflow, we aim to identify novel oxidoreductase enzymes from environmental metagenomic DNA libraries, for use in electrochemical biosensors.

Keywords
Synthetic biology; cell-free; screening; microfluidics; CAD

1. INTRODUCTION
Environmental microbes possess an incredibly diverse set of enzymes and small molecules that they produce to thrive and interact with their environment. This resource can be tapped, using high-throughput functional screens, to discover novel biomolecules with numerous applications, from biosensing to biomanufacturing. These types of screens have relied heavily on lower throughput microtiter plate-based assays, as well as higher throughput flow cytometry, but microfluidics are emerging as a cheaper, faster, ultra-high throughput alternative[3]. These devices are often designed ad-hoc using graphic design software such as Adobe Illustrator, which does not allow easy parameterization of device components or iteration of designs. These devices are also typically fabricated using polydimethylsiloxane (PDMS), which requires specialized equipment and personnel training, limiting its accessibility in many academic labs. However, emerging software tools which automate the design of microfluidic devices from a high-level functional specification, as well as fabrication of devices using CNC-milled geometries in polycarbonate, are beginning to address this issue.

A second bottleneck in these enzyme screens lies in the downstream validation of positive hits. Putative enzymes identified in a screen are usually cloned into expression vectors, transformed into expression hosts such as *E. coli* or yeast, and used to produce and purify the protein for downstream analysis. Depending on the number of positive hits identified in a screen, this can lead to a non-trivial number of protein expression vectors that need to be cloned, especially when trying to assess the optimal position of the affinity tag itself (N- or C-term). Modular DNA assembly strategies like MoClo[4,5], together with software tools like mocloassembly.com[6] offer a scalable and automatable DNA assembly workflow that can address the need to generate combinatorial protein expression vectors via liquid handling robots. Coupled with cell-free protein synthesis, a larger number of expression vector variants can be assembled and tested with unprecedented speed.

2. MICROFLUIDIC DESIGN & FABRICATION
Microfluidic device design has largely been artisanal, with researchers often resorting to graphical design software tools to manually draw out microfluidic device geometries. This process makes it difficult to iterate on new designs since individual components of the device design must be changed manually and non-parametrically. Our software tools, however, allow us to specify high level microfluidic functionality parametrically, and fabricate devices via CNC milling in thermoplastics on the order of hours[7]. This is in stark contrast to more traditional PDMS-based device fabrication which can take days. This workflow has enabled us to design, fabricate, and test many iterations of devices to rapidly identify key geometry parameters important for functional screening of enzyme libraries. These parameters dictate processes like droplet generation rate & size, droplet merging & splitting, on-chip PCR, cell-free protein expression, and fluorescence/colorimetric-based droplet sorting.

3. CELL-FREE PROTEIN SYNTHESIS
Cell-free (CF) protein synthesis is experiencing a new renaissance as a versatile, fast, and inexpensive biological prototyping platform[2]. These CF mixes typically use cellular extracts from various organisms which provide the machinery necessary for *in-vitro* transcription and translation from a DNA template. This allows researchers to simply add DNA circuits which encode for proteins of interest, to achieve high amounts of expressed protein. This circumvents several steps in more traditional protein expression workflows where DNA for expression vectors must first be assembled, then transformed into the expression organism of choice. With CF mixes, even linear DNA fragments generated via PCR can be used as protein expression templates, further shortening prototyping time[8].

For enzyme screening we are using *E. coli* cell extracts to express protein within water-in-oil droplets generated in our microfluidic

device. At the initial point of droplet generation these droplets encapsulate single members of a microbial metagenomic DNA library which enables screening of individual members in high throughput. Screening reagents and enzyme substrates are later added to each droplet via droplet merging, and droplets which exhibit a fluorescent/colorimetric signal above a predefined threshold value are isolated for downstream sequencing. We are particularly interested in identifying novel oxidoreductase enzymes since they couple well with electrochemical sensors, similar to the ubiquitous blood glucose meter. This class of enzymes catalyzes the transfer of electrons from one molecule to another, oftentimes generating hydrogen peroxide as a byproduct of catalysis of an analyte of interest. To monitor enzyme activity in droplets we will use the fluorometric chemical probe Amplex UltraRed which is oxidized to a fluorescent product in the presence of hydrogen peroxide. Droplets exhibiting a fluorescence signal intensity above a predetermined threshold, presumably as a consequence of analyte degradation, will be sorted for downstream sequencing and validation.

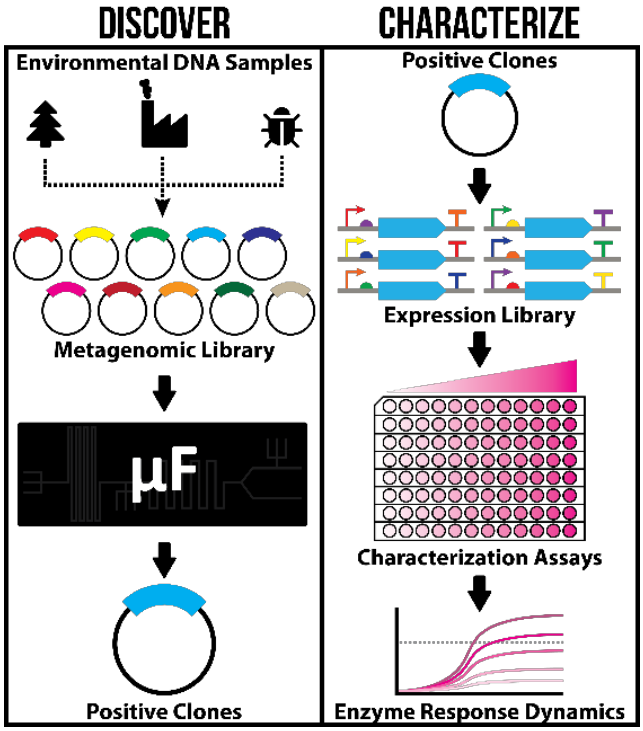


Figure 1. Automated microfluidic design & DNA assembly for rapid enzyme screening and downstream validation. [Discover] Using our software tools we can rapidly iterate across the microfluidic design space to generate variants of devices via CNC-milling. These devices take complex DNA libraries as an input, and selects droplets which respond to a molecule of interest. [Characterize] Positive hits from the screen are then sequenced, cloned into a MoClo destination vector, and expression libraries are automatically generated by liquid-handling robots using the software tool mocloassembly.com.

4. AUTOMATED DNA ASSEMBLY FOR ENZYME CHARACTERIZATION

In high-throughput screens, enzyme hits must be validated to test substrate specificity, and identify optimal parameters such as temperature, pH, and catalytic rate. This typically requires the generation of new DNA circuits for expression and purification of candidate enzymes, often via affinity chromatography. This can quickly become laborious as the number of hits from a screen increase, and the optimal position (N- or C-term) for a genetically encoded affinity tag is unknown. To address this scaling need, we are using Modular Cloning (MoClo) and the software tool mocloassembly.com to automatically build various instances of protein expression circuits from these screens via liquid handling robots. These circuits are then used in new CF reactions to rapidly generate protein for validation tests in an effort to quickly characterize new enzymes obtained from the functional screen.

5. ACKNOWLEDGMENTS

We would like to thank the NSF Living Computing Project Award #1522074 and NSF CAREER Award #1253856 for funding.

6. REFERENCES

[1] Agresti, J. J. *et al*. Ultrahigh-throughput screening in drop-based microfluidics for directed evolution. *Proc Natl Acad Sci U S A*. **107** (9), 4004-4009, doi:10.1073/pnas.0910781107, (2010).

[2] Garamella, J., Marshall, R., Rustad, M. & Noireaux, V. The All E. coli TX-TL Toolbox 2.0: A Platform for Cell-Free Synthetic Biology. *ACS Synth Biol*. **5** (4), 344-355, doi:10.1021/acssynbio.5b00296, (2016).

[3] Bunzel, H. A., Garrabou, X., Pott, M. & Hilvert, D. Speeding up enzyme discovery and engineering with ultrahigh-throughput methods. *Curr Opin Struct Biol*. **48** 149-156, doi:10.1016/j.sbi.2017.12.010, (2018).

[4] Iverson, S. V., Haddock, T. L., Beal, J. & Densmore, D. M. CIDAR MoClo: Improved MoClo Assembly Standard and New E. coli Part Library Enable Rapid Combinatorial Design for Synthetic and Traditional Biology. *ACS Synth Biol*. **5** (1), 99-103, doi:10.1021/acssynbio.5b00124, (2016).

[5] Werner, S., Engler, C., Weber, E., Gruetzner, R. & Marillonnet, S. Fast track assembly of multigene constructs using Golden Gate cloning and the MoClo system. *Bioeng Bugs*. **3** (1), 38-43, doi:10.1371/journal.pone.001676510.4161/bbug.3.1.18223, (2012).

[6] Ortiz, L., Pavan, M., McCarthy, L., Timmons, J. & Densmore, D. M. Automated Robotic Liquid Handling Assembly of Modular DNA Devices. *JoVE*. (130), e54703, doi:doi:10.3791/54703, (2017).

[7] A. Lashkaripour, R. S., and D. Densmore. Desktop micromilled microfluidics. *Microfluidics and Nanofluidics*. **Vol. 22** (Iss. 3), p. 31, doi:10.1007/s10404-018-2048-2, (2018).

[8] Schinn, S. M., Broadbent, A., Bradley, W. T. & Bundy, B. C. Protein synthesis directly from PCR: progress and applications of cell-free protein synthesis with linear DNA. *N Biotechnol*. **33** (4), 480-487, doi:10.1016/j.nbt.2016.04.002, (2016).

Specifying Combinatorial Designs with the Synthetic Biology Open Language (SBOL)

Nicholas Roehner[1], Bryan Bartley[1], Jacob Beal[1], James McLaughlin [2], Matthew Pocock[3], Michael Zhang[4], Zach Zundel[4], Chris Myers[4], Anil Wipat[2]
[1]Raytheon BBN Technologies, [2]Newcastle University, [3]Turing Ate My Hamster, Ltd., [4]University of Utah
nicholas.roehner@raytheon.com

1 INTRODUCTION

During the last decade, new technologies have been developed for the combinatorial assembly of genetic parts [8, 9], enabling synthetic biologists to more readily generate libraries of genetic construct variants. These types of combinatorial libraries can play an important role in genetic design by allowing designers to explore the impact of part choice, order, and orientation on construct behavior. In order to support the design of such libraries, new tools and formalisms have been developed to enable the specification, permutation, and sampling of combinatorial genetic design spaces [1, 2]. In turn, these formalisms have given rise to the need for a standard representation of combinatorial genetic designs in order to enable sharing of such designs between tools and laboratories and to simplify human and machine reasoning over them.

As a basis for this representation, we have chosen the Synthetic Biology Open Language (SBOL), an existing community standard for representing both structural and functional aspects of genetic designs [4, 7]. SBOL has support for hierarchical design, modular composition, and partial specification, making it a natural fit for representing combinatorial design templates and variables. Accordingly, we have developed an extension of SBOL to represent combinatorial designs, and we have incorporated this extension into the SBOL 2.2 specification [3] and SBOL software libraries (www.sbolstandard.org/libsbol). Here we briefly summarize the data model for this extension and discuss its application in two example use cases: a library of pathway variants to optimize enzyme expression [5], and a library of genetic circuit variants to optimize logic gate function [6, 9].

2 REPRESENTING COMBINATORIAL DESIGN

Building on the core data model of SBOL, the representation of combinatorial design is a relatively lightweight extension. Namely, its representational semantics involve the specification of a design template and any constraints on its structure, the variable portions of the template and their cardinality, and the variants or values that these variables can assume. SBOL does not require any particular algorithm or data structure to be used in enumerating designs from a combinatorial specification, but provides rules and best practices to validate whether these designs are a correct realization of their specification.

There are two classes in the new SBOL 2.2 combinatorial data model: the CombinatorialDerivation class and the VariableComponent class (Figure 1). The CombinatorialDerivation class is used to specify a template for a library of combinatorial designs and to link that template to a collection of variables and values that will fill in the template to form specific combinations. The template is defined

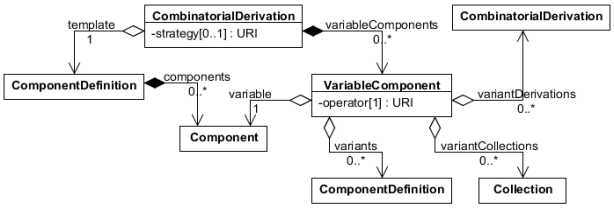


Figure 1: Combinatorial designs can be specified in SBOL 2.2 using two new classes: CombinatorialDerivation and VariableComponent.

using a ComponentDefinition: ComponentDefinition is a base class of SBOL used to specify the structure of a biopolymer in a modular, hierarchical manner, along with constraints on this structure. For instance, the ComponentDefinition for an abstract transcriptional unit (TU) would likely contain sub-Component objects for a promoter, coding sequence (CDS), and terminator without sequences and a set of SequenceConstraint objects to assert their relative ordering and orientations. The CombinatorialDerivation class can also be used to broadly recommend how many individual designs to derive from the template by setting its strategy property. At present, two strategy values are defined: either exhaustive enumeration of every possible design or sampling an unspecified subset.

The other class, VariableComponent, is used to specify the way in which a CombinatorialDerivation template is filled in to create fully instantiated designs. Each instance of the VariableComponent class specifies a set of available ComponentDefinition variants that can define a Component from the template. These variants can be aggregated individually or as part of an SBOL Collection, or can be derived in accordance with another CombinatorialDerivation, enabling the specification of a hierarchical combinatorial design. The operator property then specifies how many Component objects are expected to be derived from the template Component (one, zero-or-one, zero-or-more, or one-or-more). A more detailed description of the CombinatorialDerivation and VariableComponent classes can be found in the SBOL 2.2 technical specification [3].

3 EXAMPLE USE CASES

Use Case: Pathway Design. Figure 2 demonstrates how SBOL can be used to encode the combinatorial design of a library of 3,125 violacein pathway variants originally designed by the Dueber lab [5]. The SBOL representation consists of a two-level hierarchy of ComponentDefinition and CombinatorialDerivation objects. The root ComponentDefinition is a template that specifies the complete

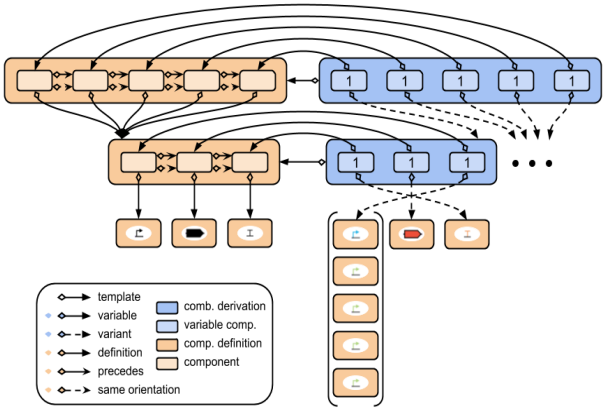


Figure 2: Representation of violacein pathway combinatorial design using SBOL.

