Software Systems for Automated Manufacturing of Engineered Organisms

Justin Vrana

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:
Eric Klavins, Chair
Georg Seelig
Hao Kueh
Luis Ceze

Program Authorized to Offer Degree:
Bioengineering

University of Washington

Abstract

Software Systems for Automated Manufacturing of Engineered Organisms

Justin Vrana

Chair of the Supervisory Committee:
Professor and Chair Eric Klavins
Electrical and Computer Engineering Department

Organism engineering is a field that employs principles of genetic engineering, synthetic biology, and bio-manufacturing to rapidly test and produce new genetically modified organisms at very large scales. Automation systems are often employed at these large scales to manage and execute laboratory operations. A current challenge in the field is how to automate organism engineering during the discovery phase of research when experimental failure rates are high and laboratory methodologies are variable. Here we demonstrate progress towards creating automated systems for rapidly engineering new strains of *S. cerevisiae*, or baker's yeast, during the discovery phase of research. This dissertation details (i) the design and construction of a new set of CRISPR-based genetic parts that can be used to engineer new behaviors in yeast, (ii) accompanying lab software that can be used to automate the engineering of new yeast strains, and (iii) applications of lab automation software to global health.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PUBLICATIONS

Publication	**Contribution**	**Status**
Digital logic circuits in yeast with CRISPR-dCas9 NOR gates	Second author. Built boolean circuit gates. Modeling, fits, and parameter sensitivity. Build inducible cascades and dose-response inducible measurements.	Published Nature Communications 2017
OLA-Simple: a software-guided HIV-1 drug resistance test for low-resource laboratories	Third author. Wrote software for automated protocol execution using Aquarium software system.	Published EBioMedicine 2019
Aquarium: open-source laboratory software for design, execution and data management	First author. Wrote code for Python API. Aquarium debugging and protocol development. Co-wrote paper.	Published Oxford Synthetic biology 2021
Implementation of an interactive mobile application to pilot a rapid assay to detect HIV drug resistance mutations in Kenya	First author. Wrote software for automated protocol execution using Aquarium software system. Conduct pilot study in Nairobi, Kenya. Collected experimental data.	Preprint at MedRXiv 2021; submission in progress
Simpler and faster Covid-19 testing: Strategies to streamline SARS-CoV-2 molecular assays	Co-author. Paper conception. Experiment planning.	Published EBioMedicine 2021
Integration of Design, Planning and Execution for Synthetic Biology Engineering	First author. Project conception. Wrote software, wrote paper, developed figures.	Planned submission, ACS Synthetic Biology 2021

Table 1: List of Publications

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor, Prof. Eric Klavins for all of his thoughtfulness, intelligence and providing me with freedom and opportunity to explore and learn over the years. His brilliance and unique approach to solving problems was contagious. I also would like to thank my committee members Barry Lutz, Georg Seelig, Hao Kueh, Luis Ceze for their thoughtful advise and feedback. I would especially like to thank Barry for opening my eyes to other opportunities and applications in the field of global health and providing me the opportunity to work in an area that was completely new to me.

I could not have done this without my friends and colleagues, who have provided endless emotional and intellectual support throughout the years: Alberto Carignano, Orlando de-Lange, Mile Gander, David Younger, William Voje, Arjun Khakhar, Yaoyu Yang, Leandra Brettner, Nick Bolten, Eriberto Lopez, Cannon Mallory, and Nuttada Panpradist. They made working a joy and I have so many good memories rock climbing, skiing, and pizza parties with all of them. I would especially like to thank Nuttada, who really was a dear friend and an unofficial graduate school mentor over the years; our collaboration together spurred inventive research, publications, and fun/stressful commercialization efforts.

I would like to give a special thanks to current and former lab managers, Michelle Parks, Cami Cordray, Samer Halabiya for the friendship and dedication in the lab; they were really the glue that held much of the lab together. They also were extraordinarily patient as I attempted (often with disastrous consequences) to automate various procedures in the lab.

Finally, I would like to thank my family for all of the support over the years: my mother and father, Sharon and Erv Vrana and my sister, Lindsay Soine. Dear friend Sarah Capser, for all of her support and mountain trips (and taking care of my dog). Lastly, I would like to

thank my dear Tamara Casper. Her patience and support made all of this possible. Truly, I would not have been able to do this without her.

Chapter 1

INTRODUCTION

Synthetic biology is an engineering discipline that aims to harness the power of biology to engineer new microbes, crops, and mammalian cells with unprecedented precision and functionality. These highly engineered biological systems have the potential to transform how we produce energy, synthesize biologics, grow crops, treat disease, and transform animal populations [1–4]. This type of high-level organism engineering requires a deep understanding of how biological systems function on molecular, cellular, and systems-level scales. However, synthetic biology suffers from a lack of understanding of how natural biological systems work. The challenge is that many biological systems are enigmatic and the underlying chemistry that governs their systems-level behaviors are inherently stochastic and noisy. Establishing underlying principles from seemingly complicated systems and noisy data is a challenge that undermines our ability to predict biological behavior and design new biological systems [5]. As a result, many predictive models for biological systems break down when using different organisms, genetic parts, or experimental conditions. Many of these difficulties remain present even when engineering well-studied host organisms such as baker's yeast (*S. cerevisiae*) or common mammalian cell lines (HEK293, CHO, etc.) and are exacerbated further when moving to less studies organisms [5, 6].

Because of the difficulties in predicting biological phenomena, industrial-scale engineering of organisms is becoming increasingly data-driven, often requiring the construction and testing of thousands of genetic variants. To operate at large scales many labs employ robotics and instrumentation automation that perform assembly-line type workflows on well-validated techniques [7–10]. These techniques borrow on concepts of industrial manufacturing, treating biomolecules and cells in a similar way as a car factory may treat steel and plastic. Distinct

molecular and cellular manipulations can be abstracted as a manufacturing operation, which takes in several materials and produces a new bio-material, such as a new cell. Several of these operations can be connected together in series or parallel and performed by a combination of humans, robots, and instruments. A lab automation system comprises several of these automation techniques into a coherent system [11, 12]. Lab automation systems can be seen prominently at synthetic biology companies such as Ginkgo Bioworks or Strateos (formerly Transcriptic), which utilizes software and an army of robot and human operators to execute hundreds to tens-of-thousands of laboratory operations in an assembly-like fashion.

While lab automation systems can automate strict laboratory procedures with high capacity, it remains challenging to automate workflows during the exploratory phase of many research projects. During this exploratory phase of research, design parameters are unknown, experimental failure rates are high, and methodologies are continuously changing. State-of-the-art lab automation systems are often unable to rapidly adapt to these rapid changes in workflows and techniques because instruments and robotics are often difficult to reconfigure. Additionally, workflows in an automated assembly line often must be redesigned, re-tweaked, and re-tested with even slight changes [13]. These problems are especially acute in biology, where slight changes in methodology may drastically affect results. Additionally, high failure rates in experiments often require processes to be repeated, known in manufacturing literature as rework loops [14–16]. Rework loops in many manufacturing systems utilize material recycling to mitigate material loss. However existing lab automation systems often do not have this recycling capacity as the properties of biological samples are often complex and hierarchical; existing systems simply do not have sufficient knowledge of how to reuse biological materials. In summary, exploratory research is difficult to automate because it suffers from high workflow variation, frequent rework loops, and material recycling.

The processes involved in exploratory research resemble features of a reconfigurable manufacturing system. Reconfigurable manufacturing system (RMS) are systems that utilize highly modular and reconfigurable machines (e.g. a CNC machine) that can rapidly adapt their capacity and functionality to meet changes in production needs [17–19]. Due to the

modular nature of the machines used in an RMS, a computer-aided process planner (CAPP) is sometimes employed to efficiently route materials and generate production workflows [20]. Variants of RMSs, CAPPs, and similar systems enable factories to rapidly change manufacturing workflows, products, and utilize material reuse [21, 22].

The chapters that follow describe algorithms and software that apply principles of reconfigurable manufacturing and computer-aided process planning to the field of synthetic biology to create an automated organism engineering system. Specifically, these techniques will be applied to a well-studied organism, *S. cerevisiae*, also known as baker's yeast. Yeast are easy to genetically manipulate and are incredibly efficient at producing biomolecules and biofuels. They are utilized heavily in industry for such purposes, which makes them an attractive choice to test the proposed manufacturing system [23–27]. While yeast are valuable in themselves to engineer, components developed in this system could potentially be applied to other organisms as well, such as plant or human cells [28, 29].

Just as industrial manufacturing systems require raw material to generate products, organism engineering requires modular and predictable parts from which to construct new cells. Since genetic circuits are an essential component of engineering organisms, chapter 2 will be dedicated to characterizing and designing genetic circuits. In this chapter, I describe the construction and characterization of a modular set of synthetic gene parts for *S. cerevisiae*. The parts are based on the deactivated RNA-guided endonuclease, CRISPR-associated protein 9 (CRISPR-Cas9) [30–34]. I demonstrate how these parts can be constructed into complex boolean logic circuits in yeast. I construct ODE (ordinary differential equation) models of their behavior from steady-state and kinetics data. Using these models in conjunction with Monte-Carlo simulations, I predict the bounds of operation for these parts. In chapter 3 I evaluate several limitations of using the CRISPR-Cas9 system in genetic circuits. I describe here how the shallow response of CRISPR-Cas9 based genetic parts limits their use in both large genetic circuits and in sequential circuits. Using *in silico* simulations, I demonstrate the difficulty in creating bistable or hysteretic systems using the CRISPR-Cas system in yeast.

In chapter 4, I port the CRISPR-Cas9 system to mammalian cells. Here I design and report a new CRISPR responsive promoter based on the Pol III promoter. I tested the new promoter part by creating a CRISPR OR-circuit in human embryonic kidney (HEK) and Chinese hamster ovary (CHO) cells. Here I report, for the first time, the ability to regulate Pol III using CRISPR-dCas9. This new class of CRISPR parts could form the basis for more complex circuitry in mammalian cells.

In chapters 5, 6, and 7 I describe a laboratory automation system, Aquarium, that serves as a computer-aided manufacturing (CAM) and a corresponding software, Terrarium, that serves as a computer-aided process planner (CAPP). In this work, I describe the first reported CAPP tool applied to biology. I describe algorithms and software for automated workflow planning and strain construction and demonstrate the ability to convert abstract cell design specifications into executable workflows that physically construct the specified cells.

In chapter 8, I describe global health applications of Aquarium, which can be used to democratize complex laboratory procedures by allowing untrained users to carefully follow on-screen instructions provided by a mobile app I created based on the Aquarium system. As part of a research study and commercialization effort, colleagues and I conducted a pilot study in Nairobi, Kenya. We used this application to evaluate its utility to train users to perform an HIV drug resistance test.

Throughout this work, I demonstrate unique applications of laboratory automation and manufacturing principles and apply them to synthetic biology. Over the last few years, I have tried to the best of my ability to identify the most important bottlenecks I see in synthetic biology and attempt to solve them. The problems I attempt to solve revolve primarily around how to realize biological designs in the laboratory. The conversion of a design into processes that realizes the design is the seminal step in manufacturing. This design-to-build process is, in my opinion, an oft-ignored process in synthetic biology. While thousands of studies are published every year on the development and characterization of new genetic parts or new computational design tools, far fewer studies have been released on how to use these genetic parts to realize biological designs in a way that is scalable and generally applicable. This

work represents an attempt to automate and resolve issues in converting digitally represented biological designs into physical products.

Chapter 2

ENGINEERING EUKARYOTIC CELL BEHAVIORS USING RNA-GUIDED CRISPR DCAS9

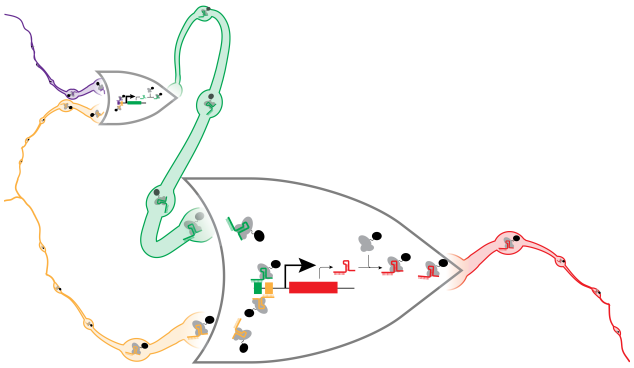


Figure 2.1: Artistic depiction of CRISPR-dCas9 enable genetic NOR gates

In this chapter, I describe the engineering and characterization of CRISPR-based genetic circuits in *S. cerevisiae*. These parts form the basis of a platform for re-engineering cells with new behaviors. This work was in collaboration with my former colleagues Miles Gander and William Voje and was presented in Gander et. al., Nature Communications 2017 [31].

Author	Contribution
Justin Vrana	Design and construction of NOR, OR, AND, XOR, XNOR, NAND circuits; ODE modeling, fitting; data analysis; parameter sensitivity analyses; mathematical analyses; measurement of steady-state repression cascade data
Miles Gander	Development of CRIPSR NOR gate architecture; construction of XNOR circuit; construction and measurements of repression static cascades; data analysis; measurement of kinetic repression cascade data
William Voje	Design of gRNA target sequences; design of ribozyme sequences

2.1 Summary

Natural cells perform incredible feats of computation through an orchestra of interconnected and interacting molecules. A grand challenge in synthetic biology is to emulate the complexities of natural cellular systems to create new sophisticated cellular behaviors through the introduction of synthetic molecular functions. The creation of new molecular functions that are orthogonal, inter-connectable, and scalable is essential for engineering complex molecular networks. However, many natural molecular functions often exhibit unpredictable behaviors when used outside of their natural genetic context, impeding the engineering of grand molecular networks in vivo. To address these issues, we engineered a set of unnatural transcriptional components in *S. cerevisiae* that can be interconnected and assembled into large genetic circuits. Our system is comprised of a set of twenty orthogonal synthetic transcriptional CRISPR/dCas9 'NOR' gates. We leveraged an RNA-guided synthetic chromatin remodeler, a dCas9-Mxi1 fusion protein, to substantially improve dCas9 guided transcriptional repression and used computer-aided design to create a set of orthogonally interacting guide-RNAs and promoter components. Since all of the components are orthogonal to the native *S. cerevisiae* host, our transcriptional system exhibits exceptional predictability and scalability with minimal cell loading. We demonstrate the power of our synthetic NOR gates by assembling them into large circuits that perform basic combinatorial logic and long delay cascading circuits. Complementing this system, we also constructed a model that predicts synthetic network behavior through global parameter optimization and Monte Carlo simulation. The model allows for predictable forward-engineering of synthetic gene circuits through dynamic behavior prediction and examination of parameter sensitivity estimation. Our approach has enabled the construction of among the largest eukaryotic gene circuits to date. We envision future iterations of the system will form the basis for complex, synthetic, decision-making systems in living cells.

2.2 Introduction

A natural step towards engineering complex transcriptional networks is to create and characterize modular parts that can be composed into larger networks, in a way analogous to electronic components. A general strategy towards this end is to utilize RNA polymerase flux on a gene promoter as a 'signal' that can be transmitted to other promoters through a gene regulatory network (GRN). Promoters in a GRN produce mRNA transcripts that are translated into transcription factors (TFs). TFs can bind in a sequence-specific way to DNA, which can result in activation or repression of other promoters in its vicinity through a myriad of mechanisms[35]. Changes in the RNA polymerase flux of promoters expressing TFs eventually result in differences in the number TFs being produced; likewise, TFs are eventually removed from the cell via protein degradation [36] and cell division. Basic assumptions about networks of TF binding can be treated as a deterministic chemical reaction network (CRN) [37] and ordinary differential equations (ODEs) can be used to model the behavior of GRNs. Predictions based on these models can be used to design GRNs that perform a specific behavior within a cell.

Programming a cell using a gene regulatory network requires highly specific and orthogonal molecular interactions. A common strategy to make modular molecular interactions is to fuse a known protein that binds specifically to a DNA sequence to a transcription factor. By engineering protein binding sites in a DNA promoter sequence, one can specifically recruit desired transcription factors to specific promoters. An organism can be engineered with a new behavior by composing multiple promoters, DNA binding proteins, and transcription factors onto DNA and integrating the DNA into the genome of an organism.

Toward the goal of creating and characterizing genetic parts to be used for engineering cell behaviors, other research groups have previously used programmable DNA binding domains, such as zinc fingers (ZNF) and TALES to target transcription factors to control gene expression in eukaryotes [38–41]. However, these systems suffer from scalability issues as it is difficult to engineer new ZNF and TALES [42]. For prokaryotes, such as E. coli, orthog-

onal repressors have been mined from known protein databases and used as orthogonal and modular repressors to build large synthetic gene networks [43].