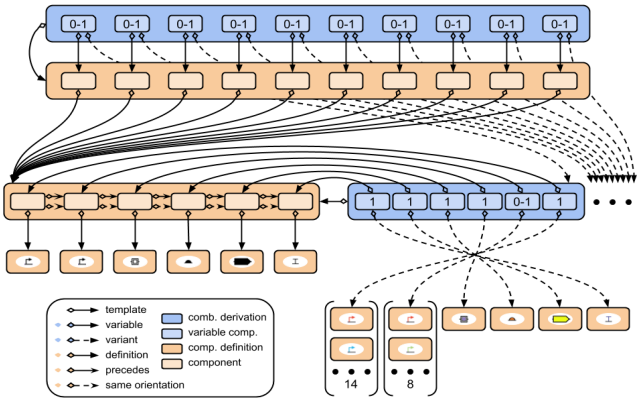


Figure 3: Representation of Cello circuit combinatorial design using SBOL.

ordering of five generic TUs, each defined by the same ComponentDefinition containing a promoter followed by a CDS and a terminator, all with the same orientation. The root CombinatorialDerivation then specifies that each of the five TUs in the template should be filled in with one of five possible TUs with different promoters as specified by a leaf CombinatorialDerivation. Each leaf CombinatorialDerivation refers to the same set of five promoter variants but refers to a different enzyme CDS in the violacein pathway.

Use Case: Genetic Circuit Design. Figure 3 demonstrates how SBOL can be used to encode the combinatorial design of all 10^{30} genetic circuit variants that can be constructed from the Cello gate NOR/NOT gate library. The key differences between this combinatorial design and that of the violacein pathway are that its root CombinatorialDerivation does not specify the relative order or orientation of any of its ten generic TUs, nor does it require that each of these TUs be filled in (because each VariableComponent has a zero-or-one operator). Consequently, the circuit derived from this combinatorial design can contain any number of TUs up to ten, and these TUs can have any ordering or orientation. In addition, each leaf CombinatorialDerivation has a single zero-or-one VariableComponent corresponding to the first promoter in the template TU ComponentDefinition, thus capturing the fact that each derived TU can have NOT or NOR logic (one promoter or two promoters).

4 DISCUSSION AND CONTRIBUTIONS

Currently, SBOL's representation of combinatorial design is equivalent in expressive power to a regular language. Though not demonstrated by these use cases, SBOL can be used to represent design patterns in which a particular component or motif is repeated an indefinite number of times. For example, this could be used to represent the design of a promoter with a variable number of operator sites. Should the need arise to represent palindromic design patterns, such as with a context-free language, SBOL can be extended with additional types of constraints to assert that the same number of components must be derived from different parts of the template.

Many key cases of combinatorial library design can be represented using SBOL with the new combinatorial design extension, ranging from existing industrial applications in optimizing biosynthetic pathways to current research in controlling biological systems. This improves over prior representations by integrating combinatorial design with hierarchical, ontology-supported representation, allowing unambiguous reasoning about complete designs, as well as their relationship to information sources, experimental products, and other designs. We thus anticipate that SBOL representation of combinatorial design will support improved tooling and workflows, facilitating better reuse and attribution of designs, faster engineering of circuits and components, and novel applications across many domains of synthetic biology.

ACKNOWLEDGMENTS

This work was supported by the DARPA Living Foundries award HR0011-15-C-0084. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

REFERENCES

[1] S. P. Bhatia, M. J. Smanski, C. A. Voigt, and D. M. Densmore. Genetic design via combinatorial constraint specification. *ACS Synth. Biol.*, 6(11):2130–2135, 2017.
[2] L. Bilitchenko et al. Eugene–a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS One*, 6(4):e18882, 2011.
[3] R. Cox et al. Synthetic Biology Open Language (SBOL) version 2.2.0. *Journal of Integrative Bioinformatics*, 15(1), 2018.
[4] M. Galdzicki et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat. Biotechnol.*, 32(6):545–550, 2014.
[5] M. E. Lee, A. Aswani, A. S. Han, C. J. Tomlin, and J. E. Dueber. Expression-level optimization of a multi-enzyme pathway in the absence of a high-throughput assay. *Nucleic Acids Res.*, 41(22):10668–10678, 2013.
[6] A. A. K. Nielsen. Genetic circuit design automation. *Science*, 352(6281):aac7341, 2016.
[7] N. Roehner et al. Sharing structure and function in biological design with SBOL 2.0. *ACS Synth. Biol.*, 5(6):498–506, 2016.
[8] E. Weber, C. Engler, R. Gruetzner, S. Werner, and S. Marillonnet. A modular cloning system for standardized assembly of multigene constructs. *PLoS One*, 6(2):e16765, 2011.
[9] L. B. A. Woodruff et al. Registry in a tube: multiplexed pools of retrievable parts for genetic design space exploration. *Nucleic Acids Res.*, 45(3):1553–1565, 2017.

The Desktop Biofoundry: Biodesign Manufacturing Automation in a Cloud-driven Digital Microfluidics Platform with Integrated Temperature Control, Optical Sensing and Purification

Sabrina Zaini
Digi.Bio
Overhoeksplein 2
+31653803135
sabrina@digi.bio

Frido Emans
Digi.Bio
Overhoeksplein 2
+31641156058
frido@digi.bio

Federico Muffatto
Digi.Bio
Overhoeksplein 2
+31642871563
federico@digi.bio

1. INTRODUCTION

Substantial efforts are underway in the development of Biodesign Automation to miniaturize and automate experimental pipelines and scale up research throughput, and new interest is rising for microfluidic technologies to fill this gap. An added challenge is to make biology programmable and hands-free, while simultaneously reducing its socioecological footprint. Digital Microfluidics is a technology that has the potential to satisfy these requirements. However, despite extensive and promising research, the field still lacks off-the-shelf solutions that integrates with other general software, hardware and wetware tools in the lab. workflow. This has been attributed to, among others, a lack of standardization. Researchers have sought to tackle this issue by developing, amongst others, a standard exchange format [1], composable and modular computer-aided design (CAD) tools [2, 3], and computer-aided manufacturing (CAM) tools which integrate various process functions [4]. This work introduces the Desktop BioFoundry, a CAM tool based on cloud-driven digital microfluidics aimed at streamlining research at different hierarchies in one seamless and user-friendly process.

2. DIGITAL MICROFLUIDICS

Digital Microfluidics (DMF) refers to a fluid handling technology which allows for the discrete manipulation of fluids using electrical signals applied over a planar electrode array. See Choi et. al. [5] for technical details. Apart from noted benefits of microfluidics technologies [6], DMF additionally offers scalability and dynamic reconfigurability [7]. Although DMF has found several applications [4, 5], several improvements are necessary to drive the technology towards commercialization. These include system integration and interfacing with other laboratory formats and devices, maintaining temperature control, lack of compatible detector technology and lack of molecular separation. Additionally, the development of DMF technologies have been hampered by the absence of standard commercial components, leading to devices with highly specific applications [4].

3. THE DESKTOP BIOFOUNDRY

The Desktop Biofoundry (DB) is a low-to-medium throughput device which consists of the following components:

3.1 Hardware

The core of the platform is the microfluidics cartridge. The cartridges are printed circuit boards encapsulated by two plates, which protect droplets within from contamination and evaporation. Multiple samples can run in parallel in one cartridge, allowing many different batches to be run at the same time. A desktop device, which hosts the cartridge, also hosts actuators for microfluidics electrode signals, temperature control, purification and optical sensing. An integrated camera in the device allows for real time droplet tracking, volume quantification and real time monitoring of the experiment.

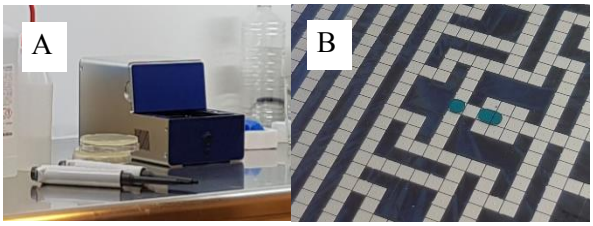


Figure 1 A. The Desktop Biofoundry B. Biochip with moving droplets

3.2 Software

The DB is supported by a cloud-based software which handles device operation and data collection and management. The cloud software connects to the user through a web browser, where the researcher can design, modify, run and share her own protocols in an intuitive graphical interface, or select one from community shared protocols. Once the protocol has been started, it will be run autonomously by the cloud software. This process can be monitored in real-time or afterwards. The software will send push messages to notify the user of any anomalies or errors along the way or when the protocol is finished.

Experimental data is accessed through the same interface, where it can be shared and referenced using a unique URL. The experimental protocol and data can be published and easily replicated by peers using the same setup for verification of results. The software allows for easy integration using an open API and webhooks with other lab software tools such as analytical software or an online lab journal.

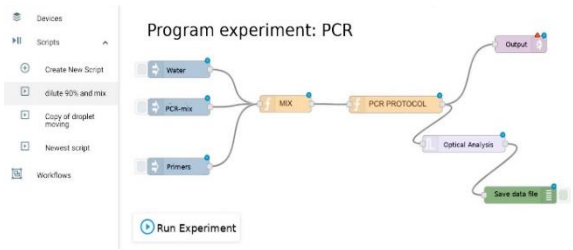


Figure 2 Drag-and-drop platform control and protocol design interface

4. ACCELERATING SYNBIO WITH THE DESKTOP BIOFOUNDRY

As a CAM tool, the DB combines several process units within one device, thereby enabling a greater degree of configurability, automation and parallelization of experiments. This allows for the establishment of more complex workflows required for, e.g., mass scale combinatorial DNA assembly [8]. The incorporation of optics features allows for the usage of machine vision, for example in the tracking of fluorescent probes [9].

Cloud storage of generated data and associated protocols offer a repository from which researchers can collaboratively derive and design new microfluidics protocols. Further empowering the platform is the possibility for integration with available CAD tools, e.g. SBOLDesigner [10], DNAplotlib [11], Cello [12] and RavenCAD. Furthermore, affordability of the DB cartridge and device makes this technology widely accessible to many synbio researchers and laboratories. These elements combined make for a rapid end-to-end platform for mass scale synthetic biology research.

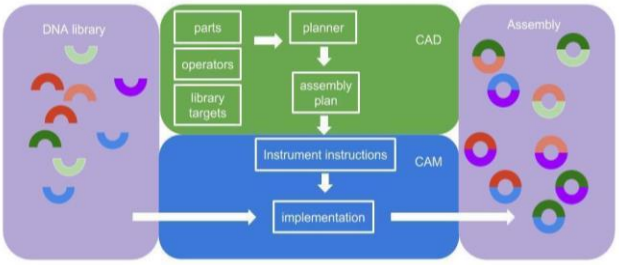


Figure 3 Conceptual diagram of a CAD-CAM integrated workflow

5. MOVING FORWARD

The integration of workflows at different hierarchies is imperative for rapid, large-scale automation of synthetic biology pipelines. The features of the DB strongly support a streamlined pipeline and can be further developed in order to more accurately address the needs of synthetic biology research. For example, identification and integration of other important workflows in the synbio hierarchy, in-line sample preparation and the development of biocompatible cartridges.

6. CONCLUSION

By combining synthetic biology workflows from different hierarchies in one affordable platform, the Desktop Biofoundry cuts short experimental validation of assembly strategies. The integration of temperature control, optical sensing and magnetic purification in the device, together with plug-and-play modularity with CAD tools supports automated, efficient and data driven implementation of complex workflows.

7. REFERENCES

[1] Galzdicki, M., et al. 2014. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology 32*, 6, 545. DOI = https://doi.org/10.1038/nbt.2891

[2] Misirli, G., Hallinan, J., and Wipat, A. 2014. Composable modular models for synthetic biology. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 11, 3, 22. DOI= https://doi.org/10.1145/2631921

[3] Sanka, R. Subacius, K., Kapadia, P., Densmore, D., McCormack, S., and Asthana, A. 2017. Fluigi Cloud – A cloud CAD platform for microfluidics. Poster presented at the 9th International Workshop on Bio-Design Automation (IWBDA), August 2017.