An alternative to modular protein-based transcription factors is the RNA-guided CRISPR systems. The Type II CRISPR-Cas system is a family of proteins found in bacterial immune systems that bind to specialized RNA molecules to specifically cleave DNA sequences that are complementary to a portion of the RNA. The best-studied CRISPR protein is Cas9 derived from the bacteria *S. pyogenes* [44]. In the native *S. pyogenes* organism, Cas9 binds two small RNA molecules named CRISPR RNA (crRNA) and trans-activating CRISPR RNA (tracrRNA). Together, crRNA and tracrRNA form an RNA secondary structure to which the Cas9 protein binds. Once bound, the Cas9:crRNA:trRNA complex can cleave DNA in a sequence-specific way. Sequence recognition occurs when the Cas9 complex first recognizes a small 1-6bp protospacer adjacent motif (PAM) sequence on one strand of DNA. If there exists complementarity between the 20bp sequence on the crRNA (known as a "spacer") and the DNA immediately upstream of the PAM site, DNA cleavage occurs [45]. In the past few years, there have been dozens of CRISPR orthologs characterized from various organisms that have different PAM, crRNA, and trRNA requirements [46–48].

From an engineering perspective, CRISPR systems provide a way to program DNA cleavage in a highly specific way. Due to the flexibility and programmability of Cas9, there has been an explosion of research on using it as a genome-editing tool [49–51]. The system can be further modified to do transcriptional programming. Two point mutations (D10A and H840A) on the Cas9 protein create a catalytically-dead Cas9 protein (dCas9) which can be used to programmatically recruit specific proteins to DNA. CRISPR activation (CRISPRa) or CRISPR interference (CRISPRi) can be used to activate or repress specific promoter sequences depending on which types of proteins are fused or bound to the dCas9 complex. CRISPRi has previously been used to create simple logic functions and to interface with host regulatory networks [52]and to interface with the host cell machinery. Further, RNA molecules have been re-engineered to be able to recruit RNA-binding proteins, allowing more flexibility in the way dCas9 can recruit transcription factors [53].

While the ease of programming CRISPR/dCas9 solves scalability issues, CRISPRi suffers from low cooperativity which affects the 'steepness' of its response curve when studied in E. coli [52]. This lack of 'steepness' affects the ability to use CRISPRi in deeply layered GRNs. When designing gene regulatory networks, it is common to treat gene transcriptional levels as binary signals; that is to treat a low transcriptional level as a "0" and a high transcriptional level as a "1". A well-engineered GRN will properly propagate 0s and 1s from each genetic part to the next. If a genetic part has a low cooperativity in its response, signals separate from their canonical 0 and 1 values, resulting in signal loss. This problem becomes amplified in deeply layered GRNs, as signals drift away from their binary values with each signal propagation step in the network. Because of the analogy between GRNs and electronic circuits, GRNs are often called genetic circuits.

Addressing these issues with CRISPRi, we combined the CRISPRi system with a powerful epigenetic silencing protein from humans, Mxi1 [54] The chromatin-remodeling motif, Mxi1, was fused directly to dCas9 to enable almost complete silencing of its cognate promoters, which we hypothesized would allow more deeply-layered synthetic gene regulatory networks to be created. To create deeply-layered genetic circuitry, we created an *in vivo* synthetic NOR gate in baker's yeast *S. cerevisiae* that leverages the power and programmability of the CRISPR/dCas9 system (2.2). As NOR-gates are functionally complete Boolean operators, any conceivable Boolean function can be expressed as a circuit composed entirely of NOR-gates. Hence, it is conceivable that complex Boolean functions can be implemented in a cell using the orthogonal transcriptional wires and logic gates created. Briefly, modular promoters were created to drive the expression of guide RNA (gRNA) molecules, which then binds to constitutively expressed and enzymatically deactivated dCas9 fused to a chromatin-remodeling protein Mxi1. These guide RNA:dCas9-Mxi1 complexes then target further downstream promoters in a sequence-dependent manner, repressing transcription at that promoter. Downstream promoters are further designed to express their own unique gRNA molecules, which can be programmed to target other promoters. Hence, promoters can be composed together to create complex transcriptional networks comprised of modular

promoters and gRNA:dCas9-Mxi1 signaling molecules.

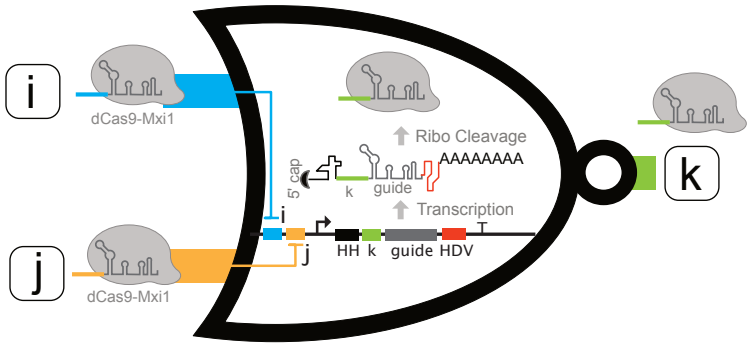


Figure 2.2: **CRISPRi dCas9-Mxi1 NOR gate architecture** Representation of the CRISPRi NOR gate implemented in *S. cerevisiae*. Conventionally, a NOR gate is a boolean logic gate with two inputs and one output. A NOR gate output is low or "0" only when both of its inputs are also "0"; the output is high or "1" in all other cases. Here, many synthetic genetic NOR gates can be integrated into a yeast cell genome to create a combinatorial genetic circuit (left). Each CRISPRi NOR gate is comprised of a synthetic gene promoter with two engineered guide RNA target sequences, labeled here as i and j; the engineer promoter is called a pGRRij. The promoter expresses a guide RNA flanked by self-cleaving RNA sequences called ribozymes, which cleave the 5'-cap and polyA tail from the guide RNA transcript. The ribozyme-gRNA-ribozyme cassette is an RGR. After transcription and ribozyme processing, the nuclearly-retained gRNA is produced with a new targeting sequence k. The gRNA k binds to a dCas9-Mxi1 and is free to repress other promoters. Once bound to another promoter, the dCas9-Mxi1 is hypothesized to recruit chromatin remodeling machinery to cause silencing of its bound promoter. A genetic circuit is created when several of these genetic parts are connected together to form a network.

To create large complex networks composed of smaller modular parts, we created a synthetic $\text{NOR}_{i,j,k}$ gate that responds to signals r_i and r_j and outputs a signal r_k. The synthetic $\text{NOR}_{i,j,k}$ gate is comprised of a gRNA-responsive Pol II promoters ($\text{pGRR}_{i,j}$) that responds to gRNA:dCas9-Mxi1 input signals, denoted as r_i and r_j and an gRNA expression system (RGR_k) that expresses a signal r_k. To efficiently express single-stranded gRNAs, $\text{pGRR}_{i,j}$ promoters are engineered to express cis-cleaving ribozyme-flanked gRNA (RGR_k). Transcription of the RGR_k construct results in self-cleavage of the 5' and 3' flanking ribozymes, removing the poly-A tail and 5' cap, leaving a gRNA molecule that is retained in the nucleus, capable of complexing with a dCas9-Mxi1 and repressing the cognate promoter. The system

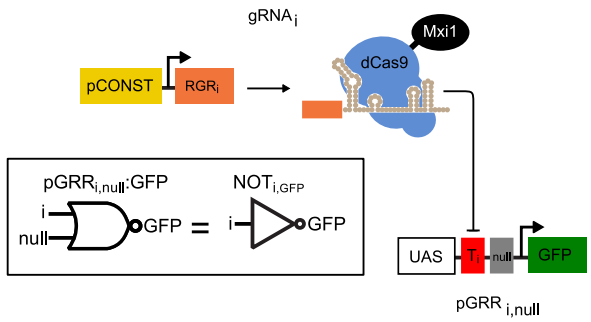
of a $pGRR_{i,j}$ promoter driving a RGR_k signal comprises a synthetic $NOR_{i,j,k}$ gate. The gRNA:dCas9-Mxi1 signals were shown to have programmability and scalability (Fig. 2.3).

2.3 Results

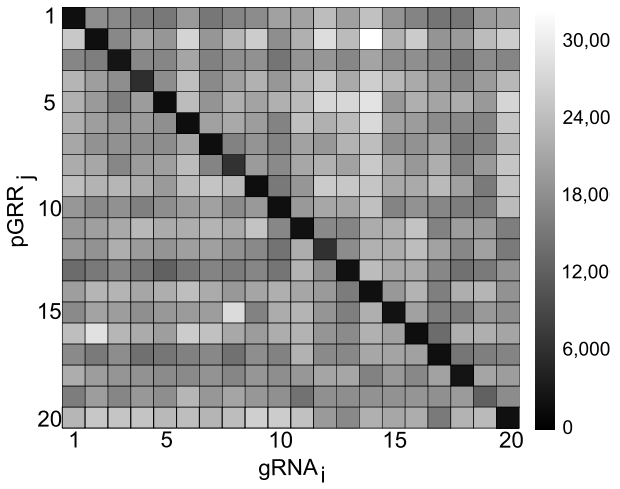
2.3.1 Performance of operational logic gates

To test the utility of using $NOR_{i,j,k}$ gates for creating large transcriptional networks, several canonical two-input logic operations (representing NOT, OR, AND, NAND, XOR, and XNOR logic) were implemented in vivo by integrating several NOR gate cassettes into different genomic loci in the yeast genome (Fig 2.4). NOR gates were designed from gRNA sequences from a twenty-component library that exhibited the strongest repression. The bipartite dCas9-Mxi1 transcription factor was shown to be highly effective at transcriptional repression in the region surrounding the transcriptional start site (TSS). Three gRNA targeting positions surrounding the TSS were tested and we observed high repression at all positions dCas9-Mxi1 was found to yield high repression at all positions, in contrast with dCas9 steric interference (Fig. 2.5). This result is corroborated by Smith et al. 2016, who reported in a library screening study in yeast that dCas9-Mxi1 has effective repression when targeted to the -200 to +50bp region surrounding the TSS[55].

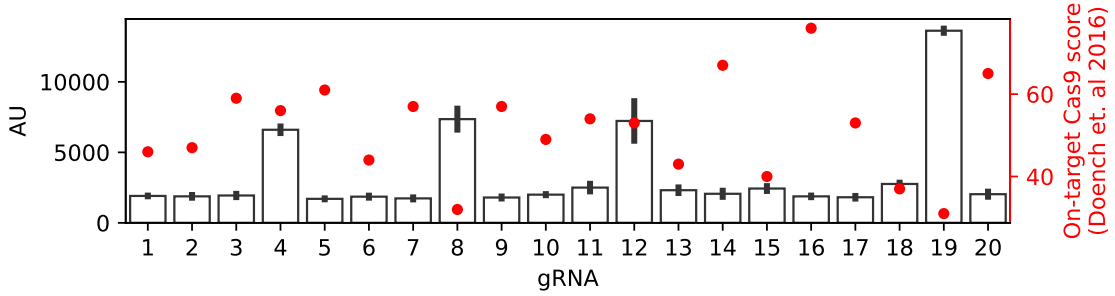
In order to demonstrate the composability of the NOR gates, complex functions were incrementally created by extending simpler functions. The NOR gate in each circuit expresses a GFP reporter so that the output for each circuit can be evaluated. Inputs to each of the two-input logic circuits are composed of an RGR cassette driven by a strong ADH1 promoter and were stably integrated into the genome. GFP was measured from the yeast strains in liquid culture at the mid-log phase using a cytometer. We successfully implemented all canonical two-input logic gates as determined by the fluorescence of the GFP reporter (Fig. 2.4). The results demonstrate the ability to construct more complex functions from simpler NOR molecular functions.



(a) CRISPR NOT gate mechanism



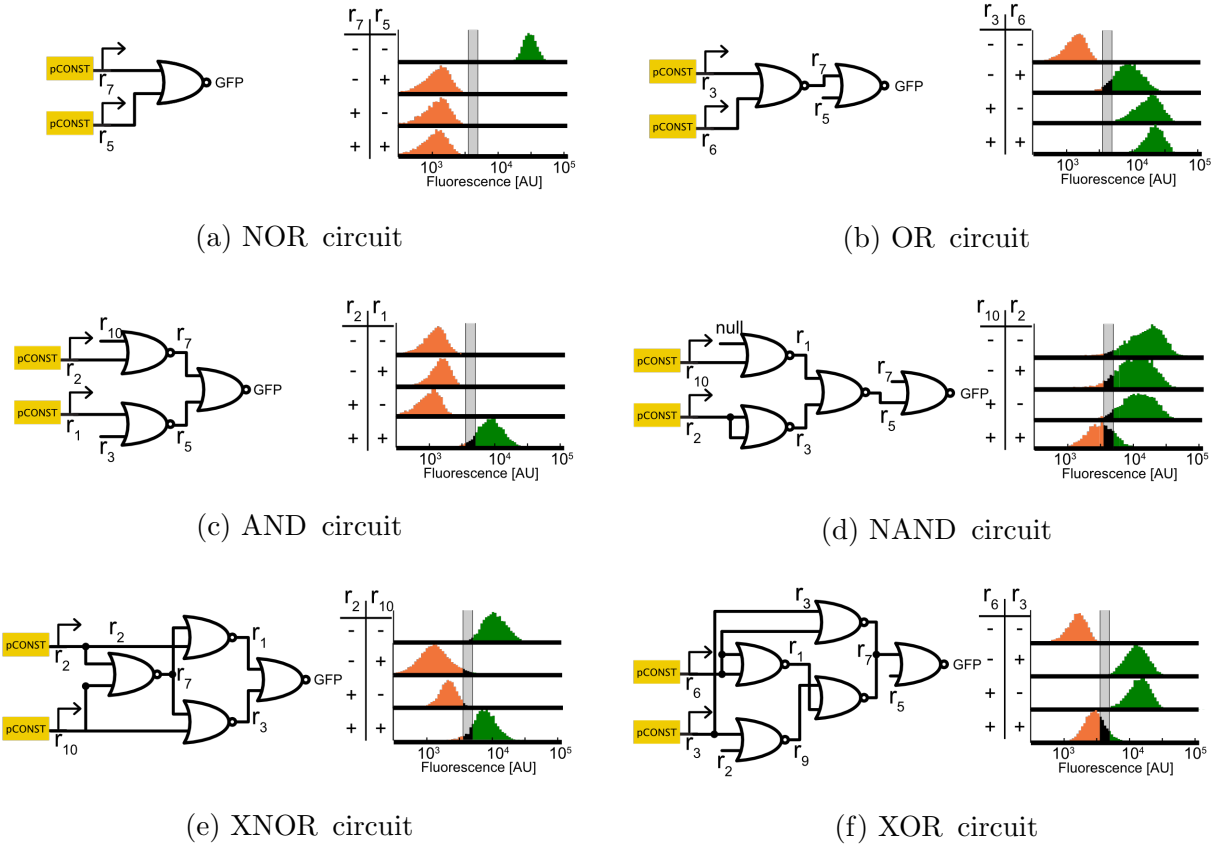
(b) gRNA orthogonality matrix



(c) on-target repression

Figure 2.3: **Orthogonality of gRNA-guides dCas9-Mxi1 repression** (a) Synthetic guide RNAs expressed from a constitutive promoter complex with dCas9-Mxi1and causes repression of its cognate promoter. (b) 20 different 20 bp target sequences were created and all 400 interactions were evaluated in *S. cerevisiae*. Darker squares along the diagonal of the heatmap represent strong repression between guide sequences and their expected targets on the promoter. All other squares are 'off-targets'. (c) Left axis (black) Measured on-target repression (diagonal of b). Right axis (red): On-target predictions gRNA target and PAM sequence [56]. Measured dCas9-Mxi1 repression has no apparent correlation with software predicted on-target activity of Cas9 indicating variance likely is not solely dependent on the target-PAM sequence, but perhaps due to variations in promoter transcription, ribozyme cleavage, or gRNA scaffold folding. Other results with alternative software predictions also yielded no correlations (not shown).

A surprising result was the variability seen in more complex circuits. As an example, the XOR gate (which consisted of 7 individual expression cassettes and up to 8 gRNA:pGRR interactions) and took 15 iterations to construct a suitable XOR behavior displayed (Fig 2.6). It should be noted that part selection was informed from an orthogonality matrix (Fig. 2.3) and only those gRNAs that showed almost no off-target effects were chosen. To troubleshoot the circuits, we attempted to use quantitative PCR (qPCR) to investigate the internal expression of the gRNA signals. However, distinguishing gRNAs proved to be difficult as each gRNA signal is molecularly identical with the exception of the 20bp target site on the 5' end, giving little room for primer design. In many cases, qPCR failed to amplify gRNAs in many of our test runs (data not shown). Instead, careful evaluation and construction of the circuits allowed us to identify poorly performing NOR gates that resulted in circuit failure and to eliminate their use in future circuits; this includes the creation of several test strains not used in the final circuits. Due to the process of sequential integration, in total around 300 strains of yeast were constructed to generate the gates described in Fig. 2.4.



(a) NOR circuit

(b) OR circuit

(c) AND circuit

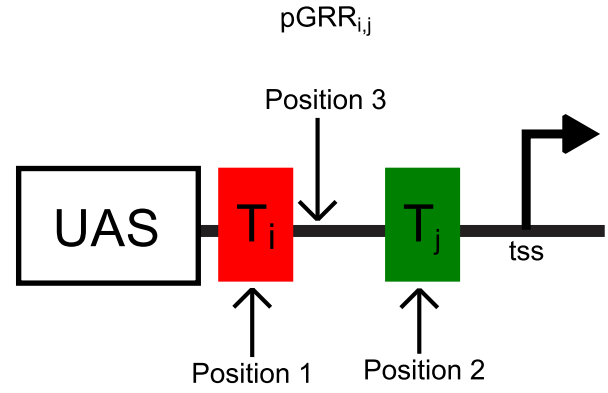
(d) NAND circuit

(e) XNOR circuit

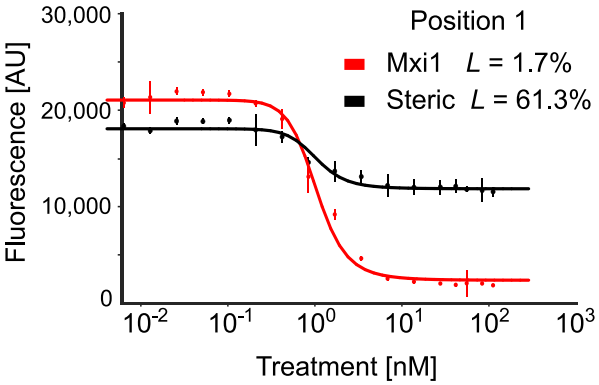
(f) XOR circuit

Figure 2.4: **Combinatorial Circuits in *S. cerevisiae*** (a–f) Six different two-input combinatorial logic circuits were constructed by interconnecting CRISPRi NOR gates and integrating them into a yeast strain constitutively expressing a dCas9-Mxi1 expression cassette. CRISPRi NOR gates are displayed as curved boxes using standardized boolean logic notation and intended gRNA:promoter interactions are displayed as wires connecting the NOR gates. Interactions are labeled as "r1" through "r20" representing their gRNA target sequence identity. The rightmost NOR gate expresses a GFP fluorescence reporter. For each of the four input possibilities ($--$, $-+$, $+-$, and $++$), a distinct strain was constructed with the corresponding inputs expressed off of constitutive promoters (for logical $+$), or not integrated at all (for logical $-$). Fluorescence values were collected using flow cytometry of cells growing in the log phase. The histograms represent population fractions from three different biological replicates measured during a single experiment and were normalized so that area sums to unity.

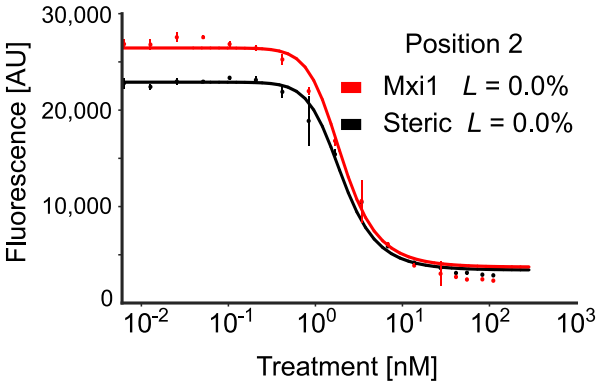
The variance of our two-input logic circuits indicated that our $\mathrm{NOR}_{i,j,k}$ gates were not as robust as we previously thought. Each NOR gate is composed of a nearly identical DNA sequence with the only variance occurring at the input target sites i and j on the $\mathrm{pGRR}_{i,j}$



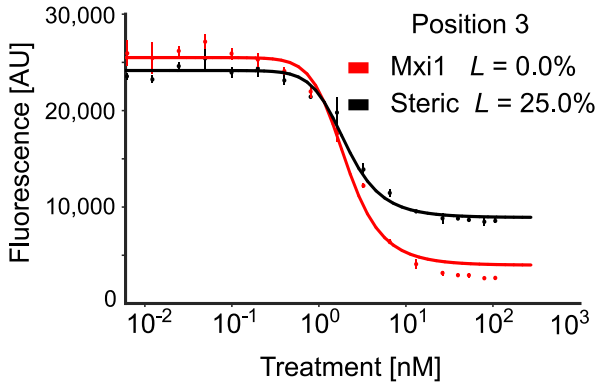
(a) Positions of gRNA target sites



(b) Position 1 (T_i)



(c) Position 2 (T_j)



(d) Position 3

Figure 2.5: **Comparison of dCas9 vs dCas9-Mxi1 repression** Dose-response curves were measured for dCas9-Mxi1 repression when targeted to 3 positions between the upstream-activating sequences (UAS) and the transcriptional start site (TSS). A beta-estradiol inducible promoter was used to induce the expression of a gRNA that targeted a promoter expressing GFP. Beta-estradiol vs fluorescence is plotted. The dCas9-Mxi1 system was shown to be position-independent, while steric hindrance (dCas9 without Mxi1) was position-dependent.

promoter and the target site on the gRNA in the RGR_k. In total, only 60bp out of the 1228bp (5%) is different between the $\text{NOR}_{i,j,k}$. It was possible that NOR gate performance was compromised due to cross-talk of gRNA to non-cognate $\text{NOR}_{i,j,k}$ gates. However, the results of the orthogonality matrix of the NOR gates seem to indicate this is not the case.

Instead, we hypothesized that the difference in performance was due to slight sequence variations of the $\mathrm{NOR}_{i,j,k}$ gate. We hypothesized that variations at sites i and j of the $\mathrm{pGRR}_{i,j}$ would result in slight changes in transcription strength of the NOR gate promoter not entirely captured in the orthogonality matrix (Fig. 2.3). Indeed, a measurement of several $\mathrm{NOR}_{i,j,\mathrm{GFP}}$ gates indicated that a variation of 18.1% (coefficient of variation; n=11). Additionally, we hypothesized that variances in the target site of the RGR_k would result in slight variances of the repression strength of each gRNA:dCas9-Mxi1 on its cognate site and that this difference would be amplified in larger circuit contexts resulting in circuit failure. To evaluate these hypotheses, we decided to model our system to better characterize the performance and variability of our NOR gates.

2.3.2 Modeling and performance characterization using repression cascades

To evaluate the performance of the synthetic NOR gates in deep, multi-layered circuits, we constructed several inducible signaling cascades. Leaving one of the two inputs unconnected (Null), the $\mathrm{NOR}_{i,j,k}$ gate effectively turns into a $\mathrm{NOT}_{i,k}$ gate. Hence by 'wiring' the i input and k output of several $\mathrm{NOT}_{i,k}$ gates together, we can create long repression cascades. Repression cascades composed of $\mathrm{NOT}_{i,k}$ gates were wired in serial to create repression cascades of increasing length, composed of one, two, three, or four layers of interacting gRNA signals. A β-estradiol inducible promoter pGALZ4 was used to activate the transcription of the input gRNA. As described previously, all components of the cascades were stably integrated into the genome of *S. cerevisiae*. In total four strains of yeast were constructed, one for each cascade constructed.

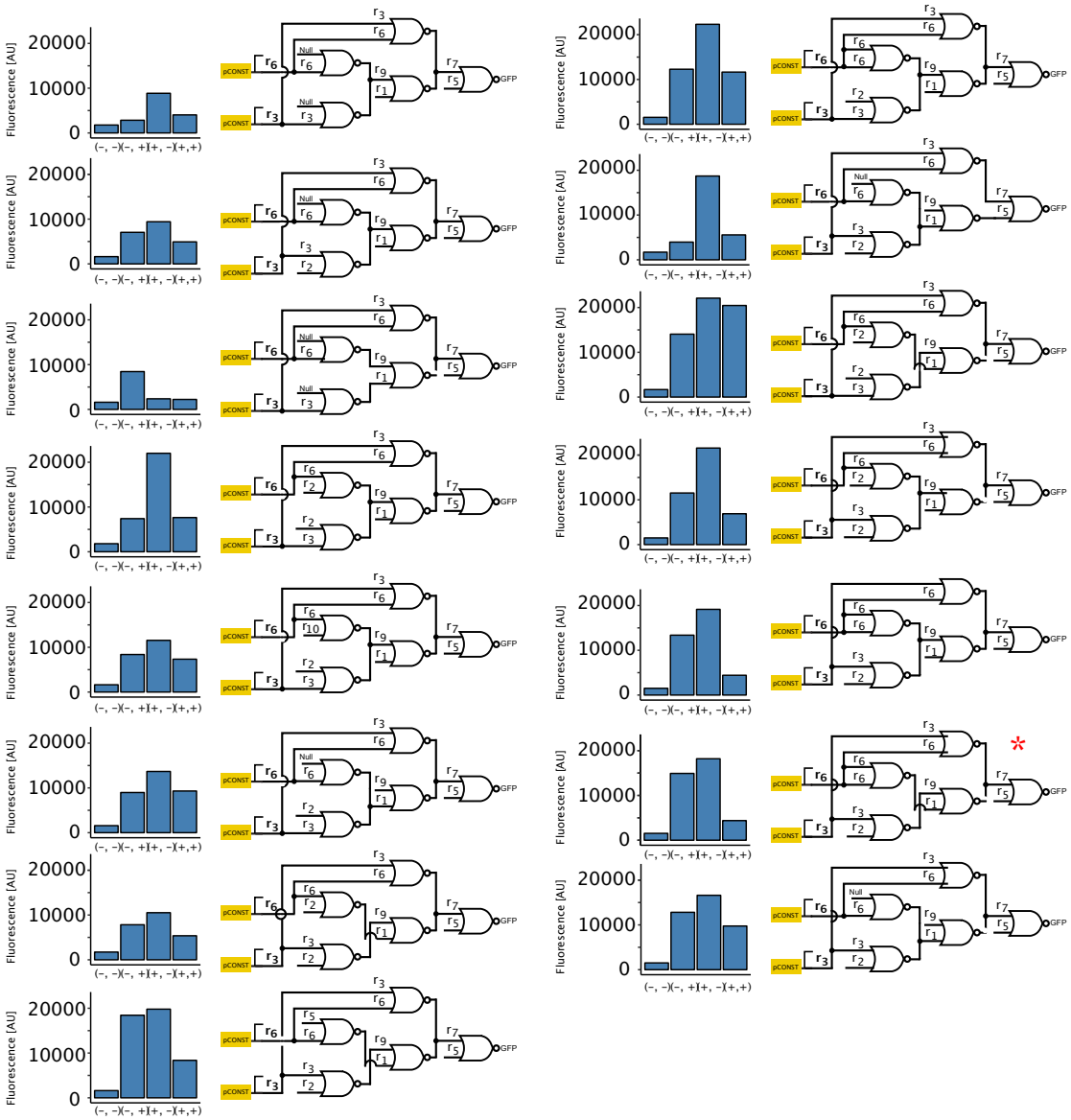
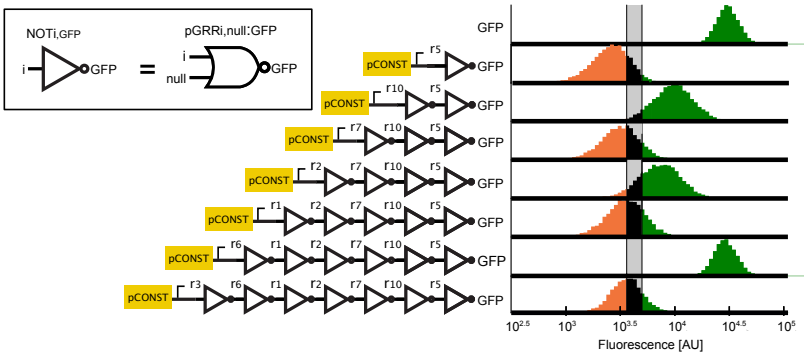


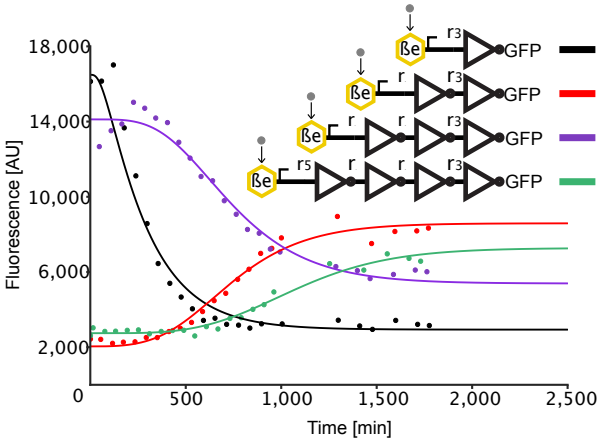
Figure 2.6: **XOR variability** The behavior of 15 XOR circuits implemented in *S. cerevisiae*. CRISPRi NOR gates are displayed as curved boxes using standardized boolean logic notation and intended gRNA:promoter interactions are displayed as wires connecting the NOR gates. Interactions are labeled as "r1" through "r20" representing their gRNA target sequence identity. Each circuit varies slightly by which gRNAs were chosen for each interaction, while overall circuit topology remained the same. As in Fig 2.4, steady-state cytometry measurements were taken for each input combination. Mean fluorescence is plotted for each input combination on the left. The variability in behavior is evident across the 15 circuits. The circuit labeled with a red asterisk was used in Fig 2.4.

First, we verified that the inducible cascades were performing properly. If properly assembled, we expected the cascades to exhibit an alternating output of GFP fluorescence with the addition of a NOT gate. As expected, steady-state measurements of the strains showed that the one and three-layer cascades exhibit high GFP fluorescence while the two- and four-layer cascades would exhibit low fluorescence in the absence of β-estradiol. Next, we tested the hypothesis that cascades of increasing length should exhibit increasingly long response times and would show a 'delay' in signaling. To measure the kinetics of the system, the inducible signaling cascades were grown to mid-log phase and input gRNA inducible with 100 μM of β-estradiol. GFP expression was measured every 30-60 minutes for 28 hours. Strains were manually diluted to ensure log-phage growth. As expected, longer cascades exhibited longer delays in response time upon β-estradiol, with the cascades reaching half-maximal expression at 4.08$\pm$0.45, 10.78$\pm$1.04, 12.01$\pm$1.18, and 17.83$\pm$1.00 hours (ressd) for cascades of depth one through four respectively (Fig 2.7).

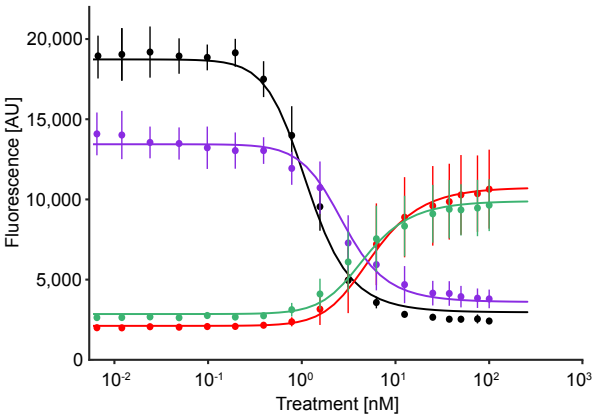
After confirming that the cascades were performing properly, we measured the steady-state characteristics of the repression cascades. The response function of these cascades would indicate the expected performance of our $\text{NOR}_{i,j,k}$ in much larger circuit contexts and would allow us to predict the limitations of our circuit architecture. To measure the response function of our signaling cascades, all four strains were grown to exponential phage and were induced with a serial dilution of β-estradiol (12 different dilutions from 0 to 100μM), constituting a total of 72 cultures. The experiment was repeated 3 times to estimate experimental variability. In the absence of a turbidostat capable of maintaining many yeast cultures, it was infeasible to maintain a culture in exponential phase, and so cultures were diluted (to OD 0.1-0.3) every 8-15 hours. The cascades reached steady-state after 5 days of culturing and were measured on the cytometer. It is interesting to note that the kinetics of the system appeared to be significantly faster when maintained in log-phase, indicating that the system is cell-state dependent. The cascades exhibited a clean response function, with what appeared to be a decreasing response in the dynamic range (Fig 2.8).