[4] Fair, R.B. 2007. Digital microfluidics: is true lab-on-a-chip possible? *Microfluidics and Nanofluidics,* 3, 3. 245-281. DOI= https://doi.org/10.1007/s10404-007-0161-8

[5] Choi, K. Ng, A.H., Fobel, R., and Wheeler, A.R. 2012. Digital microfluidics. *Annual review of analytical chemistry*. 5, 413-440. DOI= https://doi.org/10.1007/s10404-007-0161-8

[6] Fobel, R., Fobel, C., and Wheeler, A.R. 2013. DropBot: An open-source digital microfluidic control system with precise control of electrostatic driving force and instantaneous drop velocity measurement. *Applied Physics Letters* 102, 19, 193513. DOI= https://doi.org/10.1063/1.4807118

[7] Su, F., Chakrabarty, K., and Fair, R.B. 2006. Microfluidics-based biochips: technology issues, implementation platforms, and design-automation challenges. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 25, 2, 221-223. DOI= https://doi.org/10.1109/TCAD.2005.855956

[8] Zhou, X et. al. 2004. Microfluidic PicoArray synthesis of oligodeoxynucleotides and simultaneous assembling of multiple DNA sequences. *Nucleic acids research* 32, 18, 5409-5417. DOI= https://doi.org/10.1093/nar/gkh879

[9] Shin, Y, J,, and Lee, J.B. 2010. Machine vision for digital microfluidics. *Review of Scientific Instruments*, 81, 1, 014302. DOI= http://dx.doi.org/10.1063/1.3274673

[10] Zhang, M., McLaughlin, J., Wipat, A., and Myers, C. J. 2017. SBOLDesigner 2: an intuitive tool for structural genetic design. *ACS synthetic biology*, 6, 7, 1150-1160. DOI= https://pubs.acs.org/doi/abs/10.1021/acssynbio.6b00275

[11] Der, B.S., et. Al. DNAplotlib: programmable visualization of genetic designs and associated data. *ACS synthetic biology* 6, 7, 1115-1119. DOI= https://pubs.acs.org/doi/abs/10.1021/acssynbio.6b00252

[12] Nielsen A. A. et. al. 2016. Genetic circuit design automation. *Science* 352, 6281, aac7341. DOI= https://doi.org/10.1126/science.aac7341

Automating synthetic biology using microfludics

James Perry, Mathieu Husser, Philippe Q.N. Vo, Fatmeh Ahmadi, Steve C. C. Shih
Concordia University
Department of Electrical and Computer Engineering, Biology
Montreal, Quebec, Canada
Email: steve.shih@concordia.ca

***Abstract*—This paper describes new microfluidic methods for synthetic biology applications. We describe two works that are related to synthetic biology, namely, DNA assembly and transformation and strain optimization. Both are necessary processes for engineering new organisms. In addition, we have shown the power of integration and automation that can potentially expedite these processes, specifically, the design-build-test-learn cycle.**

Keywords—microfluidics, automation, synthetic biology

I. INTRODUCTION

Lab-on-chip or microfluidic technologies are characterized by a miniaturization of experiments and integration of laboratory instruments onto tiny hand-held devices. A burgeoning platform called digital microfluidics (DMF) is the manipulation of fluids as discrete droplets on an open array of electrodes.[2-4]

The greatest advantage of digital microfluidics is perhabs its amenability to integrating automation systems and coupling the platform to external detectors (or internal in-line detectors) for real-time or downstream biological analysis. [5, 6] The core of DMF automation systems interfaces with a DMF device which enables droplet movement with a standard set of basic instructions written by the user. The user will interact with the graphical user interface (GUI) to program a set of instructions to dispense, move and split droplets, merge droplets together and to mix resulting samples and sort droplets for analysis (Fig. 1). Such automation gives DMF the capacity to operate droplets in parallel on a single device, without the need for any valves or pumps.

Typically, DMF automation systems rely on an array of relay switches, each of which is responsible for one individual electrode on the device and relays AC or DC voltages to it when instructed. The state of the switches is controlled through a computer and microcontroller. Specifically, our automation system (Fig. 2) consists of a in-house program that is used to control an Arduino Uno microcontroller.[7] Driving input potentials of 100-200 V_{RMS} are generated by amplification of a sine wave output from a function generator operating at ~kHz by an amplifier and delivered to the PCB control board. The Arduino controls the state of high-voltage relays that are soldered onto the PCB control board. The logic state of an individual solid-state switch is controlled through an I^2C communication protocol by an I/O expander. This control board is mated to a pogo pin interface (104 pins), where each switch delivers a high-voltage potential (or ground) signal to a contact pad on the DMF device. See our GitHub registry (https://github.com/shihmicrolab/Automation) to assemble the hardware and to install the open-source software program to execute the automation system.

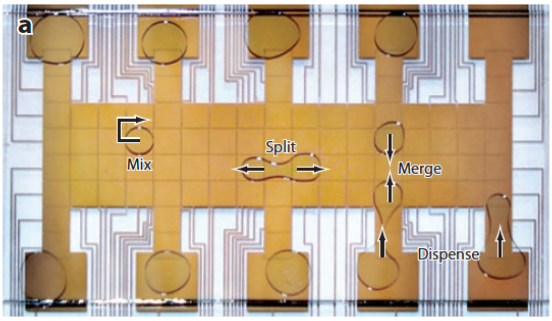


Fig.1 – Operations performed on a DMF device. *(Image obtained from Choi et al.[1])*

In synthetic biology, there has been a push to find the next technological innovation that can automate therocess of design, build, test, and learn. This cycle follows an iterative process that often requires extensive manual intervention. This process is used to engineer new microbes that contain the necessary genetic circuit and metabolic pathways to produce the required outputs for a wide range of applications such as bio-based chemicals and biofuels. While current available tools are useful in improving the synthetic biology process, further improvements in physical automation would help lower the barrier of entry into this field. Here in

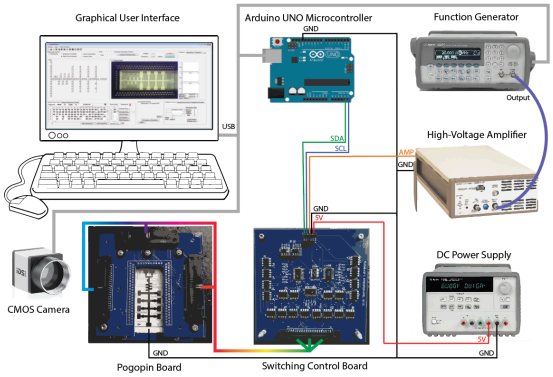


Fig.2 – Automation system for digital microfluidics. Hardware and software tools available on our GitHub website.

this abstract, we show results related to some of the innovations related to synthetic biology and microfluidics.

II. Building Microbes and Optimization

The build phase in synthetic biology workflow consists of two main processes: DNA synthesis and DNA assembly. In our first project, we describe how we can expedite DNA assembly processes with transformation using microfluidics. DNA assembly and transformation have been demonstrated on digital microfluidics platforms[8, 9] but the extensive cleanroom fabrication process remains restrictive to biologist. Alternatively, it has been shown that DMF devices can be fabricated out of printed circuit boards (PCBs) using limited resources.[10] This process, called rapid-prototyping, allows for same-day design, fabrication and use of DMF devices. We present a finely tuned and tractable rapid prototype DMF platform to automate assembly and transformation of plasmid DNA in E.coli. The design integrates PID controlled thermal electric cooling modules which assign spatial-temporal temperatures required for on-chip assembly and heat shock transformation. This platform is demonstrated by constructing expression cassette libraries spanning a range of transcriptional and translational control with further application to antibiotic production (Fig. 3a).

In addition, we present a system that describes expression of a recombinant gene in a host organism through induction can be an extensively manual and labor-intensive procedure. Several methods have been developed to simplify the protocol, but none has fully replaced the traditional IPTG-based induction. To simplify this process, we describe the development of an auto-induction platform based on digital microfluidics (Fig. 3b). This system consists of a 600 nm LED and a light sensor to enable the real-time monitoring of samples optical density (OD) coordinated with the semi-continuous mixing of a bacterial culture. A hand-held device was designed as a micro-bioreactor

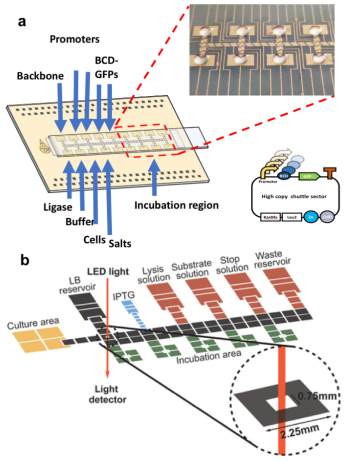


Fig.3 – DMF for (a) building plasmids and (b) strain optimization

to culture cells and to measure the OD of the bacterial culture. In addition, it serves as a platform for the analysis of regulated protein expression in *E.coli* without the requirement of standardized well-plates or pipetting-based platforms. We used our system to identify active thermophilic β-glucosidase enzymes which may be suitable candidates for biomass hydrolysis.

References

[1] K. Choi, A. H. C. Ng, R. Fobel, and A. R. Wheeler, "Digital Microfluidics," *Annual Review of Analytical Chemistry, Vol 5,* vol. 5, pp. 413-440, 2012.

[2] M. J. Jebrail, M. S. Bartsch, and K. D. Patel, "Digital microfluidics: a versatile tool for applications in chemistry, biology and medicine," *Lab Chip,* vol. 12, pp. 2452-63, Jul 21 2012.

[3] M. J. Jebrail and A. R. Wheeler, "Let's get digital: digitizing chemical biology with microfluidics," *Curr Opin Chem Biol,* vol. 14, pp. 574-81, Oct 2010.

[4] A. R. Wheeler, "Chemistry. Putting electrowetting to work," *Science,* vol. 322, pp. 539-40, Oct 24 2008.

[5] L. Malic, T. Veres, and M. Tabrizian, "Two-dimensional droplet-based surface plasmon resonance imaging using electrowetting-on-dielectric microfluidics," *Lab Chip,* vol. 9, pp. 473-5, Feb 7 2009.

[6] S. H. Au, S. C. C. Shih, and A. R. Wheeler, "Integrated microbioreactor for culture and analysis of bacteria, algae and yeast," *Biomed. Microdevices,* vol. 13, pp. 41-50, Feb 2011.

[7] P. Q. N. Vo, M. C. Husser, F. Ahmadi, H. Sinha, and S. C. C. Shih, "Image-based feedback and analysis system for digital microfluidics," *Lab Chip,* vol. 17, pp. 3437-3446, Oct 11 2017.

[8] P. C. Gach, S. C. C. Shih, J. Sustarich, J. D. Keasling, N. J. Hillson, P. D. Adams, *et al.*, "A Droplet Microfluidic Platform for Automating Genetic Engineering," *ACS Synth. Biol.,* vol. 5, pp. 426-33, May 20 2016.

[9] S. C. C. Shih, G. Goyal, P. W. Kim, N. Koutsoubelis, J. D. Keasling, P. D. Adams, *et al.*, "A versatile microfluidic device for automating synthetic biology," *ACS Synth. Biol.,* vol. 10, pp. 1151-1164, Jun 15 2015.

[10] M. Abdelgawad and A. R. Wheeler, "Rapid prototyping in copper substrates for digital microfluidics," *Advanced Materials,* vol. 19, pp. 133-137, 2007.

Toward Programming 3D Shape Formation in Mammalian Cells

Jesse Tordoff
Massachusetts Institute of Technology
tordoff@mit.edu

Jacob Beal
Raytheon BBN Technologies
jakebeal@ieee.org

Ron Weiss
Massachusetts Institute of Technology
rweiss@mit.edu

1 MOTIVATION

Biological cells are remarkably effective at predictable and resilient formation of complex three-dimensional shapes, as aptly demonstrated by most multicellular life on this planet. Not only can intricate shapes be formed with high reliability, but organisms also maintain functional integration of the entire system throughout development, as well as adapting form in response to environmental conditions, damage, and other disruptions. Moreover, these feats of manufacturing are accomplished entirely with reprocessed locally harvested materials.

Our goal is to make these sorts of capabilities available for human engineering as well, through the reprogramming of living cells. We have selected mammalian cells as an initial platform for investigation, as these cell lines are well-studied, tractable for engineering, physically large and robust, and natively host many tools useful for shape formation. Furthermore, study of natural morphogenesis and relationships between evolution and development [2] suggests a set of natural "building blocks"—such as cells self-sorting by differential adhesion or targeted migration, gradient-based coordinates, and differential growth—that might be combined modularly to program shape formation. We are thus pursuing a research program of shape formation through isolation of biological shape formation "building blocks" and development of a system of genetic circuits for combining such building blocks into programs for the formation of complex three-dimensional shapes.

Toward this end, we have developed prototype motif-based compilation software for mapping from high-level three-dimensional shape specifications to genetic construct and sample designs. This compiler is supported by characterization experiments and an analytical workflow for converting experimental results into formulae for setting design parameters. Preliminary results from this work are promising, and we are now working to extend these into a full proof of concept for three-dimensional shape formation in mammalian cells.

2 APPROACH: MOTIF-BASED COMPILATION

The core idea behind our approach is motif-based compilation, building on our previous work with the Proto BioCompiler [1]. Under this approach, each operation in a high-level programming language is associated with a biological motif—a "template" design comprising a partial system specification, with variables for inputs and outputs. These motifs are then stitched together by instantiating a motif for each operation and connecting each motif instance input to its corresponding motif instance output (as specified by high-level program structure) to form a complete biological system specification. This specification may then be further refined by optimization, mapping of abstract parts to specific instances, etc.

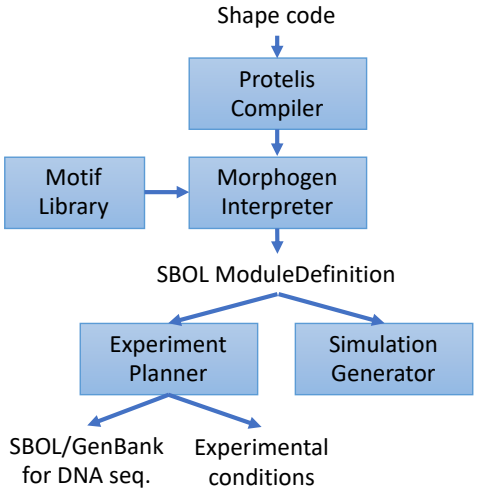


Figure 1: Programmed shape-formation architecture: a high-level shape specification is compiled by Protelis and interpreted by Morphogen (using a motif library) to produce an SBOL specification of the complete system. This may then be sent to a simulator for validation and/or exported for experiment as DNA sequence and sample specifications.

For our current implementation (Figure 1), we have updated our motif-based compiler to be based on the Protelis programming language [3], a Java-hosted aggregate programming language with a more accessible syntax and better suited for adaptation and integration. We then swap the standard Protelis interpreter for a biological interpreter implementation that we call Morphogen, which transforms the program into a biological systems specification. In particular, Morphogen uses a Java-based library of motifs, each of which maps a Protelis operation to an SBOL ModuleDefinition [4] specifying a set of biological parts and interactions. A complete system specification is also a ModuleDefinition, constructed by the Morphogen interpreter: each time an Protelis operation is invoked, its associated ModuleDefinition is instantiated as a Module in the overall system ModuleDefinition, and its ports are linked to other Modules based on the Protelis program.