(a) steady-state measurements

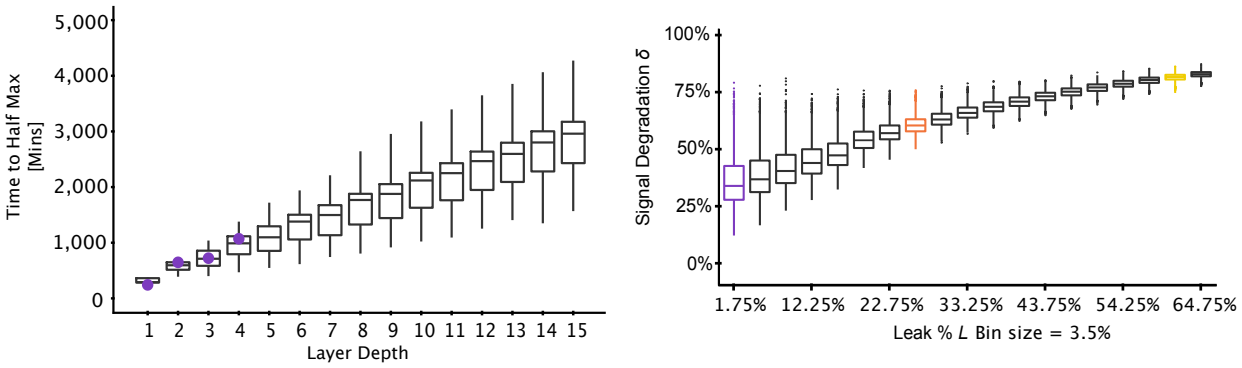


(b) kinetics



(c) dose-response (βe)

Figure 2.7: **Repression cascade characterization** Several CRISPRi dCas9-Mxi1 NOR gates were connected in series to create repression cascades composed of one to seven interactions. The final output of each cascade is a NOT gate that expresses GFP. The Cascades with an even number of layers express a high level of GFP, creating a digital ON output, and odd depth cascades express low levels of GFP, creating a digital OFF output. Fluorescence measurements were taken using flow cytometry. The histograms represent population fractions from three different biological replicates measured during a single experiment and were normalized so that area sums to unity. (b) Temporal dynamics for cascades of one to four gRNAs. Expression of the input gRNA was induced with β-estradiol. A model of the cascade, in which each layer is treated as a Hill function, was used to fit the data. The plot shows the data from one biological replicate. As the number of layers in the cascade increases, increases in signal degradation and time to steady-state is observed. (c) The steady-state response function for the four inducible cascades. Error bars represent the s.d. of three biological replicates measured over three separate experiments. (d) A representation of the model. The model was used to generate the fits for the steady-state and kinetic inducible cascade experiments.



(a) Signal delay

(b) Signal degradation

Figure 2.8: **Signal degradation analysis** (a) Simulations of time to half-maximal response using the model. Increasingly layered cascades show a positive linear relationship between circuit time to half-maximal response and circuit depth, with a slope of 184.9±0.2 (s.e.m.)min layer-1. The first four data points highlighted in purple are experimental data from Fig. 2.7. (b) Signal degradation, δ, in a cascade increases as the transcriptional leak of the gates increases. Boxplots of δ values were plotted with binned values of the leak parameter L. At values of L<1.75%, the spread of performance of the cascades is significantly larger. The bin containing the steady-state experimentally predicted value of dCas9-Mxi1, L=0.6±0.1% (s.d.), is highlighted in purple. The bins highlighted in orange and yellow contain the predicted L values for the steric repression measurements in 2.5of position 1, L=25.0%, and position 3, L=61.3%, respectively.

It appeared that the cascades fit a standard hill-equation, suggesting that the system follows the same rules as ligand-binding chemistry. So next we proposed to fit a model representing a plausible and simplified mechanism for the repression of the pGRR transcription by dCas9-Mxi1 repression; if the model fits the data, we can say that the data is not inconsistent with our proposed mechanism. We first started with measuring the input β-estradiol to gRNA transfer function by measuring pGALZ4 promoter driving GFP (data not shown); the responses fit a standard hill-equation and parameters for the transfer function were estimated. Next, we modeled the response of gRNA to a cognate promoter. Our proposed model is that of nested hill-like equations (see Methods 2.4).

In order to make inducible cascades were built in such a way that they charged parameters between the cascades; for example, the output $\mathrm{NOT}_{i,\mathrm{GFP}}$ is identical for all repression

cascades while the $\text{NOT}_{i-1,i}$ gate is identical for the 2, 3, and 4 layer cascades, and so on. This parameter sharing reduces the total number of parameters in the system. However, the first attempts to fit the data failed as the parameter landscape has many local minima, making the model inappropriate for analytical or gradient descent fitting algorithms. Instead, Differential Evolution (DE) [57], a global optimization algorithm was used to fit the data. As DE uses a meta-heuristic to estimate parameters in a stochastic way, it was necessary to repeat the fitting a number of times to ensure parameters represented a globally optimized solution. The DE fitting procedure was repeated 15 times for each steady-state experiment to estimate parameters for 72 hours of computation time, resulting in three sets of parameter fits, one for each experiment. Parameters fits for each experiment were deemed to be consistent with a normal distribution, and so the parameter means for each experiment were taken as the estimated parameter estimate. Parameter estimates for each experiment were used to find the average and standard deviation of each parameter, resulting in our best estimated parameter set derived from three separate experiments. Parameters were rescaled to concentrations of gRNA estimated from the published dissociation constants of Cas9 to a cognate site of 1.2 nM [58]. The promoters in the cascade were estimated to be capable of producing 7.06 +/- 1.47 nM; this variation in promoter strength (coefficient of variation of 21%) was found to be consistent with the variation found with other NOR gates (coefficient of variation of 18%). It was found that about 2.3 +/- 1.03 nM (cv 44%) of gRNA was required to suppress a pGRR promoter to half-strength; we note that the repression strengths appear to be highly variable. We also note that by using the estimated nuclear volume of yeast, we can easily estimate the number of gRNAs in the nucleus; using a previously published nuclear volume of 2.9 μm [59], it is estimated that there are around 12.3 gRNA molecules during maximum transcription of a pGRR promoter, suggesting that gRNA mediated signal transduction in our system is a highly stochastic process. Parameter fits are displayed in Table 3.1

The kinetics data were estimated using a similar method with the exception being that there is only one experiment. The kinetics data were fit separately as the experimental

conditions between the kinetics and steady-state experiments were different to warrant a separate fit. However, interestingly, the parameters estimated from the kinetics data fall within the error found in the steady-state parameters. Also interesting is that the half-life of the gRNA was estimated to be around 127 minutes, or approximately the time it takes for a yeast cell to divide. Hence this suggests that gRNA:Cas9-Mxi1 are fairly stable within the nucleus and the main contribution to its degradation is equal partitioning of the molecules within the nucleus during cell division. Hence, the daughter cells likely inherit a significant portion of gRNA:Cas9-Mxi1 upon division, explaining the long response times in longer cascades; the modeling this partitioning of gRNA during cell division with the low number of gRNA estimated from the previous fit is consistent with the kinetics observed (data not shown).

The hill-coefficient represent the cooperativity of the gRNA of the promoter, with high cooperativities (> 2) making the response function increasingly more 'digital' (like a step-function) and low cooperativities (< 1) making the response more linear and 'analog.' The NOT gates show slightly higher cooperativity of 1.72, indicating that it somewhere between digital and analog. Given this information, we wanted to see if we could estimate the performance of larger synthetic circuits. Using the parameter variances determined from the steady-state fit, we created models of long cascades by Monte Carlo resampling the known parameter values. We used this data to estimate the project time-to-half maximum of 10,000 virtual cascades. There is a linear relationship between the length of the cascade and the time-to-half max (197.67$\pm$0.45 (s.e.m.), suggesting the long NOR gate circuits are more appropriate for complex slow responding behaviors, such as in development. Next, we estimated the average 'signal degradation' of long cascades. We hypothesized that each additional layer of a NOR gate will, on average, result in a decrease in the dynamic range response and that this decrease in signal degradation would be largely dependent on the hill-coefficient (how more or less digital the response functions are). We generated 100,000 virtual cascades for different values of n and plotted the signal degradation, δ, calculated as the average signal degradation loss per additional layer. We found that coefficients between analog and digital

(between 1 and 3) have highly variable δ, suggesting that the variation is mainly attributed to variations in the parameters such as transcription strength and repression strength.

2.4 Materials and Methods

2.4.1 Plasmid construction and assembly

Backbone and insert fragments were amplified with PCR, gel extracted, purified, and assembled using Gibson assembly [60] using standardized assembly linkers. Backbones contained a high-copy E. coli origin of replication and ampicillin resistance for propagation. The yeast expression cassettes were flanked upstream and downstream by approximately 500 bases of chromosomal homology to the yeast genome and PmeI restriction sites for linearization before transformations. Plasmids were sequence-verified using Sanger sequencing.

2.4.2 Yeast strain construction

W303 strains were generously provided by the Dunham lab and were used as initial parent strains. Strains were constructed using genomic integration from linearized DNA. Integrative plasmids were linearized using PmeI digestion (37C, 30 min) to cut upstream and downstream of the chromosomal homology. Unpurified, linearized DNA was transformed into yeast cells using a standard lithium acetate protocol [61]. Strain selection was performed on solid synthetic-complete (SC) using auxotrophic or antibiotic selection. Diagnostic colony PCR was performed to verify integration into the proper locus. Strains were picked from single colonies and stored long term at -80°C in a sterilized 30% glycerol and media solution. Strain retrieval was performed by plating glycerol stocks onto solid media plates (YPAD) grown for 2-3 days at 30°C and picking single colonies for liquid culture. All yeast cultures and assays were grown at 30°C shaking at 275 RPM.

2.4.3 Circuit strain construction

S. cerevisiae haplotypes can be turned into a diploid genotype by co-culturing MATA and MATα haplotypes in liquid culture. This allows parallel construction of yeast haplotypes and using yeast mating to combining genetic changes into a single diploid strain. This strategy was used in this study to combine haploid strains containing smaller subsets of genetic parts needed for a circuit. All strains used in this study were eventually made diploid by a given strain with its mating partner, MATA or MATα) and selected via auxotrophic or antibiotic selection. To mate two haploid strains, MATA and MATα haplotypes were combined in liquid culture at 30°C shaking at 275 RPM for 4-8 hours, followed by selective plating and colony picking.

2.4.4 Flow cytometry assays

Fluorescence intensity was measured with a BD Accuri C6 flow cytometer equipped with a CSampler plate adapter using an excitation wavelength of 488 and an emission detection filter at 533nm. For each sample, at least 10,000 events above a 400,000 FSC-H threshold (to exclude debris) were recorded for each sample with and a core size of 22mm using the Accuri C6 CFlow Sampler software. Cytometry data were exported as FCS 3.0 files and processed using the FlowCore R software package and custom R scripts to obtain the mean FL1-A value at each data point.

2.4.5 Stead-state and dose-response measurements

For steady-state measurements, cytometry measurements were taken on cells grown in cultures diluted 1:1,000 from saturated culture for 16h at 30°C. For inducible and dose-response measurements, cells from saturated culture were diluted 1:100 into fresh media with a beta-estradiol (β) concentration of 100nm. Cytometry measurements were taken over a 30h period. During the time course, cells were periodically diluted to keep them in log growth phase. Experimental data collected for steady-state were measured for four strains, each

containing four different βe-inducible cascades. Each of the four strains was induced with 18 different doses of βe ranging from 0 to 100μM in a single batch of 72 cultures. Cells were diluted every 8–15h to prevent culture saturation. Steady-state fluorescence readings were taken after 5 days when the cultures were in log phase.

2.4.6 Modeling

A deterministic model of our system was described by three ordinary differential equations characterizing transcription, degradation and repression. The gRNA-dCas9-Mxi1 and green fluorescent protein (GFP) molecular constituents were modelled as follows:

$$\begin{aligned} \frac{dr_d}{dt} &= b\Big(\frac{v_d(1-L)}{1+(\frac{r_{d+1}}{k_d})^n} + L_{v_d} - r_d\Big) \\ \frac{dr_D}{dt} &= b\Big(\frac{V(\frac{u}{K})^{n_u}}{1+(\frac{u}{K})^{n_u}} - r_D\Big) \\ d &\in 1, ..., D-1 \\ \frac{dG}{dt} &= B\Big(\frac{1-L}{1+r_1^n} + L - G\Big) \end{aligned} \tag{2.1}$$

r_d is the concentration of the dth gRNA-dCas9-Mxi1, d ranges from 1 to $D-1$, where D is the number of layers in the cascade; r_D is the input gRNA driven by the inducible promoter; v_d is the promoter strength driving each r_d in terms of the maximum steady-state concentration of gRNA from the promoter; G is the measurable normalized concentration of GFP; b is the degradation/dilution rate of all r_d; B is the degradation/dilution for GFP; k_d is the repression strength of r_d to its cognate promoter, in terms of the number of repressors required to suppress a promoter to half strength; to its cognate promoter is modeled with k_d, the number of repressors required to suppress a promoter to half-strength; and n is a Hill coefficient. For the transfer function, V, K, n_u respectively represent the maximum transcription, Michaelis–Menten constant, and Hill coefficient of the inducible promoter; u is the input βe in μM. Concentration is rescaled as the Michaelis–Menten constant, or the number of gRNAs required to suppress a NOT gate to half-maximal. Note that the model makes the assumptions that (i) there is no crosstalk between gRNA components, (ii) Mxi1

represses transcription completely with no transcriptional leak, and (iii) dCas9-Mxi1 bind quickly and irreversibly to gRNA. A full description of the chemical reaction network (CRN) used for the repression cascades and parameter sensitivity analysis can be found in section 2.6.1.

2.4.7 Model Fitting

Parameters were optimized using differential evolution followed by minimization using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm[57]. For the steady-state experiments, optimal parameter fits for the parameters $v_0^{\text{ss}}, \ldots, v_3^{\text{ss}}$, $k_0^{\text{ss}}, \ldots, k_3^{\text{ss}}$, V^{ss}, n^{ss} were generated from three separate experiments. For each of the three experiments, 17 parameter fits were generated using differential evolution/BFGS. Parameter means were calculated for a total of 51 steady-state parameter sets. The means from each experiment were used to determine the experimental error (σ) for estimating each parameter. For the kinetics experiments, five parameter fits for $v_0^{\text{kinetics}}, \ldots, v_3^{\text{kinetics}}$, $k_0^{\text{kinetics}}, \ldots, k_3^{\text{kinetics}}$, V^{kinetics}, n^{kinetics}, b, B were generated from a single experiment. As there were only data for a single kinetics experiment, experimental errors for the kinetic parameter values were not calculated. Parameters K and nu were determined in a separate experiment by driving a YFP with the pGALZ4 β-estradiol inducible; this promoter is the same promoter used in the inducible cascades. The kinetics and steady-state parameter sets were resampled in downstream analyses to generate Monte Carlo simulations of longer repression cascades. Parameter values inducible pGALZ4 promoter is displayed in Table 2.1. Parameters fitted to steady-state data are in Table 2.2. Parameters fitted to kinetics data are in Table 2.3. For comparison, literature values for transcription, degradation, and dCas9:DNA dissociation are in Table 2.4.

2.4.8 Model predictions

Long repression cascades of 1 to 11 ($D \in \{1 \ldots 11\}$) layers were simulated using the system of ordinary differential equations. Parameters for simulated cascades were generated by resampling parameter sets generated during the fitting procedure. For the kinetic model

predictions, 10,000 simulated cascades were generated by sampling parameters from five parameter sets estimated from the kinetics experiment. The time-to-half max of GFP (G) was calculated for each cascade length D and plotted. For the signal degradation (δ) predictions, 100,000 simulated cascades of length D=7 were simulated by sampling parameters from the 51 parameter sets estimated from the three steady-state experiments. To compare L versus δ, L was sampled from a uniform distribution between 0 and 1. Signal degradation (δ) was calculated as the percent change in dynamic range per additional layer. The dynamic range at each layer d in a cascade of length D was calculated using Eq. 2.2:

$$\rho_d = \log = \left(\frac{\max(r_d)}{\min(r_d)}\right) \tag{2.2}$$

Dynamic range was found to have a log-linear relationship with the length of the cascade, and hence the average slope between d versus $\log(\rho_d)$ was calculated using linear regression for each of the 100,000 simulations of cascades of length D by Eq. 2.3

$$\rho = \frac{D\sum_d^D (d\log(\rho_d)) - \sum_d^D d\sum_d^D \log(\rho_d)}{D\sum_d^D d^2 - \sum_d^D (d)^2} \tag{2.3}$$

with $D = 7$. With η being the change in $\log(\rho_d)$ with each additional layer, the percent loss in dynamic range per layer or signal degradation δ is calculated as:

$$\delta = 1 - 10^n \tag{2.4}$$

To plot, values for L were binned using a bin size of 0.035 and δ versus L was plotted.

2.5 Conclusion

We have created a modular and scalable framework for engineering cell behaviors in *S. cerevisiae*. Utilizing the programmability of CRISPR-Cas9 and a library of synthetic guide RNA and promoter parts, we have demonstrated that we can re-compose individual characterized parts into more complex functions. We implemented several canonical examples of two-input boolean logic in *S. cerevisiae* and several examples of longer repression cascades. Use utilized

simple ODE modeling to characterize the general properties of our parts. During the process of designing and constructing these circuits, we discovered unexpected variability in more complex circuits (e.g. XOR gate) that was not immediately explainable from the characterization of individual parts. This was surprising since the genetic parts used to construct the circuits exhibited low off-target effects and performed nearly identically when characterized in isolation. This forced us into a costly procedure of evaluating many different circuit implementations to find satisfactory behaviors. The modeling we performed suggests that the 'steepness' of the response function in conjunction with slight variances in expression is likely resulting in signal loss in more complex circuits. However, it is important to note that there may be other confounding variables and interactions that were not accounted for in the ODE modeling. These may include load-resource competition [62], RNA-folding effects [63], or other unknown interactions with host-cell machinery.

2.6 Appendix

2.6.1 Chemical Reaction Network (CRN) for CRISPR repression cascades

The repression cascade is constructed from an inducible promoter driving expression of a guide RNA (gRNA). This induction is described by the following chemical reaction network (CRN):

$$P_0 + u \underset{ku_2}{\overset{ku_1}{\rightleftharpoons}} P_0 : u$$

$$P_0 : u \xrightarrow{v} P_0 : u + gRNA_0$$

$$u \xrightarrow{\beta_u} \emptyset$$

$$gRNA_0 \xrightarrow{\beta} \emptyset$$

The expression of a Cas9 is describe be the following reactions:

$$P_{cas} + r_{i-1} \underset{k_-}{\overset{k_+}{\rightleftharpoons}} P_i : r_{i-1}$$

$$P_{cas9} \xrightarrow{v} P_{cas9} + m_{cas9}$$

$$m_{cas9} \xrightarrow{v} m_{cas9} + cas9$$

$$m_{cas9} \xrightarrow{\beta_m} \emptyset$$

$$cas9 \xrightarrow{\beta_{cas9}} \emptyset$$

For a repression cascade consisting of m repressible promoters, the binding of each ith gRNA to a Cas9 is described the the following reactions:

$$cas9 + gRNA_i \underset{ku_2}{\overset{ku_1}{\rightleftharpoons}} r_i$$

$$\text{i}\in[1,\text{m}]$$

Each ith gRNA can repress the (i+1)th promoter, making a complex. These reactions can be generally described as:

$$P_i + r_{i-1} \underset{k_-}{\overset{k_+}{\rightleftharpoons}} P_i : r_{i-1}$$

$$P_i \xrightarrow{v} P_{i-1} + gRNA_i$$

$$r_i \xrightarrow{\beta} \emptyset$$

i∈[1,m]

The binding between a gRNA and Cas9 is generally thought to be very fast and the dissociation constant very low. We can assume every gRNA forms a complex with a Cas9, assuming the levels of Cas9 are much greater than the levels of any of the gRNAs. The system of equations above can be reduced to the following equations:

$$P_0 + u \underset{ku_2}{\overset{ku_1}{\rightleftharpoons}} P_0 : u$$

$$P_0 : u \xrightarrow{v} P_0 : u + r0$$

$$r0 \xrightarrow{\beta} \emptyset$$

$$P_i + r_{i-1} \underset{k_-}{\overset{k_+}{\rightleftharpoons}} P_i : r_{i-1}$$

$$P_i \xrightarrow{v} P_{i-1} + ri$$

$$r_i \xrightarrow{\beta} \emptyset$$

Dynamic of CRISPR repression cascades A repression cascade consisting of m repressible layers consists of $4(m+1)$ equations. The dynamics of the system can be described as $v = A_0 K_0(v)$ where $K(v)$ is a reaction matrix of length $4(m+1)$ and A is a $4(m+1)\times 4+3m$

matrix:

$$K_0(v) = \begin{pmatrix} vP_0^u \\ ku_1P_0u \\ ku_2P_0^u \\ \beta r_0 \\ k_+P_1r_0^n \\ k_-P_1^{r_0} \\ \alpha P_1 \\ \beta r_1 \\ ... \\ k_+P_mr_{m-1}^n \\ k_-P_m^{r_{m-1}} \\ \alpha P_m \\ betar_m \end{pmatrix}, A_0 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.5}$$

The dynamics first inducible gRNA expression is described above by the equations as the binding of activator u to the promoter: $\frac{dr_0}{dt} = P_0ku_1 - \beta r_0$ Given that the concentration of all promoters remains constant, i.e. $P_{0Total} = P_0 + P_0 : u$, and assuming the binding of the activator is much faster than the kinetics of expressing gRNA, i.e. $P_0 : u = \frac{ku_1uP_0}{ku_2}$, we use the conservation of mass is used to simplify the equation to:

$$\frac{dr_0}{dt} = \frac{P_{0Total}ku_1u^2}{1 + \frac{ku_1u^2}{ku_2}} - \beta r_0 \tag{2.6}$$

For which we can group the constants into simpler parameters:

$$\frac{dr_0}{dt} = \frac{K_u(v)(u^2)}{1 + K_u(u^2)} - \beta r_0 \tag{2.7}$$

Similarly for the repression for the last gRNA can be described as $\frac{dr_i}{dt} = P_i\alpha - \beta r_i - k_+P_{i+1}(r_i) + k_-P_{i+1} : r_i$ and the ith gRNA for for i∈[1,m] can be described as $\frac{dr_m}{dt} = P_m\alpha - \beta r_m$. However, we assume that binding between r_i and P_{i+1} and so these terms

cancel leaving the general equation for all i∈[1,m]:

$$\frac{dr_i}{dt} = P_i\alpha - \beta r_i \tag{2.8}$$

Using the conservation of mass and assuming $P_i : r_{i-1} = \frac{k_+ P_i}{ku_-}$, we simplify the equations down and group together constants to produce the following hill functions:

$$\frac{dr_i}{dt} = \frac{P_{iTotal}\frac{k_+}{k_-}}{1 + \frac{ku_1}{ku_2} r_{i-1}^n} - \beta r_i \tag{2.9}$$

$$\begin{gathered} \frac{dr_i}{dt} = \frac{\alpha}{1 + K(r_{i-1}^n)} - \beta r_i \\ i \in [1, m] \end{gathered} \tag{2.10}$$

System stability analysis for CRISPR repression cascades From the above equations, for any given repression cascade, the expression of the i-th gRNA depends only on the expression of the (i-1)th gRNA. Therefore the output and intermediate levels in the cascade depend only on the preceding levels of gRNA, and is then, in the absence of feedback, always expected to have one single stable equilibrium point. The Jacobian matrix for an m-layer repression cascade described above is of the following form: Jacobian:

$$A = \begin{pmatrix} -\beta & 0 & 0 & 0 & ... \\ \frac{\alpha k n r_0^{n-1}}{\left(k r_0^n + k r_3^n + 1\right)^2} & -\beta & 0 & 0 & ... \\ 0 & \frac{\alpha k n r_1^{n-1}}{\left(k r_1^n + 1\right)^2} & -\beta & 0 & ... \\ 0 & 0 & \frac{\alpha k n r_2^{n-1}}{\left(k r_2^n + 1\right)^2} & -\beta & ... \\ ... & ... & ... & ... & -\beta \end{pmatrix} \tag{2.11}$$

It follows that the eigenvalues of the Jacobian will be an m-length matrix

$$\lambda = \begin{pmatrix} -\beta \\ ... \\ -\beta \end{pmatrix}^{\mathsf{T}} \tag{2.12}$$

Since the β is a dilution term and must always be positive, all of the eigenvalues are always strictly negative. It follows from Lyapunov's Indirect Method that the system will always have a locally stable equilibrium. Further, since the expression of each gRNA depends only on the expression of the previous gRNA, it also follows that there will be exactly one equilibrium point for a given cascade.

Parameter sensitivity analysis for repression cascades Partial rank correlation coefficient (PRCC) was used to determine the sensitivity of the dynamic range of parameters for repression cascades (Fig: 2.9) for 6-layer repression cascades.

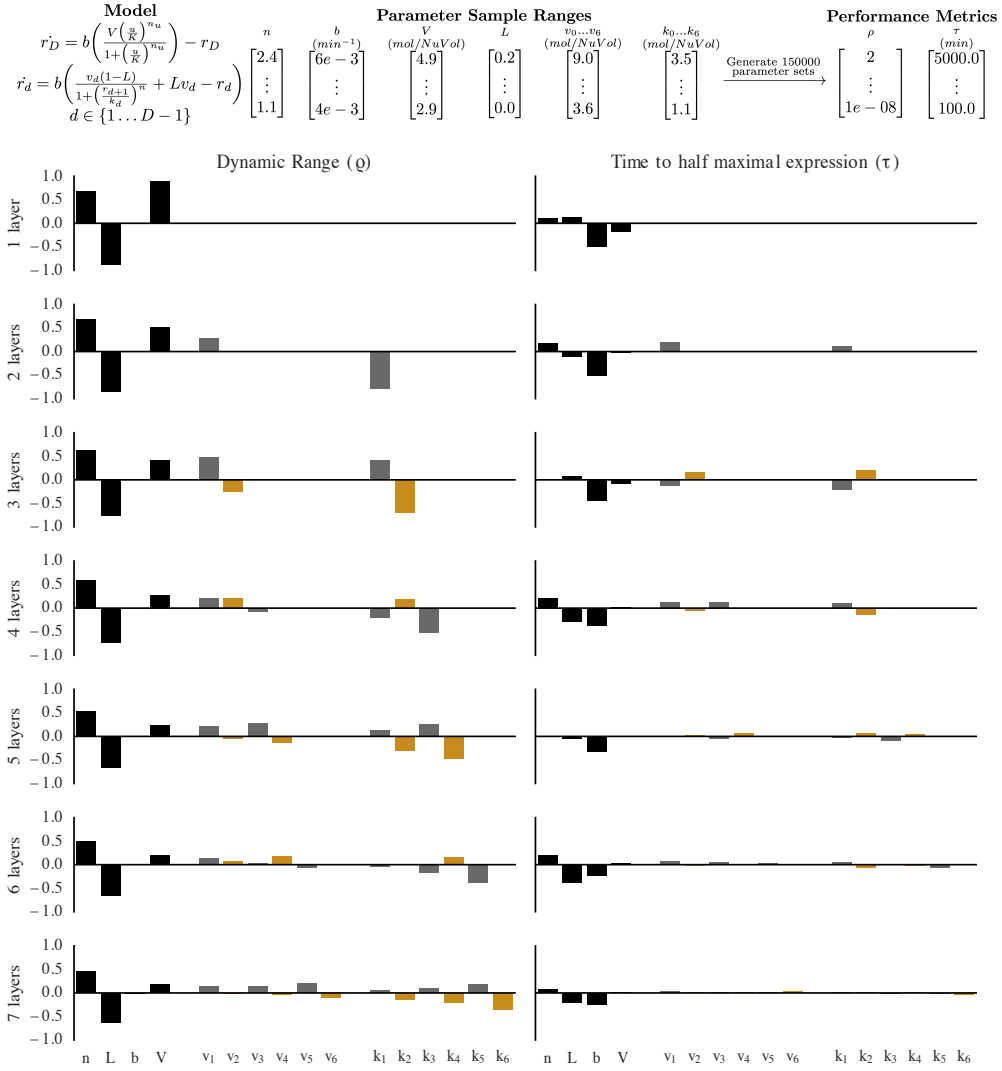


Figure 2.9: **Model parameter sensitivity** 150,000 parameter sets were re-sampled from a uniform distribution over the intervals shown and applied to our repression cascade model (see methods). (b) Partial rank correlation coefficient (PRCC) [64] was used to determine the contribution of each parameter has on either dynamic range or the time-to-half max. PRCCs were calculated using R (R Foundation for Statistical Computing, Vienna, Austria). Parameters associated with odd and even layers are colored grey and orange respectively. At all layers in the time-to-half maximal plot, b is very correlated with the output. In the dynamic range plot, n and L is strongly positively correlated at all layers with the output.

2.6.2 Details on model fitting

Below are tables describing parameter estimates for model from a differential evolution fitting algorithm from the steady-state and kinetics experiments (Figs. 2.7, 2.8). Using an estimated nuclear volume of 6 μm^3 for diploid yeast [65] and the published dissociation constant of 1.2nM for Cas9 binding to its cognate site [66], bounds for parameters during optimization were selected based on estimates of transcription rates [67] and protein degradation rates in yeast [68]. Standard deviations of the steady-state parameters were determined from three independent experiments (n=3). Kinetic parameters were determined from a single experiment and do not have an estimate for experimental error (n=1). Values labeled with "*" means the standard deviation not computed because there was only one experiment. Values labeled with "**" means this parameter was preset to equal unity, as an artifact of the fitting procedure. For transparency, fitting bounds used in the algorithm are shown, which were determined by trial-and-error. Concentrations of molecular species are in units of molecules per nuclear volume labeled as $\frac{Molecule}{NuVol}$.

Table 2.1: Parameter sensitivities for CRISPR dCas9 ODE model

Parameter	**Mean (std)**	**Units**	**Fitting Range**	**Description**
V_{ss}	16.854 (1.073)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.4, 130)	Maximum transcription from inducible promoter
K	2.880 (0)	nM	(2.880, 2.880)	Michaelis-Menten constant for βe inducible promoter
n_u	1.239 (0)	dimensionless	(1.239, 1.239)	hill-coefficient for inducible promoter

Table 2.2: Parameters for CRISPR dCas9 system (measured at steady-state)

Parameter	**Mean (std)**	**Units**	**Fitting Range**	**Description**
v_0^{ss}	1.000 (0)	AU	NA	Max fluorescence of reporter, normalized to 1.0
v_1^{ss}	31.114 (2.436)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.434, 43.359)	Max transcription from pGRR promoter
v_2^{ss}	20.876 (2.469)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.434, 43.359)	⋮
v_3^{ss}	21.183 (4.107)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.434, 43.359)	⋮
k_0^{ss}	1.000 (0)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	dissociation constant of gRNA-dCas9-Mxi1 to its cognate promoter
k_1^{ss}	6.129 (0.992)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	⋮
k_2^{ss}	12.229 (4.065)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	⋮
k_3^{ss}	11.782 (3.442)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	⋮
n^{ss}	1.882 (0.107)	dimensionless	(0.5, 4)	hill-coefficient

Table 2.3: Parameters for CRISPR dCas9 system (measured in kinetics experiment)

Parameter	**Mean (std)**	**Units**	**Fitting Range**	**Description**
v_0^{kinetics}	1.000 (0)	AU	NA	Max fluorescence of reporter, normalized to 1.0
v_1^{kinetics}	23.631 (0.266)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.434, 43.359)	Max transcription from pGRR promoter
v_2^{kinetics}	19.367 (0.172)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.434, 43.359)	⋮
v_3^{kinetics}	19.054 (0.173)	$\frac{\text{molecule}}{\text{NucVol}}$	(0.434, 43.359)	⋮
k_0^{kinetics}	1.000 (0)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	dissociation constant of gRNA-dCas9-Mxi1 to its cognate promoter
k_1^{kinetics}	6.771 (0.072)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	⋮
k_2^{kinetics}	12.988 (0.061)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	⋮
k_3^{kinetics}	14.411 (0.121)	$\frac{\text{molecule}}{\text{NucVol}}$	NA	⋮
n^{kinetics}	2.231 (0.006)	dimensionless	(0.5, 4)	hill-coefficient
B	0.005 (0.000)	min^{-1}	(0.003, 0.011)	degradation/dilution of GFP reporter
b	0.006 (0.000)	min^{-1}	(0.003, 0.011)	degradation/dilution of gRNA:dCas9-Mxi1 complexes

Table 2.4: Literature values for transcription/degradation in yeast and dissociation of Cas9:gRNA from DNA in *S. cerevisae*

Parameter	**Range**	**Units**	**Description**
Reported txn rate	0.03 - 0.5	min^{-1}	transcription rate of most *S. cerevisae* promoters [67]
Reported protein deg. rate	1.9×10^{-4} - 5.8×10^{-2}	min^{-1}	degradation rate of most *S. cerevisae* promoters
K_d of Cas9	1.2	nM	dissociation constant of gRNA:Cas9 to target measured *in vitro* [68]

in

Table 2.5: List of gRNA sequences used in the study

gRNA	**index Sequence**
r1	GGAACGTGATTGAATAACTT
r2	ACCAACGCAAAAAGATTTAG
r3	CATTGCCATACACCTTGAGG
r4	GAAAATCACAACTCTACTGA
r5	GAAGTCAGTTGACAGAGTCG
r6	GTGGTAACTTGCTCCATGTC
r7	CTTTACGTATAGGTTTAGAG
r8	CGCATTTCCTATTCAAACTT
r9	GCAACCCACAAATATCCAGT
r10	GTGACATAAACATTCGACTC
r11	GGGCAAAGAGACGCTTGTCG
r12	GAAGTCATCGCTTCTTGTCG
r13	GAGTTGACAAAGTATAACTT
r14	GAAGTTTCAGAATCTCGACG
r15	GGCTAGGATCCATCTGACTT
r16	GCAACCATAGACTCTCCAGG
r17	ACCACAACTGAGTCGAACCT
r18	GGGTAGCAACACTCGTACTT
r19	GTAAAAGATAACTCTGTTGC
r20	TCTACCCGAGACTCAAACGG
v1	GTACATACAGTAGGATCCTA
v2	TTTGGCACTACCGACACGAA
v3	TGGTCAAAAGTGCGGCTTTC
v4	CTTTCACAATCTTGACCTGC
c3	GTACATACAGTAGGATCCTA
c6	TGCAAAGGTCCTAATGTATA

Chapter 3

EXTENSIONS AND LIMITATIONS TO THE CRISPR-DCAS9 SYSTEM

In the previous chapter, I described a versatile platform for creating genetic circuitry in *S. cerevisiae* based on the CRISPR dCas9 system. Fusing a powerful chromatin remodeling protein, Mxi1 to a deactivate CRISPR Cas9 creates a RNA-guided powerful transcriptional repressor. The strength of the repression allows for digital-like signal propagation in transcriptional circuits. I demonstrated this constructing and characterizing several canonical boolean circuits (NOR, OR, AND, NAND, XOR, XNOR) constructed out of module CRISPR dCas-Mxi1 based NOR gates. The inputs to these circuits, however, was the presence or absence of constituitively expressed gRNAs. Using using ordinary differential equations and modeling of constructed inducible repression cascades, I showed in the previous chapter that the individual responses for CRISPR-based NOR gates are far from step-like or digital. It is by virtue of the extraordinarily effective repression that we were able to construct genetic circuits with digital-like outputs. The fact that the signaling molecules used in the circuits were RNA-based, rather than protein-based, likely made it possible to completely repress and un-repress gRNA production to produce digital-like responses.