Our system thus takes in a high-level shape specification in Protelis and transforms it into an SBOL ModuleDefinition that specifies a biological system that is expected to produce the specified shape. From there, the specification may either be sent to a simulator for verification or else may be exported for realization in the laboratory as a set of DNA sequences and specifications of how sequences, strains, and other reagents should be combined to form experimental samples. We have implemented a preliminary version of this architecture, and are in the process of developing a number

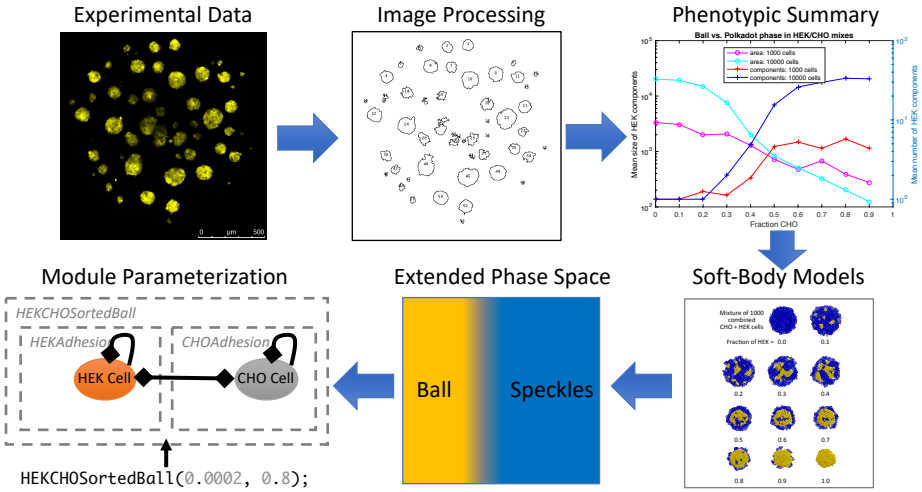


Figure 2: Motif development workflow: microscopy images are processed to produce an initial phenotypic analysis of shape formation behavior. This analysis is used to parameterize models to refine characterization of the space of realizable behaviors, from which functions are extracted to constrain and parameterize applications of motifs in Protelis programs.

of composable motifs, including cell-sorting, cell-to-cell communication, symmetry breaking, cell type differentiation, and phase synchronization.

3 DEVELOPMENT OF MOTIFS

Motif development is key to realizing our approach. For each shape formation "building block," we need to not only demonstrate the capability of interest, such as cell-sorting, but also need to evaluate the effective range over which that capability can be realized and need to establish the numerical relationship between the values of inputs and experimental parameters and the properties of realized shapes. To this end, we have also developed a workflow for analysis of experimental data and its refinement into functions for validity testing and parameterization of motif applications in Morphogen interpretation of Protelis programs.

Figure 2 illustrates this workflow, along with examples of preliminary results taken from our use of this workflow in development of cell-sorting motifs. Shape formation experiments with a prospective motif produce microscopy data, which is processed through an image analysis pipeline to produce detailed statistics of the patterns formed by cells—in this example, cell-sorting producing a "polka dot" pattern. From these statistics, we produce an initial phenotypic analysis (in this example, the transition between "sorted ball" and "polka dot" behaviors), which is then refined with the aid of soft-body models to produce a predicted phase space of behaviors interpolated and extrapolated beyond experimental results. From this, we then extract functions that both constrain and parameterize applications of the motifs in Protelis programs.

We have applied this workflow to develop motifs based on cell-sorting: a "sorted ball" with differentiated interior and exterior, and "polka-dot" patterns. Furthermore, we have demonstrated that with such motifs we can reconstruct the plans that produced experimental results from appropriate Protelis specifications and that we can produce compiler errors for shapes that cannot be reliably realized

4 CONTRIBUTIONS & FUTURE DIRECTIONS

Thus far, we have developed an architecture for programming three-dimensional shape formation in mammalian cells, comprising a motif-based compiler and a supporting experimental and analytical workflow for parameterization of designs being produced by the compiler. Preliminary results from this architecture are promising, showing that the compiler can reconstruct experimental designs and reject specifications that cannot be achieved. The next steps in development will be to extend and enhance the collection of experiments and models to expand the set of motifs available to the compiler, then deploy these to predict and demonstrate programmed formation of more complex three-dimensional shapes.

ACKNOWLEDGMENTS

Additional authors (alphabetically): Bryan Bartley, Gizem Gumuskaya, Katherine Kiwimagi, Matej Krajnc, Kevin Lebo, Stanislav Shvartsman, Allen Tseng, Nicholas Walczak. This work has been supported by the Defense Advanced Research Projects Agency under Contract No. W911NF-17-2-0098. The views, opinions, and/or findings expressed are of the author(s) and should not be interpreted as representing official views or policies of the Department of Defense or the U.S. Government. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

REFERENCES

[1] J. Beal, T. Lu, and R. Weiss. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS ONE*, 6(8):e22490, August 2011.
[2] S. B. Carroll. *Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom.* W. W. Norton & Company, 2005.
[3] D. Pianini, M. Viroli, and J. Beal. Protelis: practical aggregate programming. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 1846–1853. ACM, 2015.
[4] N. Roehner et al. Sharing structure and function in biological design with SBOL 2.0. *ACS Synthetic Biology*, 5(6):498–506, 2016.

Software Projects of the Edinburgh Genome Foundry

Valentin Zulkower*
Edinburgh Genome Foundry

Aitor Bleda
Edinburgh Genome Foundry

Isaac Luo
Edinburgh Genome Foundry

the Edinburgh Genome Foundry team
University of Edinburgh, UK

KEYWORDS

Synthetic Biology, DNA foundry, computer-aided design, computer-aided manufacturing

1 INTRODUCTION

We present some software projects of the Edinburgh Genome Foundry (EGF), a research facility specialized in the automated assembly of DNA constructs. The EGF operates an integrated robotic setup automating all operations of DNA assembly: liquid dispensing, thermo-cycling, plating and colony picking, plasmid extraction, fragment analysis, etc.

The EGF software enables assembly batches to be swiftly and reliably carried out on the platform, by automating interactions between EGF customers, operators, databases, and machines (Figure 1). Many features are not specific to the EGF workflow and could support the routine cloning of other facilities and individual researchers, notably for assembly planning, quality-control, and troubleshooting.

In this perspective, EGF software projects are organized as a modular collection of open-source Python libraries [1] and public web applications[2] (Figure 2) to encourage their use by other groups. We will highlight some of these projects and showcase their use at the EGF.

2 CAD SOFTWARE

The EGF's computer-aided design (CAD) software aims at assisting customers, in particular non-specialists, in the design of custom sequences to be assembled on the robotic platform.

DNA Chisel is a sequence optimizer that builds on previous work [3, 6, 8] and lets users define sequence design specifications via Genbank annotations. Easily extensible using Python scripts, it has been used to design projects involving large sequences - up to 50kb - and hundreds of specifications. It is also used in the routine domestication of parts at the EGF.

The *Golden Hinges* framework can be used to generate collections of compatible overhangs to create new type-IIs assembly standards, to extend existing standards, or to decompose arbitrary sequences into assembly-compatible fragments.

3 CAM SOFTWARE

The EGF's computer-aided manufacturing (CAM) software aims at automating all processes between the reception of an order and its delivery, at once freeing up operator time and avoiding human error.

DNA Cauldron is a cloning simulation framework with a focus on restriction-based assembly (Golden Gate, BASIC). It extends a previous approach [7] in order to predict the final sequences of single and combinatorial assemblies, autocomplete assembly designs with *linker parts*, and provide visual troubleshooting aid for invalid assembly designs.

Plateo is a Python laboratory automation framework for parsing machine files into human-readable formats, and generating liquid dispensing *picklists* for different robots. It can also simulate picklists to predict final microplate layouts and prevent pipetting a well over capacity or under dead volume.

Two additional libraries automate quality control. *Bandwitch* continues previous efforts [2] for restriction digest planning and validation of large assembly batches, and helps troubleshooting failed assemblies by identifying partially-cutting enzymes or deficient parts. *Primavera* automates the selection of primers (available or newly designed) for Sanger sequencing of large assembly batches.

4 FUTURE DIRECTIONS

Software efforts at the EGF are driven by customer projects and operator needs. On-going projects include *EMMA-DB*, a website to guide users without prior knowledge of common assembly standards (e.g. MoClo or EMMA [5]) towards the right construct structure for their needs, *Smart Ass*, an assembly assistant for foundry operators, and *DNA Weaver*, a framework to find optimal assembly strategies for large sequences. The foundry is collaborating with Genemill on onboarding LEAF LIMS [1] for assembly project management, and with ThermoFisher to ensure tighter, real-time communication between the foundry's software and the robotic setup.

*valentin.zulkower@ed.ac.uk

[1] Hosted on Github at *https://github.com/Edinburgh-Genome-Foundry*. Software home page at *https://edinburgh-genome-foundry.github.io*

[2] Hosted at *http://cuba.genomefoundry.org/*

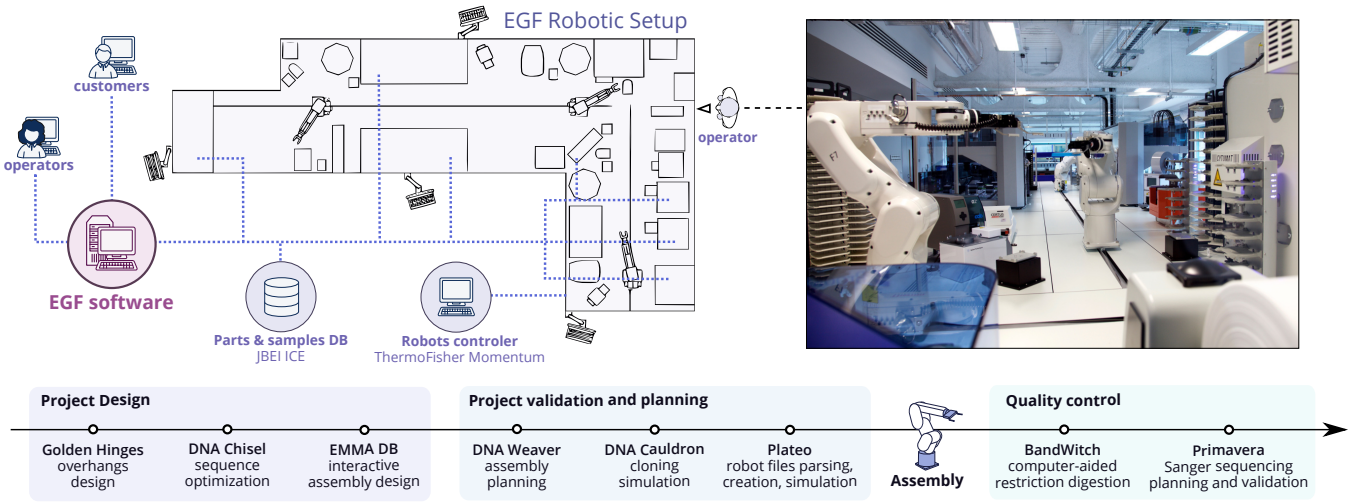


Figure 1: The EGF software interfaces the foundry's customers, operators, and databases, with the different terminals of the robotic platform. Third-party software *ThermoFisher Momentum* and *JBEI-ICE* [4] ensure direct robot control and genetic parts management, respectively. The inset photo shows the robotic setup from the operator's viewpoint. The time-line represents the different software projects discussed here, in the order in which they may be used in a typical assembly project.

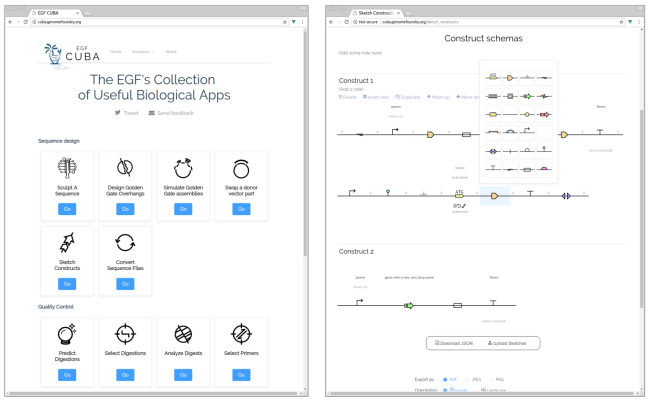


Figure 2: Screen captures of the EGF's public Collection of Useful Biological Apps (EGF CUBA), currently featuring a dozen applications.

5 ACKNOWLEDGEMENTS

The EGF team is supported by the Research Councils' UK Synthetic Biology for Growth Programme (BBSRC grants BB/M025659/1, BB/M025640/1, and BB/M00029X/1 to YC)

REFERENCES

[1] Thomas Craig, Richard Holland, Rosalinda D'Amore, James R. Johnson, Hannah V. McCue, Anthony West, Valentin Zulkower, Hille Tekotte, Yizhi Cai, Daniel Swan, Robert P. Davey, Christiane Hertz-Fowler, Anthony Hall, and Mark Caddick. 2017. Leaf LIMS: A Flexible Laboratory Information Management System with a Synthetic Biology Focus. *ACS Synthetic Biology* 6, 12 (2017), 2273–2280. https://doi.org/10.1021/acssynbio.7b00212

[2] Yandi Dharmadi, Kedar Patel, Elaine Shapland, Daniel Hollis, Todd Slaby, Nicole Klinkner, Jed Dean, and Sunil S. Chandran. 2014. High-throughput, cost-effective verification of structural DNA assembly. *Nucleic Acids Research* 42, 4 (2014). https://doi.org/10.1093/nar/gkt1088

[3] Joao C. Guimaraes, Miguel Rocha, Adam P. Arkin, and Guillaume Cambray. 2014. D-Tailor: Automated analysis and design of DNA sequences. *Bioinformatics* 30, 8 (2014), 1087–1094. https://doi.org/10.1093/bioinformatics/btt742

[4] Timothy S. Ham, Zinovii Dmytriv, Hector Plahar, Joanna Chen, Nathan J. Hillson, and Jay D. Keasling. 2012. Design, implementation and practice of JBEI-ICE: An open source biological part registry platform and tools. *Nucleic Acids Research* 40, 18 (2012). https://doi.org/10.1093/nar/gks531

[5] Andrea Martella, Mantas Matjusaitis, Jamie Auxillos, Steven M Pollard, and Yizhi Cai. 2017. EMMA: An Extensible Mammalian Modular Assembly Toolkit for the Rapid Design and Production of Diverse Expression Vectors. (2017). https://doi.org/10.1021/acssynbio.7b00016

[6] Ernst Oberortner, Jan Fang Cheng, Nathan J. Hillson, and Samuel Deutsch. 2017. Streamlining the Design-to-Build Transition with Build-Optimization Software Tools. *ACS Synthetic Biology* 6, 3 (2017), 485–496. https://doi.org/10.1021/acssynbio.6b00200

[7] Filipa Pereira, Flávio Azevedo, Ângela Carvalho, Gabriela F. Ribeiro, Mark W. Budde, and Björn Johansson. 2015. Pydna: A simulation and documentation tool for DNA assembly strategies using python. *BMC Bioinformatics* 16, 1 (2015). https://doi.org/10.1186/s12859-015-0544-x

[8] David Raab, Marcus Graf, Frank Notka, Thomas Schödl, and Ralf Wagner. 2010. The GeneOptimizer Algorithm: Using a sliding window approach to cope with the vast sequence space in multiparameter DNA sequence optimization. *Systems and Synthetic Biology* 4, 3 (2010), 215–225. https://doi.org/10.1007/s11693-010-9062-3

Context-aware predictive tools for portable genetic circuit engineering

Pablo Carbonell, Sandra Taylor, Rehana Sung, Adrian J. Jervis, Rainer Breitling, Jean-Loup Faulon, Nigel S. Scrutton

SYNBIOCHEM, Manchester Institute of Biotechnology, University of Manchester, UK

{pablo.carbonell,sandra.taylor,rehana.sung,adrian.jervis,rainer.breitling,jean-loup.faulon,nigel.scrutton}@manchester.ac.uk

ABSTRACT

Accelerating the engineering cycle of synthetic biology requires rapid transfer of results from the set of circuit prototypes characterized in the initial pilot tests into the design of the circuitry employed at the industrial phase. By focusing on specific areas of the design space, machine learning can be used to boost data-driven engineering and provide compatibility with the scale-up stage. Here, we present a toolbox of predictive tools for selecting genetic parts under different growth conditions such as chassis, media, copy number, induction point or resistance cassette. Such predictive models provide an engineering biology toolbox for designing reusable and portable genetic circuits.