In this chapter, I explore the limitations to the CRISPR-dCas9 system. First, in I tell a story of an attempt to increase the steepness of the response functions for CRISPR-based NOR gates by attempting to use molecular titration to produce an ultrasensitive response. Instead of producing the desired response however, I show that the CRISPR dCas9 system is highly sensitive to the presence of alternative gRNA binding sites within the cell. Importantly, the presence of just seven additional gRNA binding sites elsewhere in the cell is enough to drastically reduce the steepness of NOR gate's response. This has

important ramifications on the ability to create large gene circuitry using the system. The XOR and XNOR boolean circuits we presented, for example, have multiple gRNA binding sites, specifically, a 'forking' of one of the gRNAs (r7). The XOR and XNOR circuits in particular exhibited the over lowest performance across all versions of these we constructed. Beyond their larger size, the titration effects of for 'forking' and gRNA may be part an explanation of why these two circuits were the most challenging to make. Second I explore limitations of the CRISPR-dCas9 system to produce dynamical and bistable systems. I briefly show, using Monte Carlo simulation, that a basic flip-flop toggle switch circuit is unlikely to exhibit bistability if they are implemented using the CRISPR-dCas9 system. In a simulation I varied parameter of an ODE model based on the uncertainty of those parameters as seen in the previous chapter and found that there is a only 5% probability of constructing a bistable flip-flop circuit. This was corroborated by other students in our group, who constructed over 20 different flip-flops, none of which exhibited bistability. This result is not all unsurprising, given the low shallow response characteristic of the NOR gates. Finally, I explore alternative designs that may exhibit bistability using the CRISPR-dCas9 system. Namely, I use designs from a new tool called Dynamic Signatures Generated by Regulatory Networks (DSGRN) that explores the dynamic characteristics of circuits by searching through network topology space in a computationally efficient way. Colleagues and I filtered circuit design topology based on what would be feasible to implement in *S. cerevisiae* and chose 3-node circuit topologies with a single inducer and single output reporter that would exhibit hysteresis (a memory-effect). DSGRN produces designs a human designer typically would not think to design; incredibly, when modeled using Monte Carlo simulation and the ODE parameters for our NOR gates, the best DSGRN designs showed that 10-25% of circuits would exhibit hysteresis behavior, a marked improvement over the flip-flop design. These results are encouraging as they demonstrate that some limitations of individual circuit parts can be overcome using circuit topology alone.

Author	Contribution
Justin Vrana	all experiments, data analysis, modeling (except DSGRN circuit designs)
Bree Cummins (MSU)	DSGRN designs
Konstantin Mischaikow (Rutgers)	DSGRN designs

3.1 Titration effects of the CRISPR-dCas9 system

Given the flexibility and successes of using the CRISPR-dCas9 for transcriptional circuits, a natural extension might be to attempt to improve the sensitivity of the system. Titration appears to be a good avenue. Because the system uses gRNA molecules to propagate signals in a circuit, there is not additional translational amplification occurring after transcription. This means that the number of gRNA molecular might be low enough that I could provide enough dummy or 'decoy' binding sites to get a titration effect. In the next sections, I find I was unsuccessful in accomplishing this effect. However, I found that the response function was flattened. This result has implications for how the circuits respond when composed in very large circuits. Misbinding to alternative sites may have a diminishing effect on the ability to engineer new behaviors, especially with large complex circuits.

Minimizing noise propagation is essential for maintaining signal integrity in a circuit. Electrical engineers use digital circuits, or circuits with step-like or ultrasensitive responses, to transmit signals with minimal degradation due to noise. Ultra-sensitive responses can be found naturally in circuits as co-operatively acting transcription factors, such as the Tet-R and lambda repressors [69, 70]. It has been shown in previous studies that sequestration of transcription factors plays a role in regulating gene expression in a form of post-translational regulation [71–74], possibly producing sigmoidal-like responses in the cell. Previous studies have shown that the number of possible binding sites and relative strengths of these binding interests can result in a more sigmoidal response in gene expression [71, 75] and statistical modeling indicates that size of the genome also effects the titration effect as the transcription factors sample the genomic space for binding sites [76]. These previous studies indicate that molecular titration may provide a possible and tunable mechanism generating steep and ultrasensitive dose-responses.

By the introduction of artificial transcription factor target sites, it may be possible to artificially engineer an ultrasensitive dose-response to a transcriptional expression whose response is not typically characterized to be sigmoidal. Additional alternative guide RNA

target sites provided on a plasmid (called the **decoy sites** here), I attempt here to produce a molecular titration effect in the CRISPR dCas9-Mxi1 system previously described. For molecular titration to work, there must be a preference for dCas9-Mxi1 to bind to the decoy sites. In order to artificially create this preference, I mutated the cognate gRNA target site so that the gRNA would perfectly match the decoy sites, but imperfectly match the promoter cognate site. The idea here is that the CRISPR dCas9 repression might be strong enough that the advantage in producing a sigmoid response using molecular titration may override potential decreases in sensitivity at the promoter site.

3.1.1 Development of dCas9 testing platform

A beta-estradiol (βe) inducible was design to express a ribozyme flanked gRNA cassette (gRNA **r2**) similar constructs to previously described. A cognate CRISPR NOT gate with a single cognate gRNA site for gRNA sequence r2 was designed to express GFP (NOT_{r2}). A βe inducible promoter (pGALZ4) is used for the inducible expression of gRNA r2. Briefly, βe causes activation of cytoplasmically-localized estrogen-receptor α fused to a Zev4 transcriptional activator (ERα-Zev4). Once activated by βe, ERα-Zev4 localizes to the nucleus, where it activates transcription of gRNA r2. Expression of gRNA r2 binds to CRISPR dcas9-Mxi1 to cause transcriptional repression of the GFP reporter at the CRISPR NOT_{r2} promoter (Fig. 3.1). To distinguish the effects of the response of the inducible βe promoter (pGALZ4) and NOT_{r2} and additional control strain is included in all experiments that express a YFP from the inducible pGALZ4promoter. This allows for the calculation of relative promoter units (RPU) by plotting the response of the control strain vs the response of the circuit (Fig. 3.1).

3.1.2 Characterizing mismatches between gRNA and mutated NOT_{r2} gates

The gRNA target site at the NOT gate can be mutated to alter the dose-response function. Mutations causes a binding mismatch between the expressed gRNA sequence r2 and the cognate site, resulting in a altered response function (Fig. 3.1c). As an example, ($\mathrm{NOT}_{r2}^{\mathrm{A9T}}$).

Mutation at position 9 the 20bp cognate gRNA sequence causes a substantial decrease in the response of the NOT gate. This new promoter is designated as $\mathrm{NOT}_{r2}^{\mathrm{A9T}}$. In total four mutated NOT gate promoters were evaluated, $\mathrm{NOT}_{r2}^{\mathrm{A9T}}$, $\mathrm{NOT}_{r2}^{\mathrm{G7C}}$, $\mathrm{NOT}_{r2}^{\mathrm{A12T}}$, $\mathrm{NOT}_{r2}^{\mathrm{C8G,A12T}}$ (Fig. 3.2). I saw sequence specific variations in the effect of these mutations on the response function. In particular, the double mutation $\mathrm{NOT}_{r2}^{\mathrm{C8G,A12T}}$ resulted in an obliteration of gRNA response.

3.1.3 Modeling of mutated NOT_{r2} gates

The effect of the mutation can be modeled using a simple ODE model. The following chemical reaction network describes the system:

$$\begin{aligned} \mathrm{gRNA} + \mathrm{dCas9} &\underset{\mathrm{k_2}}{\overset{\mathrm{k_1}}{\rightleftharpoons}} \mathrm{dCas9(gRNA)} \\ \mathrm{dCas9(gRNA)} + \mathrm{P_{GFP}} &\underset{\mathrm{k_4}}{\overset{\mathrm{k_3}}{\rightleftharpoons}} \mathrm{P_{GFP}(dCas9(gRNA))} \\ \mathrm{P_{GFP}} &\xrightarrow{\alpha_{\mathrm{GFP}}} \mathrm{P_{GFP}} + \mathrm{mGFP} \end{aligned} \tag{3.1}$$

Since dCas9-Mxi1 is constitutively expressed, it is not modeled. The binding of gRNA to Cas9 is assumed to be very fast (dissociation constant $k_2 = 10^-8$)[77] and so for simplicity, gRNA:dCas9-Mxi1 heteroduplex is assumed to be equivalent to free gRNA. Hence repression is modeled by

$$m_{\mathrm{GFP}} = \frac{A_{\mathrm{GFP}}}{1 + \frac{k_3}{k_4}[\mathrm{gRNA}]^n} \tag{3.2}$$

with gRNA representing the amount of free unbound gRNA, and dissociation constant $K_D^{\mathrm{promoter}} = \frac{k_3}{k_4}$. I use an additional hill-coefficient n to capture non-linearity in the models as described previously. The fluorescence output of control strain expressing YFP off of pGALZ4promoter is used as a proxy for gRNA concentration $[gRNA]$ (Fig. 3.1. Using this simple model, the response of the $\mathrm{NOT}_{r2}^{\mathrm{mut}}$ promoters can be quantified by model fitting. I model The resulting fits for this are display in Table 3.1. The alterations in the response primarily captured in decreases in both the hill-coefficient n and $\frac{k_3}{k_4}$, indicating weaker binding of r2 to the mutated cognate promoters. There may be a general trend of the closeness

	A	$\frac{k_3}{k_4}$	**n**	**b**
$\mathbf{NOT}_{r2}$	1.29×10^3	309.6	1.44	1.05×10^3
$\mathbf{NOT}_{r2}^{\mathbf{A12T}}$	401.6	$2.48 \times 10^{-}3$	1.15	2.48×10^3
$\mathbf{NOT}_{r2}^{\mathbf{G7C}}$	124.2	$8.05 \times 10^{-}3$	1.23	2.61×10^3
$\mathbf{NOT}_{r2}^{\mathbf{A9T}}$	10	$9.94 \times 10^{-}2$	0.84	1.73×10^3

Table 3.1: Parameter fits for mutant NOT gate promoters

of the mismatch to the potency of the mismatch to disrupt repression effects, indicating this response might be tunable.

3.1.4 Molecular titration of gRNA:dCas9-Mxi1 using 2μ and CEN-ARS plasmids

To induce a titration effect in the Cas9 repression system, a 2μ plasmid containing seven r2 gRNA and PAM sites interspaced with 45bp of random DNA sequences was created (called $2\mu^{7\times r2}$). The copy number of 2μ plasmids varies from 10 to 30 copies per cell [**Futcher2984CopyCerevisiae**]), resulting in between 70 and 210 copies of decoy r2 target sites per cell. The presence of these decoy plasmids result in a alteration of the response function (Fig. 3.3). This is primarily seen as a decrease in maximum overall repression of the NOT_{r2} promoter. To further investigate this effect a plasmid was created with centromeric origin of replication (CEN) and an autonomous replicating sequence (ARS) to create a CEN-ARS plasmid. Unlike 2μ plasmids, CEN-ARS plasmids are stably maintained in yeast at just 1-2 copies [78]. From this plasmid seven decoy gRNA r2 sites were inserted to create the CEN-ARS$^{7\times r2}$plasmid. Surprisingly, at just 1-2 copies, the CEN-ARS$^{7\times r2}$plasmid resulted in a similar effect, resulting in a decrease response at the NOT_{r2} promoter (Fig. 3.3). To see if effects were due to the expression of the plasmid itself, I included an empty 2μ plasmid, which did not appear to affect the response. This indicates the presence of just a few additional gRNA sites was sufficient to drastically affect the response function of CRISPR

NOT promoters. Further, this effect is apparently saturated with at least 7-14 gRNA sites as increasing the number of sites to 70-210 did not enhance the effect.

3.1.5 Combining titration and promoter mutations

Combining mutations at the cognate site and providing a molecular titration plasmid did not yield an ultrasensitive response. Five strains were compared NOT_{r2}, $\text{NOT}_{r2}^{\text{A12T}}$, $\text{NOT}_{r2}^{\text{A12T}}$+$2\mu^{7\times r2}$, $\text{NOT}_{r2}^{\text{G7C}}$, $\text{NOT}_{r2}^{\text{G7C}}$+$2\mu^{7\times r2}$(Figs. 3.4, 3.5). Instead of producing a sigmoidal response that would be expected if gRNA preferentially bound to the $2\mu^{7\times r2}$, I instead find that the decrease in response is additive. Addition $2\mu^{7\times r2}$*and* mutations in the cognate target site result in even greater changes in the response function.

3.1.6 Conclusions of titration effect on CRISPR NOR promoters

These results show that the CRISPR NOR gate system is quite sensitive to the effects of molecular titration. Providing just 7-14 additional gRNAs using the CEN-ARS$^{7\times r2}$plasmid appears to shift the response function significantly. Further, these results show that mismatches in the gRNA target sequence can still result in a significant response. Both of these results have important implications for using the system for large-scale circuitry. The first is that many circuit topologies may have a 'forking' wire built into the circuit, meaning a single promoter expressing a gRNA targets more than one different site. This 'forking', for example, was used in the XOR and XNOR circuit design previously reported in this document [31]. It is unclear how sensitive this effect is and whether just providing a single additional gRNA binding site would alter the response significantly. What is a bit troublesome is that providing additional target sites using the $2\mu^{7\times r2}$did not seem to enhance the effect, indicating somehow the effect was saturated when binding sizes $\leq [7, 14]$. This may mean the effect is quite sensitive to the number of binding sites. Consider that the CRISPR dCas9 system exists alongside the rest of the host yeast genome (12Mb). While gRNAs were designed to not target the host genome sequence, these results show that gRNAs can imperfectly match alternative binding sites and this effect can affect the response function

of the CRISPR NOT gates. Given the sensitivity of the system to titration effects, it seems probable that the response function of CRISPR NOT gates is highly sensitive to both misbinding to sites on the host genome and to other binding sites within the circuit. This effect may be responsible for the sequence-specific variances in responses for our CRISPR NOR gates that we previously reported (see above; orthogonality heatmap). When porting the system to other host organisms, alternative binding sites must be considered. The human genome is 3,100 Mbp, roughly 250 times larger than the 12Mb *S. cerevisiae* genome. This provides many more opportunities for dCas9 to misbind to the host genome, which could result in decreased circuit performance and possibly complete circuit failure.