1 INTRODUCTION

Accelerating the engineering cycle of synthetic biology requires of data-driven learning algorithms able to generate rules-based models to augment circuit diversity [1]. To that end, powerful automated modeling tools are being increasingly integrated into the Design–Build–Test–Learn cycle [2]. Challenges however still remain when developing engineered organisms for the production of high-value compounds in order to identify the most suitable combinations of enzymes, regulatory components, chassis organism and growing conditions for the desired biosynthetic pathway [3].

Combinatorial libraries of constructs spanning the design space can be initially characterized and explored in prototypes and then ported into libraries to be embedded into the chassis for process development and scaling-up. However, enzyme efficiency and transcription and translation rates can greatly vary from one host to another and depend on other factors such as growth media, resistance cassette or induction point. For instance, transcription rates observed from promoters can largely vary from one host to another and depend on other factors such as growth media, resistance cassette or induction point. Several groups have worked on characterizing libraries of promoters [6] and compared promoter behavior across model organisms [9], developing characterization methods based on mathematical models [7] as well as control of their dynamic range [5, 8].

With the aim of expanding the catalog of characterized parts that can be modularly reused in multiple engineering biology projects, we present here tools allowing portability of genetic constructs through context-aware machine-learning predictive modeling.

2 ENZYME SELECTION TOOL

Once a producing pathway has been identified, a first requirement is to screen for enzymes for desired target reactions in the pathway and select best candidates sequences depending on host context. To that end, Selenzyme is an online tool that allows querying for target reactions, including novel or hypothetical generic reactions [4]. The query reaction, which is input using SMIRKS representation for generic reaction rules or an external database id, is screened against the SYNBIOCHEM graph database of biochemical knowledge (http://biochem4j.synbiochem.co.uk/). The algorithm proceeds through the list of reactions ranked by decreasing similarity and expression context based on phylogenetic distance to the host chassis as well as other useful predicted physicochemical and conservation properties, including a multiple sequence alignment of the identified hits.

On top of this core tool, there is a web server and a KNIME node for automated workflows. Once the reaction query is submitted, the ranked list of sequence candidates is presented as an interactive table, which can be sorted on user-defined summary scores based on a weighted average of selected columns or properties. Moreover, a RESTful service has been implemented, so that Selenzyme can accept multiple queries from any other web-based application.

3 PROMOTER SELECTION TOOL

Similarly, context-aware selection for promoters regulating transcriptional activity of the pathway constructs is required for portable scale-up. For that purpose, experimental parameters from a promoters library of inducible or constitutive promoters measured at different conditions, such as growth media, chassis, resistance cassette or copy number were used to generate a training set for a predictive model of promoter relative strengths. For constitutive promoters, models for each promoter were trained in the same way as for inducible promoters.

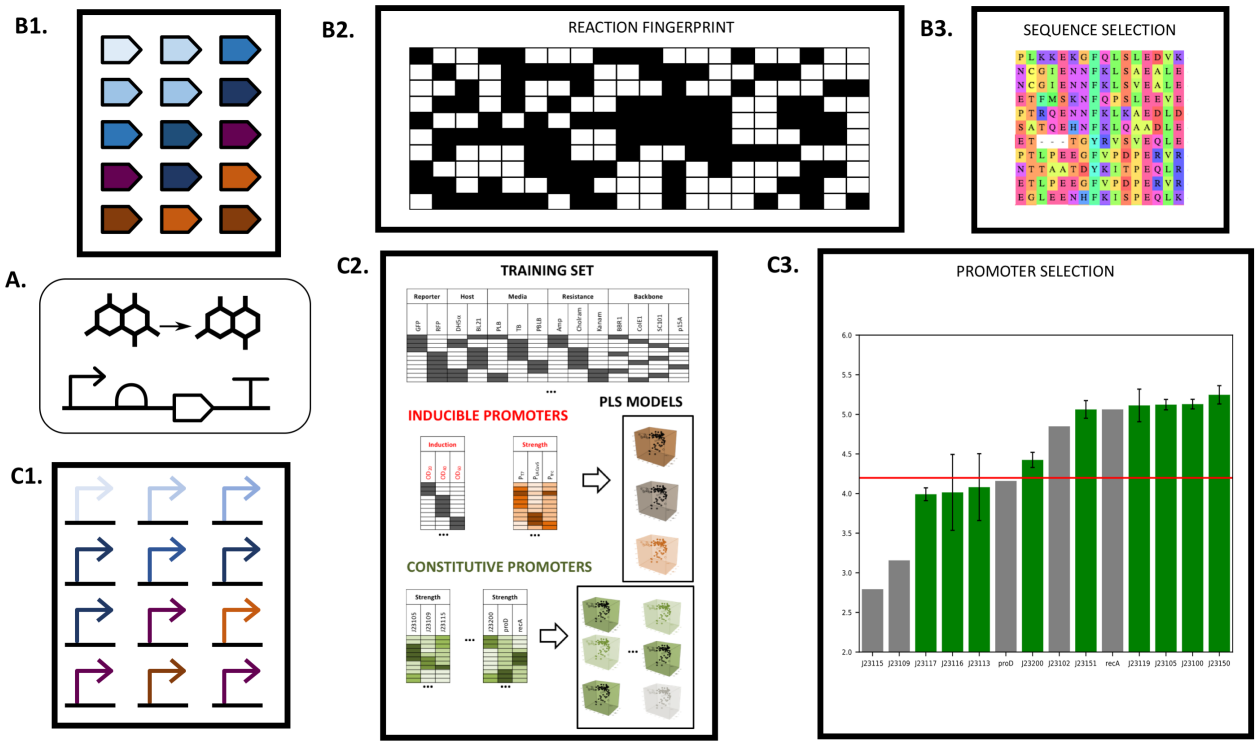


Figure 1: Context-aware predictive tools. A) A design blueprint of genetic parts and their associated pathway activity. B) The enzyme selection tool provides a list of candidates B1 ranked based on their reaction similarity (B2, B3). C) A pool of promoters is selected based on desired growth conditions (C1), a predictive model C2 is used to estimate promoter activity (C3).

A web-based interface provides a selection dashboard where the user selects the experimental set-up of the promoter construct from the training parameters. Promoter strengths for the selected conditions are shown on a bar chart, either from the model prediction or from experimental data with associated errors when available. For each promoter, a link is provided to a data sheet that informs of the measured and predicted strengths, experimental responses at the different growth conditions, and to the deposited parts in a JBEI-ICE repository. A RESTful service will be provided in order to allow the integration of the tool into computational workflows and automated pipelines.

4 CONCLUSIONS

Precise selection tools for DNA parts of genetic circuits are required when transferring initial prototype constructs into industrial scaling-up. Building context-aware biodesign tools such as those presented here for enzyme and promoter selection will accelerate the engineering biology process and ultimately contribute to streamlining the automated biomanufacturing pipeline.

ACKNOWLEDGMENTS

This work has been funded by the BBSRC under Grants BB/M017702/1 and Flexible Talent Mobility BB/R506497/1.

REFERENCES

[1] C. J. Bashor and J. J. Collins. 2018. Understanding Biological Regulation Through Synthetic Biology. *Annu Rev Biophys* 47, (2018).
[2] P Carbonell, A Currin, A Jervis, *et al.* Bioinformatics for the synthetic biology of natural products: integrating across the Design–Build–Test cycle. *Nat Prod Rep* 33, (2016), 925–932.
[3] P Carbonell, A Jervis, C Robinson, *et al.* An automated Design–Build–Test–Learn pipeline for enhanced microbial production of fine chemicals. *Comm Biol* 1, (2018), 66.
[4] P Carbonell, J Wong, N Swainston, *et al.* Selenzyme: enzyme selection tool for pathway design. *Bioinformatics* 34, (2018) 2153–2154.
[5] Y Chen, J Ho, D Shis, *et al.* Tuning the dynamic range of bacterial promoters regulated by ligand-inducible transcription factors. *Nat Comm* 9, (2018), 64.
[6] J R Kelly, A J Rubin, J H Davis, *et al.* Measuring the activity of BioBrick promoters using an in vivo reference standard. *J Biol Eng* 3, 1 (2009), 4.
[7] T Rudge, J Brown, F Federici, *et al.* Characterization of Intrinsic Properties of Promoters. *ACS Synth Biol* (2015).
[8] T Segall, E Sontag, and C Voigt. Engineered promoters enable constant gene expression at any copy number in bacteria. *Nat Biotech* (2018).
[9] S Yang, Q Liu, Y Zhang, *et al.* Construction and characterization of broad-spectrum promoters for synth biol. *ACS Synth Biol* 7, (2018), 287.

Asynchronous Genetic Circuit Design Automation with Cloud-based Component Libraries

Timothy S. Jones
Boston University
8 Saint Mary's Street, Office #324
Boston, Massachussetts 02215
jonests@bu.edu

Tramy Nguyen
University of Utah
50 S. Central Campus Dr., Rm. 2110
Salt Lake City, Utah 84112
tramy.nguy@gmail.com

Zach Zundel
University of Utah
50 S. Central Campus Dr., Rm. 2110
Salt Lake City, Utah 84112
me@zachzundel.com

Chris J. Myers
University of Utah
50 S. Central Campus Dr., Rm. 4112
Salt Lake City, Utah 84112
myers@ece.utah.edu

Douglas Densmore
Boston University
8 Saint Mary's Street, Office #324
Boston, Massachussetts 02215
dougd@bu.edu

ABSTRACT

Most electrical circuits utilize a timing reference to synchronize the progression of signals and enable sequential memory elements. These designs may not be realizable in biological substrates due to the lack of a reliable clock signal. Asynchronous designs eliminate the need for a clock with dual-rail input encoding and signal receipt acknowledgement handshake protocol. We propose a workflow to automate the synthesis of asynchronous genetic circuit designs.

CCS CONCEPTS

•**Applied computing** → *Biological networks;* •**Hardware** → *Biology-related information processing;*

KEYWORDS

genetic circuit design automation, asynchronous genetic circuits, synthetic biology

ACM Reference format:
Timothy S. Jones, Tramy Nguyen, Zach Zundel, Chris J. Myers, and Douglas Densmore. 2018. Asynchronous Genetic Circuit Design Automation with Cloud-based Component Libraries. In *Proceedings of International Workshop on Bio-Design Automation, Berkeley, California USA, July 31 – August 3 2018 (IWBDA 2018),* 2 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Cello [9] is a *computer-aided design* (CAD) tool aimed at the design of genetic, *combinational circuits* where the input signals map directly to the output signals produced. *Sequential circuits*, on the other hand, have input signals that are combined with internal states to produce the desired output signal. While several genetic memory circuits have been created [1], general methodologies for genetic sequential circuit design have not been developed. Sequential genetic circuits could be utilized in applications such as tumor detection circuits as described in Ref. [8]. While most electronic sequential circuits utilize a periodic timing reference, or clock, to order operations, creating such a synchronous clock in biological systems is not practical. Therefore, sequential genetic circuits likely must follow an *asynchronous* paradigm, in which operations are ordered using *handshakes*. The goal of this abstract is to describe a workflow that could enable CAD tools, such as Cello, to be extended to support asynchronous genetic circuits.

2 METHODOLOGY

The proposed workflow for asynchronous genetic circuit design is shown in Figure 1. The workflow begins with a high-level specification encoded using the Verilog language. This high-level specification is then compiled to a *labeled Petri net* (LPN) [3] following a fairly direct syntax-directed translation. This LPN can then be simulated to check behavior using the iBioSim software [4]. At this point, the asynchronous synthesis tool ATACS is used to produce logic equations [7], which can then be converted to a regulatory network expressed in the *Synthetic Biology Open Language* (SBOL) [10]. At this point, the design can be decomposed into two combinational logic networks, one that feeds the set input of a genetic toggle switch and another to feed the reset input. Each of these resulting networks can then be mapped to genetic gates constructed from genetic parts from a library stored in a SynBioHub repository [5]. This final technology mapping approach will leverage graph based covering approach described in Ref. [11] that is guided by Cello's simulated annealing algorithm that maximizes the on-to-off ratio at the circuit's output for all possible input

IWBDA 2018, Berkeley, California USA
2018. 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

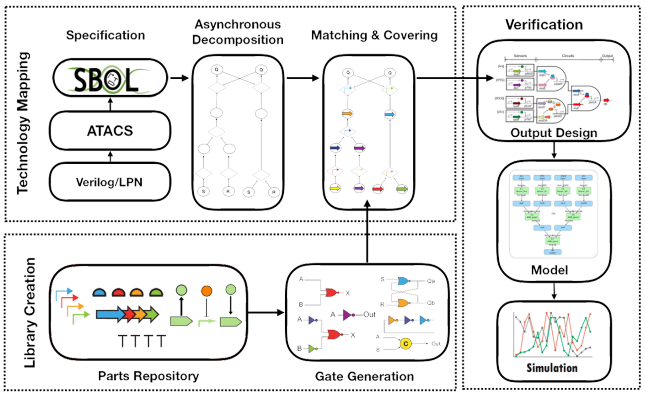


Figure 1: A proposed workflow to perform technology mapping of asynchronous genetic circuit designs.

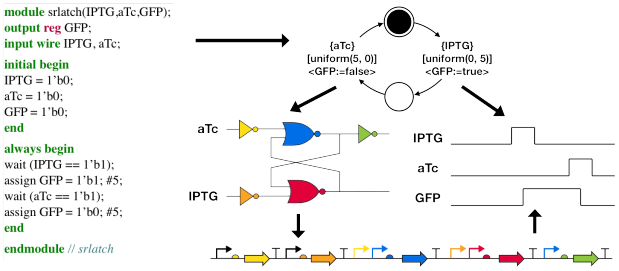


Figure 2: An example of a genetic toggle switch stepping through the proposed workflow.

states [9]. The resulting design can then be converted to a computational model expressed in the Systems Biology Markup Language (SBML) [2] for simulation [6]. The resulting simulation of the design is then be compared to the simulation of the specification to verify that the circuit behavior is as desired.

Figure 2 illustrates this methodology using the example of a genetic toggle switch design [1]. First, the genetic toggle switch is described in Verilog using asynchronous protocols (i.e. wait and assign statements). The Verilog specification is then compiled to an LPN model. The LPN model can be analyzed with a testbench also written in Verilog. After verifying the design meets the specification, the LPN is converted into logic equations. The logic equations are then mapped to a DNA level design using technology mapping. Finally, a model is created for the DNA level design, and it is verified using simulation against the behavior of the original specification.

3 CONCLUSION

This abstract proposes a methodology to extend Cello to support asynchronous genetic circuit designs using parts stored in SynBioHub design repositories. This proposed workflow allows sequential circuits to be described and the resulting designs can be verified through simulation.

ACKNOWLEDGMENTS

The authors of this work are supported by the National Science Foundation under Grant No., 1522074 (T.J., C.M., and D.D.), CCF-1218095 (T.N. and C.M.), and DBI-1356041 (T.N. and C.M.). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] T. Gardner, C. Cantor, and J. Collins. 2000. Construction of a genetic toggle switch in Escherichia coli. *Nature* 403 (20 01 2000), 339 EP –. http://dx.doi.org/10.1038/35002131

[2] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, and et al. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 4 (March 2003), 524–531.

[3] S. Little, D. Walter, C. Myers, R. Thacker, S. Batchu, and T. Yoneda. 2011. Verification of Analog/Mixed-Signal Circuits Using Labeled Hybrid Petri Nets. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30, 4 (April 2011), 617–630. DOI: http://dx.doi.org/10.1109/TCAD.2010.2097450

[4] C. Madsen, C. J. Myers, T. Patterson, N. Roehner, J. T. Stevens, and C. Winstead. 2012. Design and Test of Genetic Circuits Using iBioSim. *IEEE Design Test of Computers* 29, 3 (June 2012), 32–39. DOI: http://dx.doi.org/10.1109/MDT.2012.2187875

[5] J. McLaughlin, C. Myers, Z. Zundel, G. Mısırlı, M. Zhang, I. Ofiteru, A. Goñi Moreno, and A. Wipat. 2018. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS synthetic biology* 7, 2 (2018), 682–fi?!688.

[6] G. Mısırlı, T. Nguyen, , J. McLaughlin, , P. Vaidyanathan, T. Jones, D. Densmore, C. Myers, and A. Wipat. 2018. A computational workflow for the automated generation of models of genetic designs. *ACS synthetic biology* (2018).

[7] C. Myers, W. Belluomini, K. Kallpack, E. Peskin, and H. Zheng. 2001. Timed circuits: A new paradigm for high-speed design. In *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*. ACM, 335–340.

[8] N. Nguyen, C. Myers, H. Kuwahara, Chris W., and J. Keener. 2010. Design and analysis of a robust genetic Muller C-element. *Journal of Theoretical Biology* 264, 2 (2010), 174 – 187. DOI: http://dx.doi.org/https://doi.org/10.1016/j.jtbi.2009.10.026

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

[10] N. Roehner, J. Beal, K. Clancy, B. Bartley, G. Mısırlı, R. Grünberg, E. Oberortner, M. Pocock, M. Bissell, C. Madsen, T. Nguyen, M. Zhang, Z. Zhang, Z. Zundel, D. Densmore, J. Gennari, A. Wipat, H. Sauro, and C. Myers. 2016. Sharing structure and function in biological design with SBOL 2.0. *ACS synthetic biology* 5, 6 (2016), 498–506.

[11] N. Roehner and C. Myers. 2014. Directed Acyclic Graph-Based Technology Mapping of Genetic Circuit Models. *ACS Synthetic Biology* 3, 8 (08 2014), 543–555. DOI: http://dx.doi.org/10.1021/sb400135t

Tracking the provenance of synthetic biological system construction at the DOE Joint Genome Institute (JGI)

Xianwei Meng[1], **Ernst Oberortner**[1], **Nathan J. Hillson**[1,2], **Samuel Deutsch**[1]
[1]DOE Joint Genome Institute, [2] DOE Joint BioEnergy Institute
{xianweimeng,eoberortner,njhillson,sdeutsch}@lbl.gov

INTRODUCTION

The U.S. Department of Energy (DOE) Joint Genome Institute (JGI) is a user-facility, providing DNA sequencing and synthesis services to the scientific community. The DNA Synthesis program[1] enables users to design, build, and characterize biological systems that are relevant to the DOE mission. The JGI DNA synthesis group needs to establish and communicate to the user the workflow for the design, build, and characterization tasks specific to each user project. An example workflow includes (i) phenotypic sequence repository mining, (ii) heterologous expression construct design, (iii) synthetic DNA requisition, (iv) synthetic construct Type-IIs/Golden-Gate, Chewback, or Yeast-based assembly, (v) assembled construct transformation into the target host organism, and (vi) mass-spectrometry secondary metabolite detection.

In this abstract, we describe our initial development to evaluate an approach to provenance tracking for synthetic DNA requisition and synthetic construct assembly. Provenance tracking is paramount for purposes of managing scientific intellectual property, allowing the data process to be reproduced systematically, and identifying defective designs that can be replaced with non-defective alternatives.

RESULTS

Two JGI tools (the Build-OptimizatiOn Software Tools (BOOST) [3] and SynTrack) could communicate with each other via an instance of SynBioHub [2], locally deployed at JGI, using the standardized Synthetic Biology Open Language (SBOL) data exchange format with its recently adopted W3C Provenance (PROV) extension [1].

BOOST provides several design functionalities for DNA sequence synthesis and assembly. At DOE JGI, BOOST is used to modify protein-coding DNA sequences to satisfy commercial DNA synthesis vendor criteria (e.g., %GC, repeats) and remove certain sequence patterns (e.g., restriction sites). BOOST is also used to partition large sequences into synthesizable building blocks with, if desired, overlap sequences for assembly. BOOST outputs a high-level specification of the build process, including tasks regarding (i) which DNA sequences need to be synthesized, (ii) which primers must be used to PCR amplify the synthetic DNA constructs or to linearize the destination vectors, (iii) which enzymes (e.g., for restriction, ligation, or amplification) are required, (iv) which vectors should be used and how they should be linearized — by PCR or restriction digest, and (v) how synthetic DNA constructs are to be joined together (e.g., either homologous recombination or ligation).

To specify build processes, BOOST supports the following five types of activities: *purchase*, *archive*, *amplify*, *cut*, and *join*. BOOST compiles a build process specification into SBOL, using `ComponentDefinition` to represent building elements, PROV-O `Activity` to encode build activities, and connecting elements and activities together using the `prov:wasGeneratedBy` and `prov:qualifiedUsage` properties. BOOST could then push the resulting SBOL document into an instance of SynBioHub, locally deployed at JGI.

To start the physical build process, SynTrack pulls the BOOST-specified build process from SynBioHub, and translates the specification to step-by-step instructions for JGI staff (leveraging robotics) to operate. SynTrack is a workflow-driven system for carrying out and tracking the complex multi-step processes of DNA assembly in a production setting by integrating laboratory automation equipment, such as liquid handling robots, plate readers, colony pickers, and sequencers. At DOE JGI, SynTrack manages (i) the distributions of DNA constructs, (ii) the tracking of plates and their well contents for each batch of DNA assemblies, (iii) the QA/QC outcomes that determine the status of each construct, and (iv) the data that is being accumulated during the assembly processes. SynTrack divides the build process into pre-assembly and assembly stages. Pre-assembly supports (i) collecting synthetic DNA fragments and the primers, (ii) preparing the DNA constructs for assembly after PCR amplification. The assembly process involves (i) joining the linearized vectors with the assembled building blocks of DNA after the PCR gene fusion or chawback, (ii) validating the status of each final construct. SynTrack tracks each step of the build process and augments the BOOST-specified build process with information according to the SBOL and PROV-O data model, such as the start/end times of the activities,

[1]https://jgi.doe.gov/our-science/science-programs/synthetic-biology/

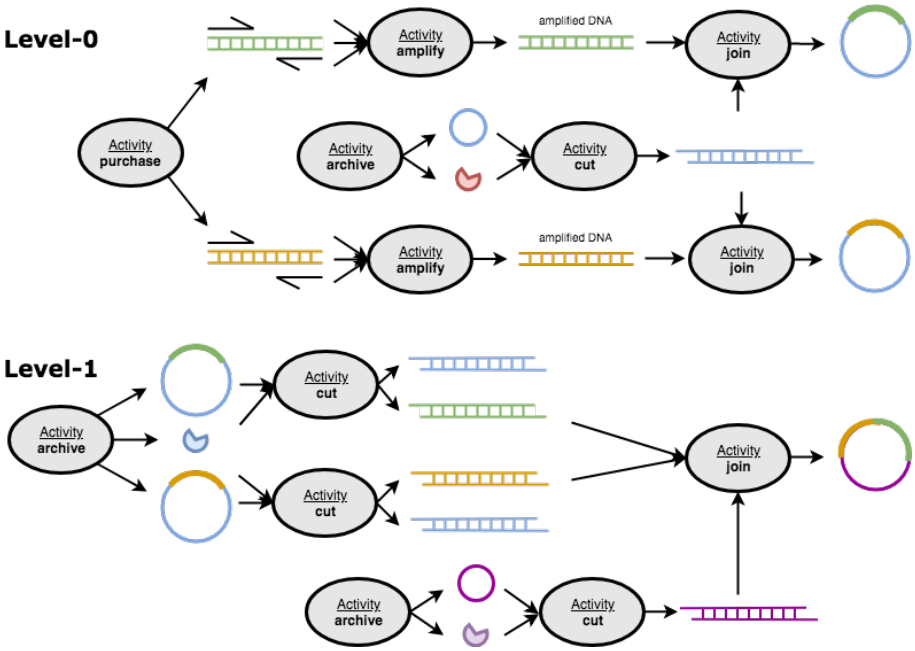


Figure 1: Visualization of multi-level build instructions based on the building elements (synthetic DNA, primer/oligo, vector/plasmid, enzyme) and build activities (purchase, archive, amplify, cut, join).

the agent (human or robotics) who performed the activity, the lineage of DNA fragment data and QA/QC results.

Figure 1 shows a build process specification for a multi-level hierarchical DNA assembly of a two-gene pathway. Level-0 entails the synthesis of the two genes, their PCR and insertion into (digested) intermediate cloning vectors, and subsequent freezer storage. The final Level-0 plasmids (light blue circle with green insert and light-blue circle with orange inserts) are stored in the JGI archive. Level-1 uses the Level-0 constructs, by retrieving the Level-0 plasmids from the archive, using restriction enzymes to cut the inserts out of their cloning vectors, fusing the inserts together, and then inserting the fusion into the destination vector. The final plasmid (magenta circle with green and orange insert) is then again stored in the JGI archive and can be reused for further assembly hierarchies.

DISCUSSION

In this abstract, we describe our initial vision and efforts towards synthetic DNA construction provenance tracking. A high-level specification of the intended build-process (in addition to tracking the build process, its utilized protocols and integrated QA/QC measurements) enables the comparison of the designed build-process with the actual process outcomes. This enables the further analysis of failures and deviations, making it possible to draw conclusions about why certain build steps fail, such as specific sequence features, assembly protocols, or reagent kits. We envision the further develop the DNA Design, Implementation, and Verification Automation (DIVA) platform, which is currently being used to provide real-time status updates about the synthetic constructs. For the user and the JGI, tracking provenance in a standardized fashion enables provenance visualization using off-the-shelf tools, such as PROV-O-Viz[2].

At DOE JGI, we utilize DNA assembly strategies that are generally representative of, but do not exhaustively cover all building elements and DNA assembly techniques used in the scientific community at-large. Our goal is to form collaborations to define an ontology of common terms that describe the required activities in the synthetic biology field that span across the entire design, build, test, and learn cycle and to further develop the infrastructure for automated and standardized provenance tracking.

REFERENCES

[1] Robert Sidney Cox et al. 2018. Synthetic Biology Open Language (SBOL) Version 2.2.0. *J Integr Bioinform* 15, 1 (Apr 2018). DOI: http://dx.doi.org/10.1515/jib-2018-0001

[2] McLaughlin et al. 2018. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synthetic Biology* 7, 2 (2018), 682–688. DOI: http://dx.doi.org/10.1021/acssynbio.7b00403

[3] Ernst Oberortner et al. 2017. Streamlining the Design-to-Build Transition with Build-Optimization Software Tools. *ACS Synthetic Biology* 6, 3 (2017), 485–496. DOI: http://dx.doi.org/10.1021/acssynbio.6b00200

[2] http://provoviz.org/

Open Vector Editor - DNA Visualization and Annotation

THOMAS RICH, TeselaGen Biotechnology, Inc., tnrich@teselagen.com
TIFFANY DAI PHD TeselaGen Biotechnology, Inc., tiffany.dai@teselagen.com
SAM DENICOLA, TeselaGen Biotechnology, Inc., sam.g.denicola@gmail.com
XIMENA MORALES, TeselaGen Biotechnology, Inc.,, ximena@teselagen.com
NATHAN HILLSON PHD, TeselaGen Biotechnology, Inc, njhillson@teselagen.com
MICHAEL FERO PHD, TeselaGen Biotechnology, Inc., mike.fero@teselagen.com

ABSTRACT

TeselaGen's Open Vector Editor[tm] software provides an interface and compute infrastructure for translating Genbank files to well-rendered graphical views; allows users to add and edit DNA annotation; and provides an API for incorporation into other browser-based applications.