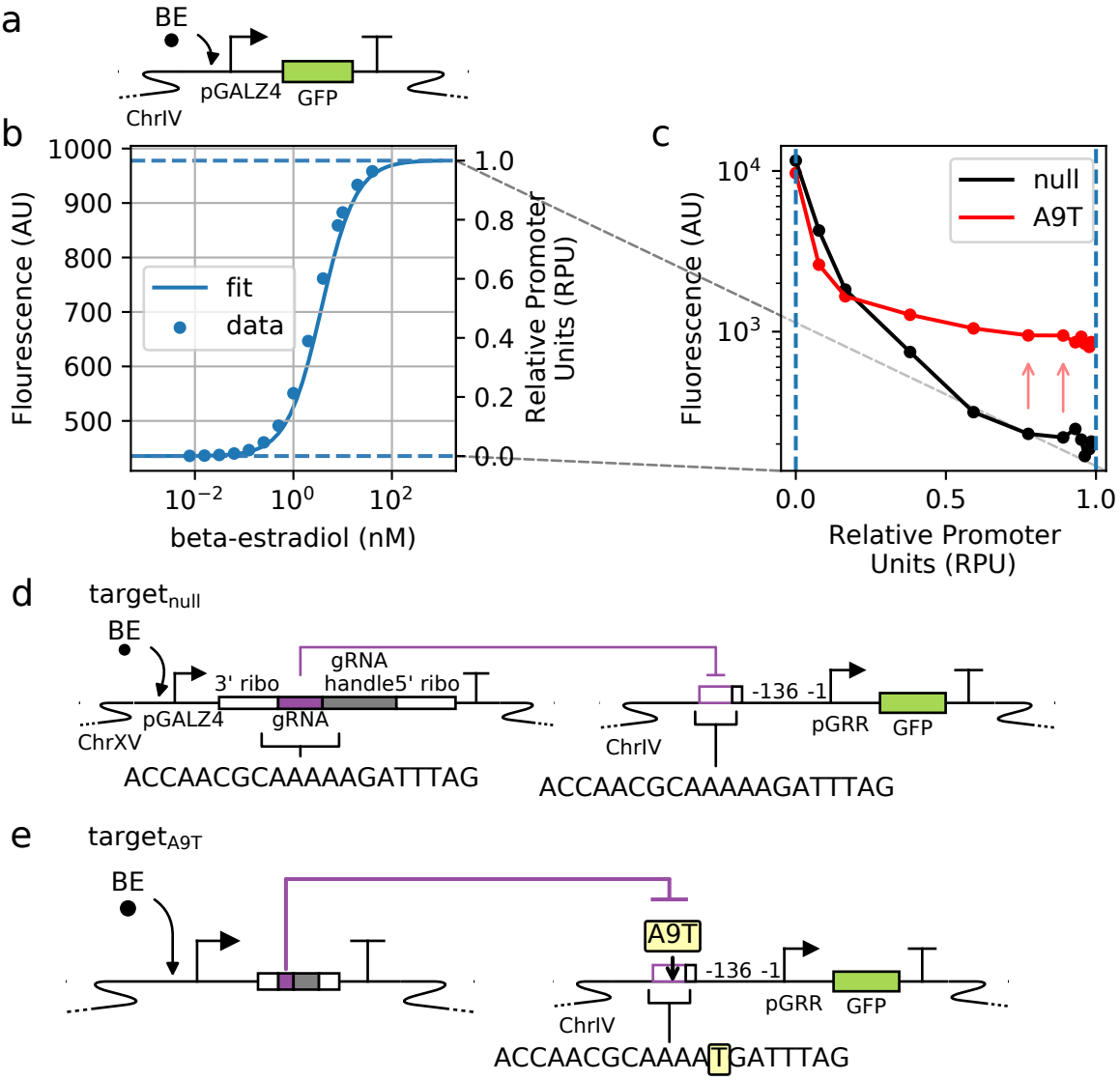


Figure 3.1: **Effect of sequence mutation on dCas9 response** (a, b) An inducible control strain is used for the calculation of relative promoter units (RPU). (c) The responses of CRISPR NOT circuits are plotted against the output of the control strain to determine the individual response function of a gate. (d) A simple one-layer repression system is used to evaluate the performance of a CRISPR NOT gate using an inducible gRNA and a cognate gRNA responsive promoter (c; black curve). (e) Mutations at the cognate promoter cause a mismatch between the expressed gRNA target and promoter, resulting in a decreased response (c; red curve). For brevity, the depiction of additional components such as the ZEV4 beta-estradiol inducible transcription factor and dCas9-Mxi1 cassette are omitted.

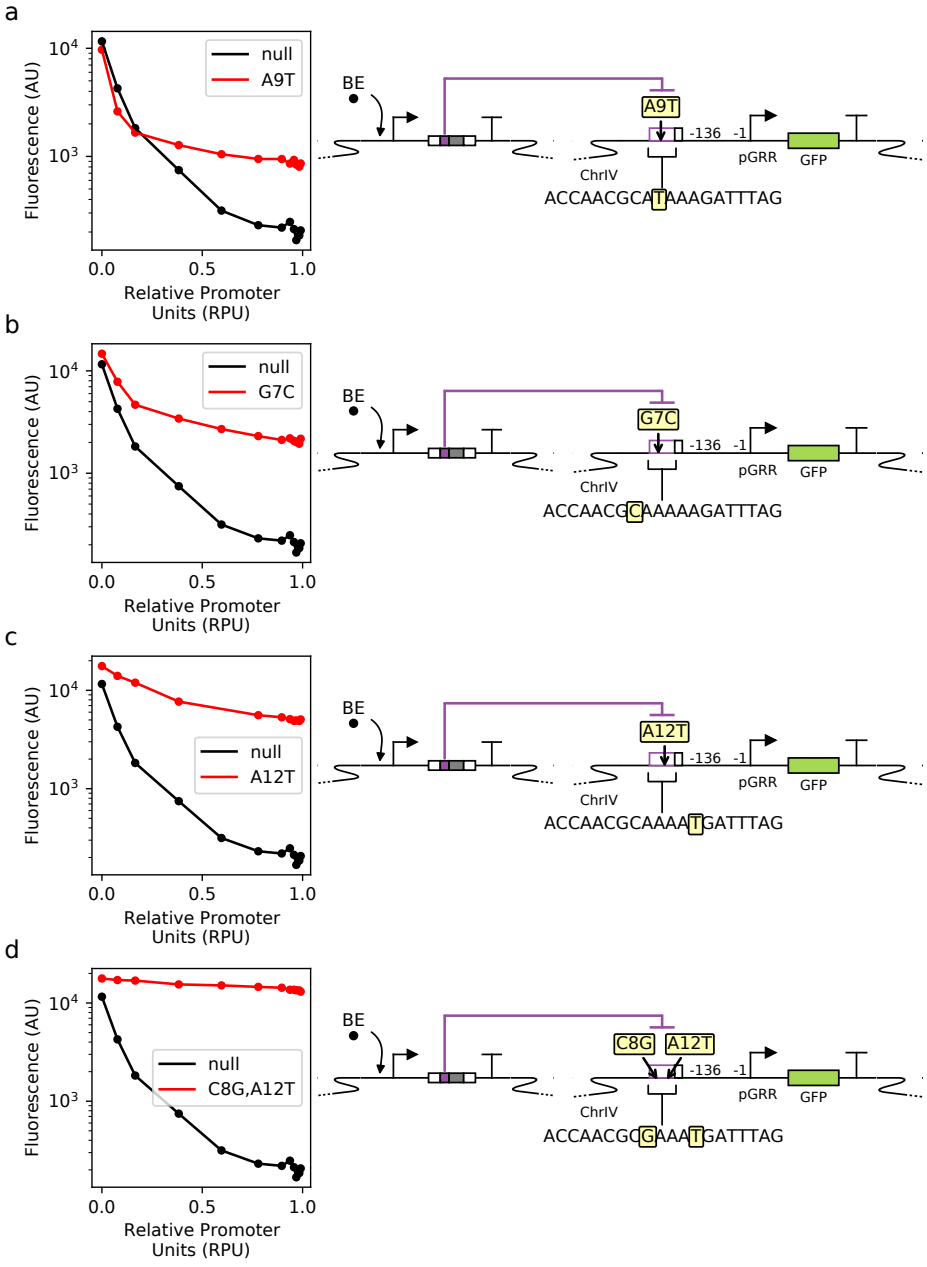


Figure 3.2: **Comparison of effects of mutation** (a-d; red plot) Several mutations were applied to a dCas9-Mxi1 circuit by mutating sequence "ACCAACGCAAAAAGATTTAG" at the specified positions. (a-d; black plot) Unmutated ("null") response shown for comparison.;

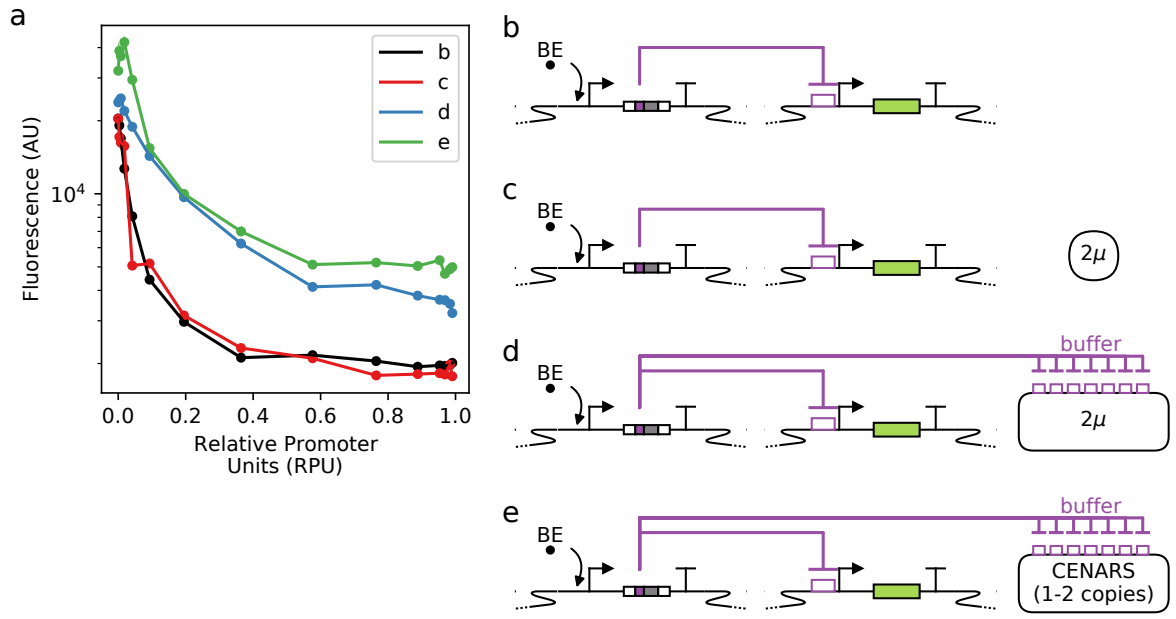


Figure 3.3: **Effects of decoy sites on NOT$_{r2}$** Four strains were compared: NOT$_{r2}$, NOT$_{r2}$+2μ, NOT$_{r2}$+2$\mu^{7\times r2}$, and NOT$_{r2}$+CEN-ARS$^{7\times r2}$.

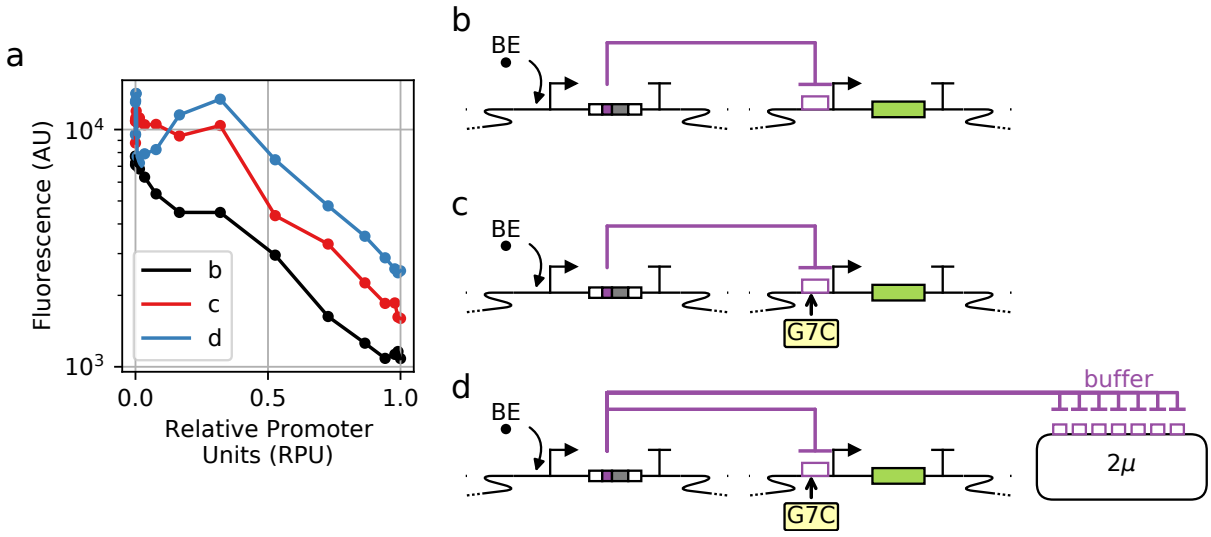


Figure 3.4: **Titration effect for mutated G7C mutant NOT gate promoters** Titration effects of NOT$_{r2}^{G7C}$+2$\mu^{7\times r2}$were compared against NOT$_{r2}$ and NOT$_{r2}^{G7C}$. The decrease in NOT$_{r2}$ responses when using mutated cognate site G7C and additional 2$\mu^{7\times r2}$gRNA sites is additive.

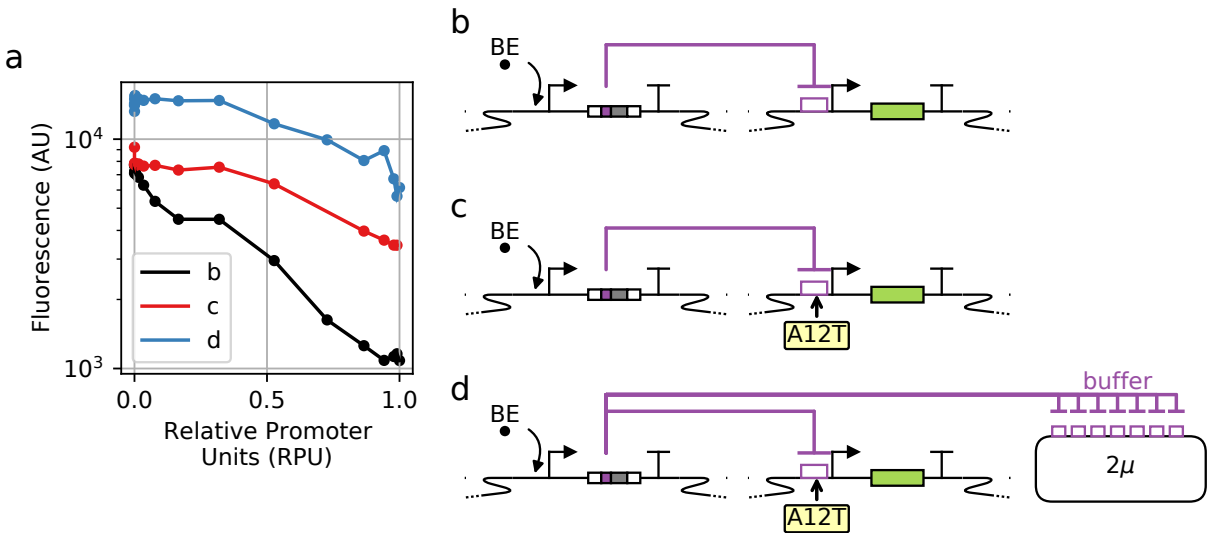


Figure 3.5: **Titration effect for mutated A12T mutant NOT gate promoters** Titration effects of $\text{NOT}_{r2}^{\text{A12T}}$+$2\mu^{7\times r2}$were compared against NOT_{r2} and $\text{NOT}_{r2}^{\text{A12T}}$. The decrease in NOT_{r2} responses when using mutated cognate site A12T and additional $2\mu^{7\times r2}$gRNA sites is additive.

3.2 CRISPR-dCas9 is unlikely to yield bistable systems using simple toggle switches

I previously reported on using the CRISPR-dCas9 system for combinatorial circuits. These circuits have no feedback or elements of memory. Memory is a key element in digital systems and developing complex behavior as they allow for the storage and processing of past information. Given the successes of the CRISPR dCas9 system, it is desirable to implement memory systems, that is to move from combinatorial circuits to sequential circuits (circuits with feedback or memory elements). However, as I have previously shown, the CRISPR-dCas9 NOR and NOT gates have shallow response characteristics. This makes it difficult to create sequential circuits as states in between fully "OFF" and "ON" results in degradation of information throughout the circuit. Here I perform a simple simulation and *in silico* analysis of a basic memory element, the **flip-flop** constructed out of CRISPR dCas9-Mxi1 NOR gates. The flip-flop consists of two CRISPR NOR gates wired with feedback. Two inducible promoters, "pGALZ4" and "pGALZ4 ortholog" serve as inputs R ("reset") and S ("set") to the system. Outputs are modeled as reporters "Q" and "Qbar". Parameters were re-sampled from best-estimated fits as determined from previous analysis [31] to generate 1000 randomly generated circuits (Fig. 3.6A). Similar to models described previously for the CRISPR Cas9 system, the flip-flop is modeled as follows:

$$\begin{aligned}\frac{dq}{dt} &= b\left(\frac{v_{\overline{q}}}{1+(\frac{r}{k_1})^n+(\frac{\overline{q}}{k_2})^n}-q\right)\\ \frac{d\overline{q}}{dt} &= b\left(\frac{v_{\overline{q}}}{1+(\frac{s}{k_3})^n+(\frac{q}{k_4})^n}-\overline{q}\right)\end{aligned} \tag{3.3}$$

where r, s, q, and $\overline{q}$ correspond to concentrations of gRNAs, v is the promoter strength, k is the dissociation constant for a given promoter site and b is the degradation of the system (by cell division). For simplicity, I omit modeling of the inducible promoters that produce r and s. Bistable behavior was determined by simulating changes inputs r and s and evaluating the output responses of q and $\overline{q}$ expression levels in response to changing inputs R and S. Of these circuits about 5-6% exhibited robust bistable behavior (Fig. 3.6). Analyzing the parameter

sensitivity of the system, higher promoter strengths at the NOR gate promoters tended to correlate with a higher probability of bistable circuits compared to non-bistable circuits, which indicates that the gRNA expression strength is a critical factor for creating a bistable flip-flop. Other parameters seemed to be only weakly correlated with circuit bistability.