CCS CONCEPTS

• **Human-centered computing** → **Visualization**; *Visualization application domains*; Scientific visualization

KEYWORDS

Synthetic Biology, DNA Design, Recombinant DNA, Biotechnology, Sequence Alignment Viewer

1 INTRODUCTION

The TeselaGen Synthetic Evolution[tm] enterprise platform for synthetic biology consists of four major software modules; DESIGN, BUILD, TEST, and EVOLVE. All modules rely on the accurate representation of DNA and DNA annotation in user-friendly graphical views and interfaces. Open Vector Editor is the third generation of a vector editing tool originally developed at LBNL [1]. OVE provides the necessary viewing and data manipulation components, supporting several major requirements:

- DNA viewing and editing
- Annotation viewing and editing
- Sequence alignment views
- An Open Source distribution

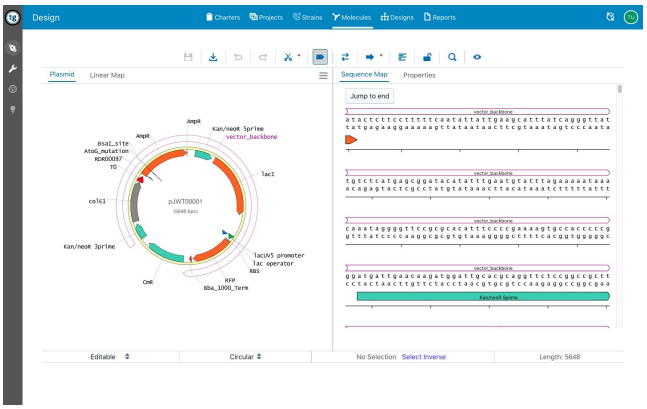


Figure 1: TeselaGen's Open Vector Editor used in the context of the TeselaGen DESIGN module.

2 DNA VIEWING and EDITING

The fundamental role of the OVE module is the seamless translation of Genbank data into a clear, easy to interpret visualization of that data. Additionally, OVE provides a straightforward editor for changing the DNA sequence using familiar text editing tools. While incorporating all of the attributes of a text editor, OVE speaks the language of DNA, introducing a number of additional requirements. Examples of features integrated into OVE to meet scientific requirements include translation from DNA to amino acid sequence in different reading frames; a controlled vocabulary of possible nucleotides; the ability to set the map view to circular versus linear; and modification of the origin (the first base pair) of a sequence. The tool has numerous visibility options as well as a properties view that allows users to visualize:

1. General Properties - Name, Circular/Linear, Length, Is Editable
2. Features - Color, Name, Type, Size, Strand
3. Parts - Name, Type, Size, Strand
4. Primers - Name, Type, Size, Strand
5. Translations - Size (aa), Size (bp), Strand
6. Cutsites - Name, Number of Cuts
7. Orfs - Color, Size (aa), Size (bp), Frame, Strand
8. Genbank - Preview

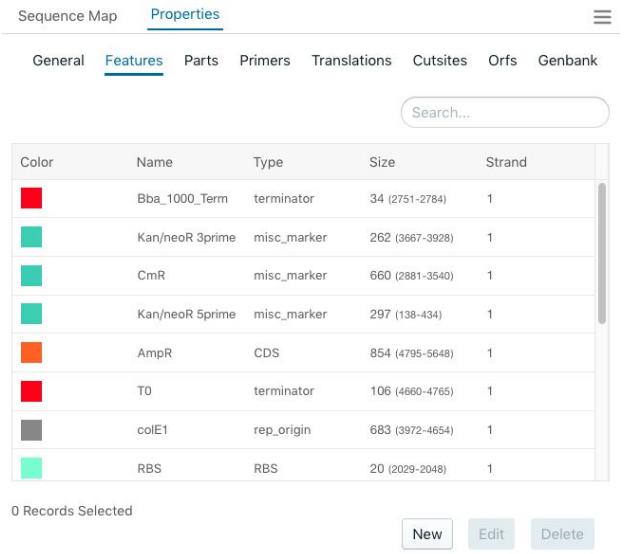


Figure 2: Properties Interface of Open Vector Editor.

3 ANNOTATION VIEWING and EDITING

Genbank annotation editing is available through a straightforward modal window that allows the user to create or edit annotations for any stretch of DNA. The controlled vocabulary of possible annotations is configurable and includes over 100 standard annotations.

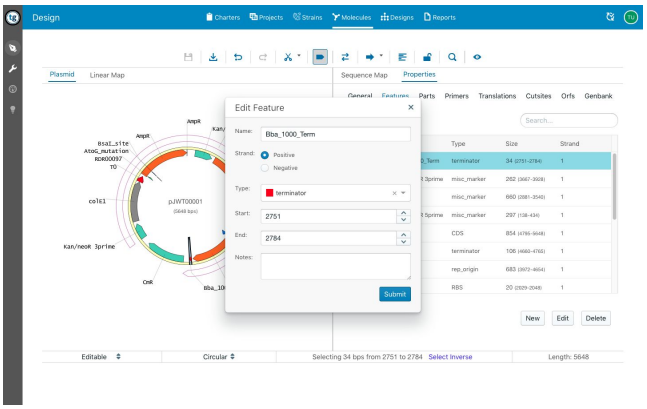


Figure 3: Annotation Interface.

4 SEQUENCE ALIGNMENT VIEWER

The platform also includes a tool for visualizing alignments of DNA sequences. Currently, OVE offers three types of sequence alignment: 1) multiple sequence alignment (MSA), 2) pairwise alignment, and 3) alignment of short sequences to a long template. The alignment algorithms MAFFT, MUSCLE, and Bowtie2 are employed.

MSA and pairwise alignments are relevant to the comparison of sequences that are approximately the same length and span the same gene, as regions of homology are aligned. MSA allows for alignment of three or more sequences, whereas a pairwise alignment individually aligns each uploaded file against the template. Pairwise alignment is particularly useful when checking a constructed or synthesized DNA sequence against the expected sequence specified during the design process. Alignment of short sequences to a long template is appropriate for aligning short sequencing reads to a long reference sequence.

To align sequences in OVE, users are able to input sequences as text or upload sequences in various file formats, such as .ab1 files containing chromatogram data from Sanger sequencing. Users select the type of alignment and tag a sequence in the list as a template when appropriate. OVE displays the resulting sequence alignment as scrollable alignment tracks. If a reference sequence is chosen, the reference is fixed at the top of the display for ease of comparison to other sequences. Annotations appear above each sequence and may be toggled on or off in a visibility menu. Users may also adjust the zoom for an overview of the sequences or for a closer view of the nucleotides. The sequences are shown as a gapped alignment, and mismatches are delineated by red highlighting. Below the alignment tracks, a minimap provides a general view of the sequence alignment. The minimap provides a snapshot of both the base pair position of alignment as well as sequence homology, with identical regions depicted in gray and mismatches indicated in red.

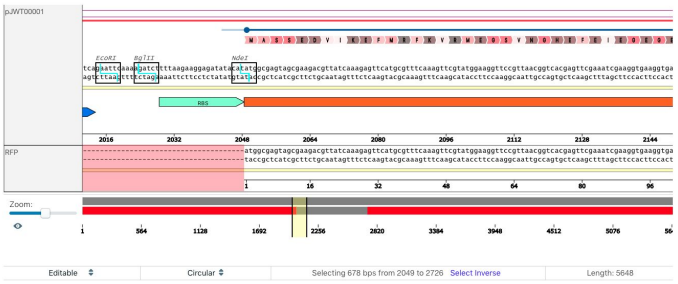


Figure 4: Sequence alignment view of Open Vector Editor.

5 Open Source Access via NPM

Users are able to access the library elements through an open source library with a well documented API accessible as an npm package on Github:

https://www.npmjs.com/package/open-vector-editor

6 CONCLUSIONS

TeselaGen's Open Vector Editor[tm] software library is intended to offer flexibility and functionality to users preparing designs incorporating DNA parts as an organizational paradigm. In addition, tools are provided to assist in the analysis and quality assurance of synthetically built DNA sequences. Scientists are able to visualize, annotate, and edit DNA sequences for various use cases, from viewing constructs during experimental design to analyzing alignments after sequencing in the lab. OVE offers clear graphical representations of DNA sequences as well as part and functional annotation.

ACKNOWLEDGMENTS

This work was supported in part by NSF SBIR Phase IIB 1430986.

REFERENCES

[1] Timothy S. Ham, Zinovii Dmytriv, Hector Plahar, Joanna Chen, Nathan J. Hillson, Jay D. Keasling, 2012. Design, implementation and practice of JBEI-ICE: an open source biological part registry platform and tools *Nucleic Acids Research,* 40 (18) 1 Pages e141, https://doi.org/10.1093/nar/gks531

GeneTech 2.0: Improved Genetic Circuit Synthesis and Technology Mapping

Muhammad Abdullah Siddiqui[1], Adil Ali Khan[1], Hasan Baig[1] and Jan Madsen[2]
[1]Habib University, Pakistan; [2]Technical University of Denmark, Denmark
hasan.baig@sse.habib.edu.pk, jama@dtu.dk

1. INTRODUCTION

Genetic circuit synthesis is an important emerging field aiming to perform logical computations inside living cells. These circuits consist of a genetic component encoded in DNA and operate inside living cells to execute desired logical operations activated by the presence or absence of certain proteins or other species.

There are several existing tools for the synthesis and technology mapping of genetic circuits [1]. One of such tools is *GeneTech* [2] which generates all feasible genetic circuits from a given Boolean expression, thus allowing the user to synthesize genetic circuits only by specifying the desired logical function to be performed in s living cell. *GeneTech* implements a top-down approach to synthesize logic circuits by converting a high-level (Boolean) description of a genetic circuit (Figure 1(a)) into its low-level representation similar to that of the SBOL visual notation [3] (Figure 1(b)). The software performs this operation by first minimizing the logic expression to obtain a function with minimum possible *literals*, and then transforming this optimized expression into NOR-NOT form. Finally, the software generates the circuits using the actual NOR/NOT gates available in the genetic gates library [4], thus achieving all genetic circuits for the desired logical behavior.

However, the *GeneTech* software has some limitations that affect the user experience and restrict the range of its practical implementations. Apart from the software layout being more task oriented, rather than user friendly; the following four major areas need improvements; first, the software accepts the input Boolean expression in the "Sum of Products" (SOP) form [5] only. Second, it doesn't fully address the problem of *unintended negative feedback loops* [6]. A *negative feedback* loop is said to exist in a genetic logic circuit when an output signal of any stage of the circuit is also the input of any previous stage of the circuit. Due to the nature of the stochastic environment of a living cell, such an *unintended feedback* loop essentially deteriorates the working of the entire circuit, rendering it void. Third, in some cases, the optimized expressions generated by the tool contain multiple *nested* NOR-NOT expressions, which indicate that the circuits would have multiple outputs (for example, see the circuits in [4] which have multiple outputs added with OR gate, e.g., x1C, 60, 87 etc.). The tool is currently not able to generate multiple output circuits. Fourth, the current version of the software does not provide the users with an option to decide the design constraints for the output logic circuits (for e.g. time taken, energy required, area cost [7] etc.), instead simply generates all possible circuits. For a larger set of circuits, this is impractical and a time-consuming approach, as it requires the user to manually scrutinize all the output circuits.

Similar to electronic circuits, it has been demonstrated that the *timing* is a crucial design characteristic of genetic circuits [8], in order to make sure that the correct output signal is generated within a certain time duration. If the synthesis tools do not allow user to define their design constraints, they may end up having circuits, which may not only affect the circuits' functionality in terms of timings, but also in terms of signal strength to trigger the circuit's output. The need for constraints is also crucial because implementing a large circuit in a living cell can increase the metabolic burden, since it would require more cellular energy to simply maintain its presence in a host cell and this in turn would increase the probability of resource and energy redistribution among different species, decreasing the efficiency of the circuit as a whole, while simultaneously increasing the time taken to process it [7][8].

In this work, we address these limitations of the *GeneTech* tool by implementing some additional functionalities to make it more user friendly, and more understandable to a broader audience including *engineers*.

2. METHODOLOGY

GeneTech 2.0 is an improvement upon its predecessor in several aspects. In the very first stage, we intend to allow the user the flexibility of inputting either the SOP form of the Boolean expression or the Product of Sum (POS) form. Since the *GeneTech* software in place operates by taking a standard SOP form and simplifying it for further stages; we have employed the 'Quine-McCluskey' method [9] to detect and convert the standard POS form to SOP form; thereafter the tool carries out its usual operations.

Furthermore, the algorithm maintains a record of all the proteins/promoters generated throughout any given circuit that it is processing; checking all the time whether an *unintended feedback loop* is occurring within that circuit. This is an improvement from the original *GeneTech* software which merely attempts to ensure at each stage, that an *unintended loop* is not being formed with the preceding stage. This mechanism will greatly increase the benefit of the software to the users since they will no longer have to manually go through the circuit outputs, verifying that no *unintended feedback loop* is present, before implementation.

Whereas the above points are only alterations to the original software, one major addition to the new version is that it allows the users the flexibility to generate logic circuit combinations considering a variety of possible constraints, including the maximum number of gates in a circuit and the number of possible circuits required (see Figure 1(d)). These constraints are basically intended to ensure that a minimum cost expectation is met by the circuits, generated by the tool, allowing the users to select the most cost-effective output. We accomplished this by annotating the gates library, assigning arbitrary semi-realistic values to costs for each logic gate. These costs can be anything ranging from the *'time taken to process the logic in real life'* to the '*energy required for the logic to process in the human cells* [7]. The algorithm in turn accepts the input constraint(s) by the user (if any) and generates an output in increasing order of cost within those constraints. If more than one constraint has been specified, the software displays different sets of circuits, each in increasing order of the

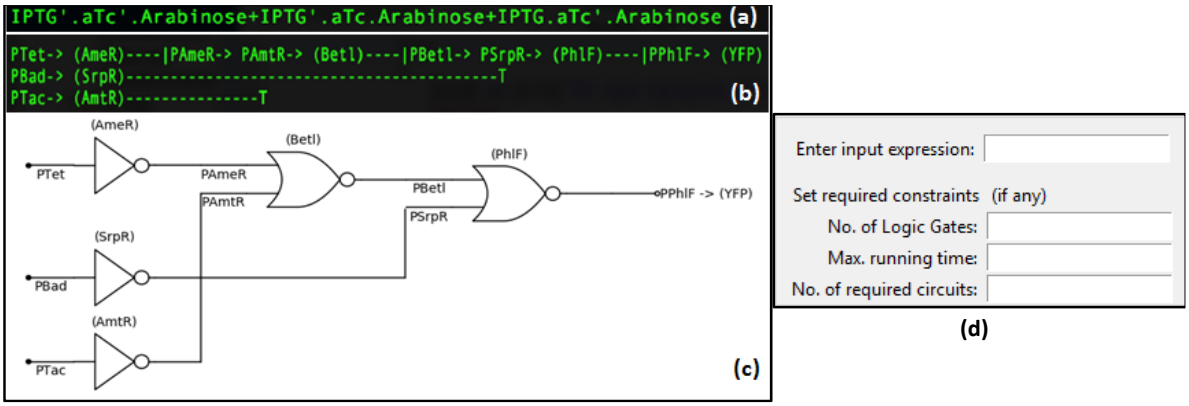


Figure 1. Sample input and outputs of *GeneTech*. (a) Input function to *GeneTech* [10]. One possible logic circuit output generated in (b) the previous version [2] [10]. (c) Additional output diagram for the same input generated in the latest version. (d) A glimpse of the layout of the updated version of *GeneTech* 2.0, allowing users to define the constraints.

cost with respect to a particular constraint, while also displaying the cost for each circuit with respect to other constraints. This allows the user to prioritize between different costs and cherry pick the best output.

Moreover, the users can specify the number of logic circuits possibilities they require. This functionality can have two effects; first, if the user has specified no cost constraints, the program terminates only after producing the required number of circuits, improving its efficiency in time; Second, if the user has specified cost constraints, the program runs through all possible logic circuits and returns only the most cost-effective circuits within the number of logic outputs specified. Additionally, the improved version allows the user to specify the maximum number of logic gates the user wishes to employ, and only produces the circuits if they meet the requirements.

The tool is further upgraded to generate multiple-output circuits as compared to its previous version in which the tool was only able to generate the single-output circuits (as discussed in *Introduction*). Lastly, *GeneTech* 2.0 generates an additional output in the form of logic representation similar to electronic circuits' diagrams, as shown in Figure 1(c).

3. EXPERIMENTATION AND RESULTS

In this work, we tested the same set of circuits which are used in [6]. We verified that *GeneTech* 2.0 eliminates all those circuits (not shown due to space limitations) which contain an *unintended feedback loop* with any of the previous stages, thus giving a more filtered range of outputs as compared to previous version of *GeneTech* [2]. The percentage of circuits eliminated varies from 0-40% for different input expressions, depending on the number of *unintended loops* they contain.

In [8], it has been demonstrated, using D-VASim [11], that the timings of individual circuit components depend upon several design parameters (e.g., degradation rate). Similarly, the components in genetic gates library can be modeled and characterized based on their timing values. These values can then be used by the tool to generate circuits to meet the desired constraints. Since, we do not have the real data, therefore we assigned arbitrary values of time for each logic stage and ran *GeneTech* to generate the list of output circuits. The timings of the generated circuits can be verified using D-VASim, which requires the input in the Systems Biology Markup Language (SBML) format. Therefore, it is necessary that *GeneTech* should generate the circuits in the SBOL format [3] first, which can then be converted to SBML using SBOL-SBML converter [12].

4. SUMMARY

The genetic circuit synthesis process of *GeneTech* software has been significantly improved by making it more user friendly and allowing the users to generate circuits with specific design constraints. The capability of generating the additional output in the form of *logic circuit schematic* would ensure that a wider audience (including design engineers) would be able to benefit from the software.

The functionality of generating a SBOL file, and a SBOL visual representation from *GeneTech* is currently under development. Furthermore, in future, we plan to incorporate the SBOL import for gate libraries as well integration with online repositories. We aim to integrate this functionality in the current release which would allow user to integrate *GeneTech* with SBML-based tools like iBioSim or D-VASim to gain a more enhanced idea about the applicability of the logic circuits that *GeneTech* generates.

5. References

[1]. M.A.Marchisio, *et al*., "Computational design tools for synthetic biology", *Curr. Opin. In Biotech.,* 20, 4, pp-479-485, 2009.

[2]. H. Baig, *et. al.,* "A Top-down approach to Genetic Circuit Synthesis and Optimized Technology Mapping", *IWBDA,* 2017.

[3]. B. Bartley, *et al.*, "Synthetic Biology Open Language (SBOL) Version 2.0.0.," *J. Integr. Bioinform.*, 2015.

[4]. A. A K. Nielsen, *et al.*, "Genetic circuit design automation", *Science*, vol. 352, no. 6281, pp. 7341, 2016.

[5]. T.L. Floyd, 'Digital Fundamentals' 11th Edition, *Pearson Education Limited,* pp.210-216, 2015.

[6]. H. Baig,, "Methods and Tools for the Analysis, Verification and Synthesis of Genetic Logic Circuits", *PhD Thesis*, 2017.

[7]. T. Chiu and J.R. Jiang, "Logic Synthesis of Recombinase Based Genetic Circuits", *Scientific Methods,* 2017.

[8]. H. Baig and J. Madsen, "Simulation Approach for Timing Analysis of Genetic Logic Circuits," *ACS Synth. Biol.*, 2017.

[9]. Jain, T. K, *et al*., "Optimization of the Quine-Mccluskey method for minimization of the boolean expressions", *ICAS,* 2008.

[10]. "*GeneTech*, A Technology mapping tool for Genetic Circuits", Quick Start Guide (QSG) v1.0, DTU, 2017.

[11]. H. Baig, *et al.,* "D-VASim: an interactive virtual laboratory environment for the simulation and analysis of genetic circuits," *Bioinformatics*, vol. 33, no. 2, pp. 297–299, 2017.

[12]. N. Roehner, et. al., "Generating Systems Biology Markup Language Models from the Synthetic Biology Open Language," *ACS Synth. Biol.*, vol. 4, no. 8, pp. 873–879, 2014.

CoRegCAD : a framework from regulatory network to metabolic engineering

Pauline Trébulle*
Micalis Institute, INRA,
AgroParisTech, Université
Paris-Saclay, France
pauline.trebulle@inra.fr

Jean-Marc Nicaud
Micalis Institute, INRA,
AgroParisTech, Université
Paris-Saclay, France
jean-marc.nicaud@inra.fr

Mohamed Elati
iSSB, Génomique métabolique, CEA,
Univ Evry, CNRS, Université
Paris-Saclay, 91057, Evry, France
mohamed.elati@univ-lille.fr

ABSTRACT

CoRegCAD aims at providing a framework for network inference, interrogation and implementation for the rational design of pathway and the metabolic engineering of yeast for the production of compounds of interest in a context-specific manners.

KEYWORDS

regulatory network, genome-scale modeling, computer-aided design, metabolic engineering

ACM Reference Format:
Pauline Trébulle, Jean-Marc Nicaud, and Mohamed Elati. 2018. CoRegCAD : a framework from regulatory network to metabolic engineering . In *Proceedings of 10th International Workshop on Bio-Design Automation (IWBDA).* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

Bio-design automation (BDA) and biological computer-aided design (BioCAD) tools are crucial for the development of synthetic biology and industrial biotechnology which aim at designing and engineering large, self-adaptive, coupled regulatory and metabolic systems at whole-genome scale for useful purposes in a cost-effective manner. Although the landscape of BDA and CAD tools has significantly grown for the last few years [1], in particular regarding the design of complex genetic circuit based on characterized part and specification, tools for context-specific and adaptive rational pathway design are yet to be generalized.

This work aims at providing a framework for the design and optimization of pathways and phenotypes of interest in industrial strains. To meet that goal, our team has developped several building blocks integrated together in CoRegCAD in an iterative process from network inference and interrogation [5] of the strain regulatory process to the integration of genome architecture when re-factoring chromosomes [3]. In this study, we propose to combine regulatory and metabolic network to:

- Identify the best constructions to improve the production yield in context-specific conditions
- Highlight new regulatory elements of interest for further characterization and integration in parts libraries.

This work will be demonstrated on *Yarrowia lipolytica*, a chassis of industrial interest for which standardized Golden Gate modular cloning strategy has been developed [4].

2 MATERIALS AND METHODS

CoRegCAD framework includes several tools working together as represented in Figure 1. From a large dataset, a background gene regulatory network (GRN) is build using the network inference package CoRegNet [5]. This GRN allows to calculate the regulators influence, a sample-specific statistical value corresponding to an estimation of the transcription factors (TF) activities. By integrating the reverse engineered gene regulatory network into the metabolic model (CoRegFlux [2]) and learning from the regulators influence, our model can predict the metabolic genes expression levels in context-specific conditions. These predicted expressions are then converted into constraint for flux balance analysis leading to phenotype prediction and possible calculation of biomass-product coupled yield. Using data from *S. cerevisiae*, we applied our method to a high-dimensional gene expression dataset to infer a background gene regulatory network and compared the resulting phenotype simulations with those obtained by other relevant methods. Our method was shown to have a better performance and robustness to noise and was successfully used to study complex context-specific phenotype such as diauxic shift [2]. More specifically, CoRegCAD aims at providing a set of functions to simulate the engineering of the regulatory network as well as relevant gene knock-out or over-expression. These simulations will then be used to optimize the best constructions to improve

*Also with iSSB, Génomique métabolique, CEA, Univ Evry, CNRS, Université Paris-Saclay, 91057, Evry cedex, France.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
IWBDA, August 2018, Berkeley, CA
© 2018 Copyright held by the owner/author(s).

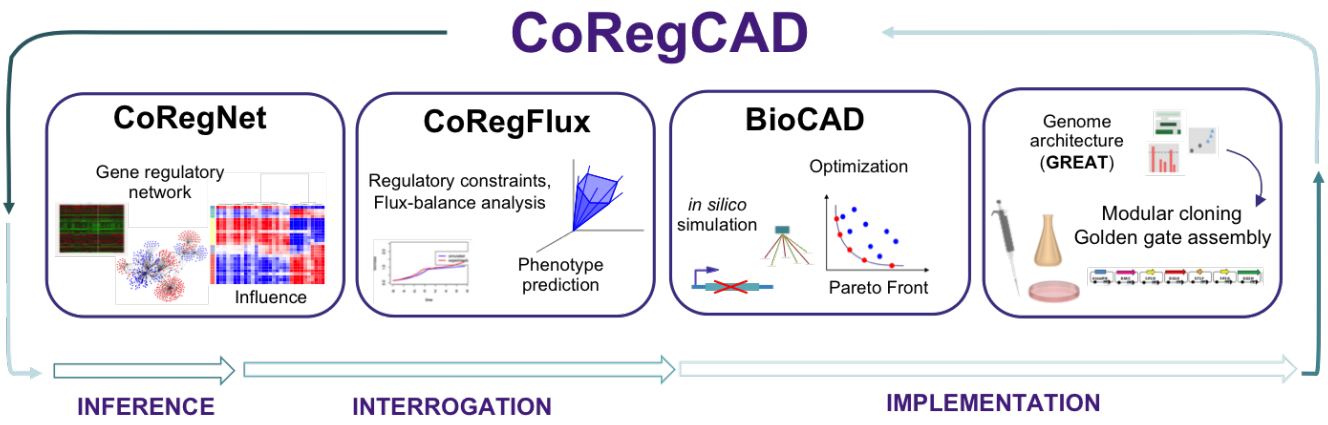


Figure 1: CoRegCAD aims to provide a framework for rational design of pathways and strains metabolic engineering through an iterative process consisting of 1) *Inference* of a co-regulatory network in context-specific conditions 2) *Interrogation* of the network and mapping to the metabolic model to predict genes expression and phenotypes 3) Simulations and optimization to identify the best strategies to improve the product yield and to guide wet-lab experiments for the *Implementation* of the construction. The cycle then start over by improving and refining the network based on experimental observations.

production and to select the most appropriate regulatory element to be included in the expression cassette in the chassis organism. The determination of its optimal insertion point within the genome to maximize the clustering of co-regulated genes will also be considered (GREAT [3]).

3 CASE-STUDY ON AN INDUSTRIAL CHASSIS: *Y. LIPOLYTICA*

To demonstrate the relevance of our strategy for less common organism of industrial interest, these methods will be developed and tested in *Y. lipolytica*, an oleaginous yeast whose metabolism is prone to lipid accumulation under conditions of nitrogen limitation. Following the CoRegCAD framework, a regulatory network consisting of 111 TF, 4451 target genes and 17048 regulatory interactions (YL-GRN-1) was inferred. Interrogation of this network highlighted the relevance of our method to identify several regulatory state corresponding to the yeast adaptation to nitrogen depletion. Using influence, we were also able to identify potential regulators and drivers of lipid accumulation, some of which were tested in the lab with 6 out of 9 being validated for their impact on lipid accumulation [6]. This work will provide proof-of-concept for the context-specific design of metabolic pathways of interest, by improving the yield under specific constraints.

4 CONCLUSIONS

While further development still need to be carry out, CoRegCAD purpose is to provide a framework relying on network inference and interrogation to guide the metabolic engineering of industrial chassis and achieve higher production of metabolite of interest in context-specific conditions. Using CoRegCAD, researchers will be able to reduce time-consuming and costly laboratory effort, to carry out functionalities studies and to identify regulatory element of interest for context-specific expression through the interrogation step and iterative learning process.

ACKNOWLEDGMENTS

The work was supported by a fellowship for PT from the French National Research Agency (ANR) through the IDEX-Saclay, ANR-11-IDEX-0003-02. This work was partially supported by CHIST-ERA grant, AdaLab ANR 14-CHR2-0001-01.

REFERENCES

[1] Evan Appleton, Curtis Madsen, Nicholas Roehner, and Douglas Densmore. 2017. Design Automation in Synthetic Biology. *Cold Spring Harbor perspectives in biology* (2017), a023978.

[2] Daniel Trejo Banos, Pauline Trébulle, and Mohamed Elati. 2017. Integrating transcriptional activity in genome-scale models of metabolism. *BMC systems biology* 11, 7 (2017), 134.

[3] Costas Bouyioukos, François Bucchini, Mohamed Elati, and François Képes. 2016. GREAT: a web portal for Genome Regulatory Architecture Tools. *Nucleic acids research* 44, Web Server issue (2016), W77.

[4] Ewelina Celińska, Rodrigo Ledesma-Amaro, Macarena Larroude, Tristan Rossignol, Cyrille Pauthenier, and Jean-Marc Nicaud. 2017. Golden gate assembly system dedicated to complex pathway manipulation in Yarrowia lipolytica. *Microbial biotechnology* 10, 2 (2017), 450–455.

[5] Rémy Nicolle, François Radvanyi, and Mohamed Elati. 2015. Coregnet: reconstruction and integrated analysis of co-regulatory networks. *Bioinformatics* 31, 18 (2015), 3066–3068.

[6] Pauline Trébulle, Jean-Marc Nicaud, Christophe Leplat, and Mohamed Elati. 2017. Inference and interrogation of a coregulatory network in the context of lipid accumulation in Yarrowia lipolytica. *NPJ systems biology and applications* 3, 1 (2017), 21.