Overall this simple analysis shows that the creation of sequential circuits, like flip flops, is challenging using the CRISPR dCas9-Mxi1 system. This result was corroborated by other students in our group, who constructed over 20 different flip-flops, none of which exhibited bistability.

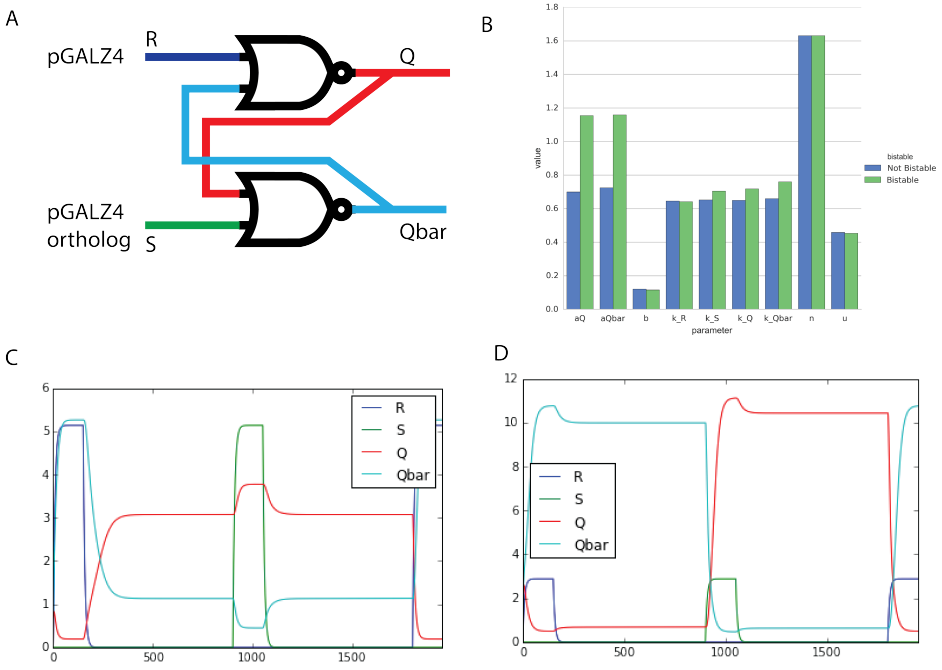


Figure 3.6: **Flip-flop analysis results** (A) Model of the flip-flop. Flip-flop was modeled as two wired pGRR NOR gates. Inputs R and S were modeled as gRNAs being driven by pGALZ4 promoters. (B) Circuit characteristics of bistable circuits. Mean parameter values of bistable vs non-bistable circuits are displayed. Parameters for promoter/expression strengths (a), gRNA repression strengths (k), pGALZ4 promoter strength (u), and gRNA degradation/dilution were resampled from best estimates for 1000 Monte Carlo simulations. (C) Example of a non-bistable circuit generated by Monte Carlo simulation. Pulse of the gRNA S results in no change in states of Q and Qbar. (timepoint=1000, green curve). (D) Example of a bistable circuit generated by Monte Carlo simulation. Q and Qbar successfully switch upon the pulse of gRNA S.

3.3 Circuit topologies for systems that exhibit hysteresis

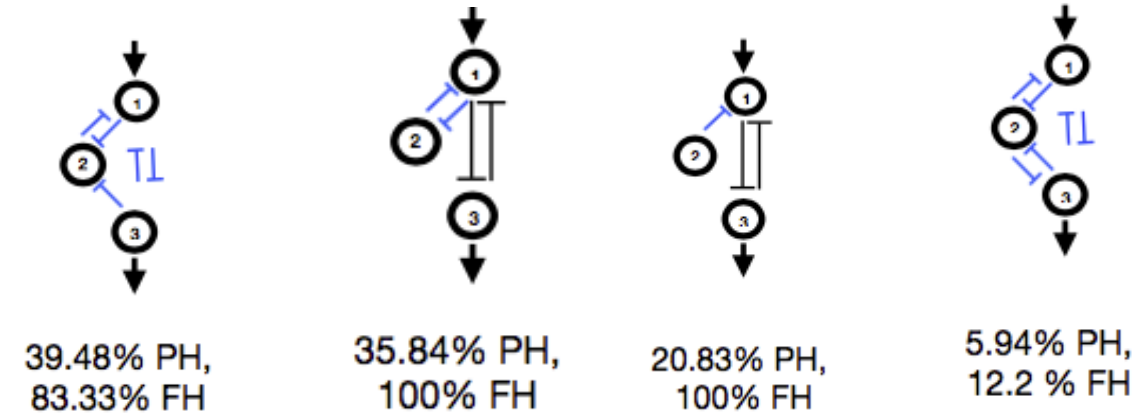
Though it seems unlikely the CRISPR-dCas9 system would yield bistability from a flip-flop architecture, it may be possible to exhibit bistable systems with alternative designs. We explore designs beyond the bistable toggle switch that exhibit hysteresis, which is the general property of a system in which its state depends on its history of past states. Hysteresis makes it possible to develop systems with persistent memory.

Here I perform an analysis is to determine the feasibility of finding CRISPR/dCas9-Mxi1 based NOR gate circuit topologies that exhibit hysteresis. A new computation tool Dynamic Signatures Generated by Regulatory Networks (DSGRN)[1] was used to identify 3-node circuit networks that have a high probability of exhibiting hysteresis. Briefly, the tool scores the dynamics of circuit topologies, represented as directed graphs, by assuming each node in a circuit network exhibits a step-like response behavior. This step-like behavior is very computationally efficient to compute and so the tool examines a massive number of different circuit topologies for a user-provided behavior. In this case, we used the DSGRN tool to examine all small 3-node circuit networks that would exhibit hysteresis properties. These circuit topologies were then mapped to the *S. cerevisiae* CRISPR-dCas9 genetic circuit system previously described [31].

3.3.1 Parameter sensitivity of hysteresis using Monte Carlo simulation

Monte Carlo simulations using experimental parameter bounds (Table 2.2) were performed for the circuit designs. 3-node circuit networks were implemented using CRISPR NOR gates (Figs 3.8, 3.9). Two inducible promoters were included in the design, one for input activating node 1, and one for resetting for repressing node 1 (Fig. 3.9). These inducible promoters were modeled after the beta-estradiol (βe) inducible promoters previously described. NOR

[1]Developed by our collaborators Bree Cummins, Thomas Gedeon, Shaun Harker (Montana State University), and Konstantin Mischaikow (Rutgers)



(a) Rank 1: 'Good' (b) Rank 2: 'Good-2' (c) Rank 3: 'Medium' (d) Rank 4: 'Poor'

Figure 3.7: **DSGRN design circuit topologies** Selection of networks received from the DSGRN design tool filtered by topologies that used repressors only. The percentage of found paths that exhibited partial hysteresis (PH) or full hysteresis (FH) are displayed below. (a,b) For analysis, two high-performing circuits were selected with (c) one medium-performing circuit and (d) one poor-performing circuit. (c) The medium performing circuit is functionally equivalent to a bistable toggle switch

gates were modeled using the following ODE equation:

$$\frac{dr_2}{dt} = b\Big(\frac{a_2}{1 + (\frac{r_0}{k_1})^n + (\frac{r_1}{k_2})^n} - r_2\Big) \tag{3.4}$$

where r_2 is the output gRNA concentration, r_0, r_1 are input gRNAs to the NOR gate, b is the degradation, a_2 is the transcription strength of the output promoter for gRNA r_2, and k_1, k_2 are the dissociation constants for each gRNA site. For brevity, I omit the full ODE model for these networks, as the full system of equations is repetitive, but the full system of equations is straight-forward to generated by parsing a directed graph representing the circuit topology and merely matching variables r_i to input variables on the denominator entry $(\frac{r_i}{k_j})^n$ for every edge in the graph. During Monte Carlo simulations, dose responses were measured using the inducible input node and measuring node 2 as the output. ODE models were equilibrated to steady-state at low input and slowly raised to high input to capture an ascending or rising dose-response. Then ODE models were equilibrated to steady-state at high input and slowly

lowered to low input to capture the descending or falling input. Rising and falling inputs were compared and the differences in the area under the curve (AUC) were used to quantify hysteresis in the circuits exhibited (Fig. 3.9). This was quantified as the area between the rising and falling trajectories as $\delta = |\mathrm{AUC}_{\mathrm{rising}} - \mathrm{AUC}_{\mathrm{falling}}|$, where AUC is the area under the curve for each dose-response trajectory. In most cases, descending dose responses did not return to the original steady-state, requiring induction of the reset gRNA to return to the previous state (Fig. 3.9; bottom right).

Circuits were analyzed at different parameter regimes by simulating at least 10,000 dose-response curves. Curves were collected and the differences in curve-areas were recorded as δ (Figs. 3.8, 3.9). As predicted by our collaborators, very high hill-coefficients using the 3-node circuit topologies resulted in dose-response curves with substantial separation between rising and falling dose-response curves, i.e a high δ (3.9; right-panel). However, experimentally, the NOR gates have a hill-coefficient of around 1.7 to 2.2. MC simulations for all circuit topologies using experimentally predicted parameters were predicted to fail 80-90% of the time for the best circuit topologies (Fig. 3.10; green bar). At high transcriptional rates or high hill-coefficients, the percentage of hysteretic circuits increases substantially.

3.3.2 Parameter sensitivity analysis

Experimentally, transcription rates promoters are relatively easy to manipulate. Adding new promoter enhancers or switching promoter sequences can result in higher or lower transcription rates. Transcription rate, therefore, is a tunable parameter of the system. To examine the effect of transcription rate on hysteresis for designed circuits, 60000 circuits were generated using experimental fits, but with a wider transcription rate sampling range ($3 < a < 25$). Parameters for each a and k for the three NOR gates were recorded. k plots were uninteresting within the parameter regime sampled. Comparisons for binned a_0, a_1, and a_2 (corresponding to promoter strengths of nodes 0, 1, and 2 respectively) were made for the percentage of hysteretic circuits within that bin.

Top-ranked circuits designed by DSGRN were also found to exhibit the highest perfor-

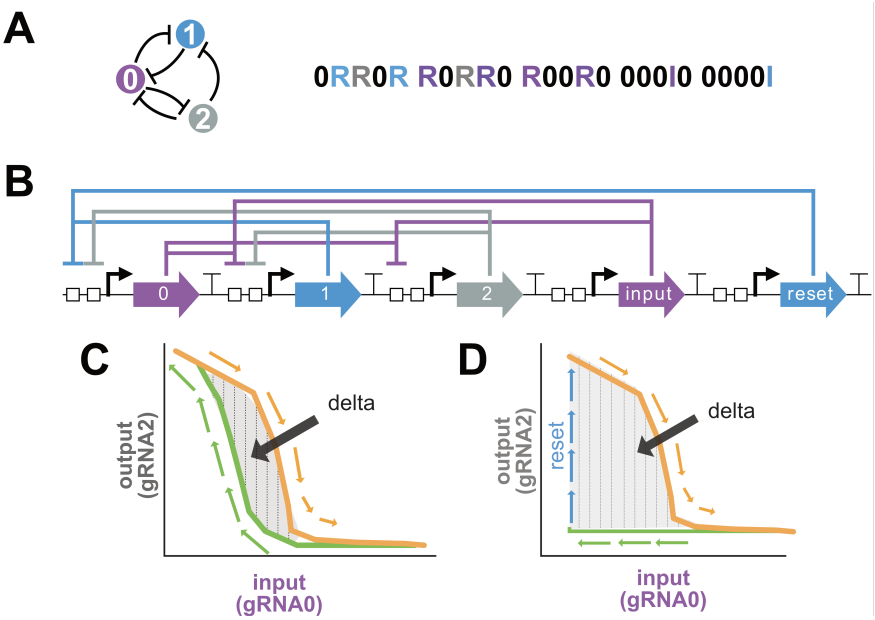


Figure 3.8: **Conversion of DSGRN topologies to gene circuits** Circuit networks were converted from provided network graphs (A) to a NOR gate implementation and the string representing the circuit topology as received by the DSGRN software tool. This string representation was parsed to a genetic network using custom scripts. (B). To simulate inputs, two additional inducible promoters were simulated, an 'input' and 'reset' cassette. Dose responses for the circuits were simulated by inducing the 'input' gRNA and measuring the output gRNA 2. Hysteresis was estimated by the different between the *rising* (orange) and *falling* (green) trajectories. The area between these two trajectories was used in the calculation of δ ("delta"), which quantifies the degree of hysteresis (C and D). In many cases, dose-response paths would not recover without inducing the 'reset' gRNA (D; blue)

mance in ODE simulations (Fig. 3.11). Between 10-20% of simulated circuits were predicted to exhibit hysteresis. In comparison, only $< 5\%$ of simulated circuits using bottom-ranked designs showed hysteresis (Fig. 3.12). Interestingly, this analysis indicates that increasing transcriptional strength could greatly improve hysteresis behavior. As an example, doubling the transcription rate of a_0 and a_2 results in 50% of simulated circuits exhibiting hysteresis (Fig. 3.11; right panels). In almost all cases, increasing the transcription rate would increase the chances of hysteresis. The exception to this rule is with circuit "Good2", in which increasing a_2 alone can result in decreased performance (Fig. 3.11; right panel, follow heatmap

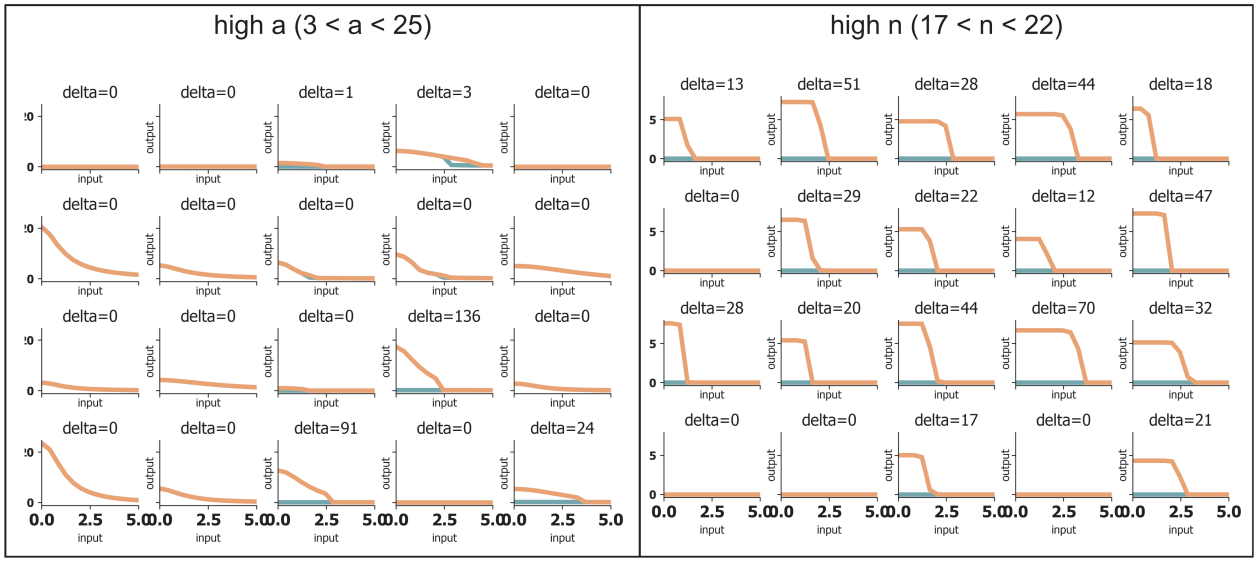


Figure 3.9: **Selected simulations of 3-node circuits** Several dose-response traces for rising (orange) and falling (green) dose-response curves for several circuits in high transcriptional parameter regime (left) and high hill-coefficient parameter regime (right). To determine whether a circuit exhibited hysteresis, delta was set at a threshold of 10 AU, which was visually chosen based on plots like those displayed above; the delta threshold of 10AU was selected since it seems like enough of a difference that would be observable in an experiment.

up from white square). The 'medium' circuit topology, which essentially implements a flip-flop, did not exhibit hysteresis; this may be due to an artifact in how delta was measured via rising and falling dose responses and is not an indication that none of these circuits exhibited bistability.

3.3.3 Conclusion

Using a new computational tool, Dynamic Signatures Generated by Regulatory Networks (DSGRN), and ODE modeling from experimentally derived parameters, I was able to show that alternative circuit topologies can result in hysteresis behaviors. These results are encouraging as they demonstrate that some limitations of individual circuit parts can be overcome using circuit topology alone.

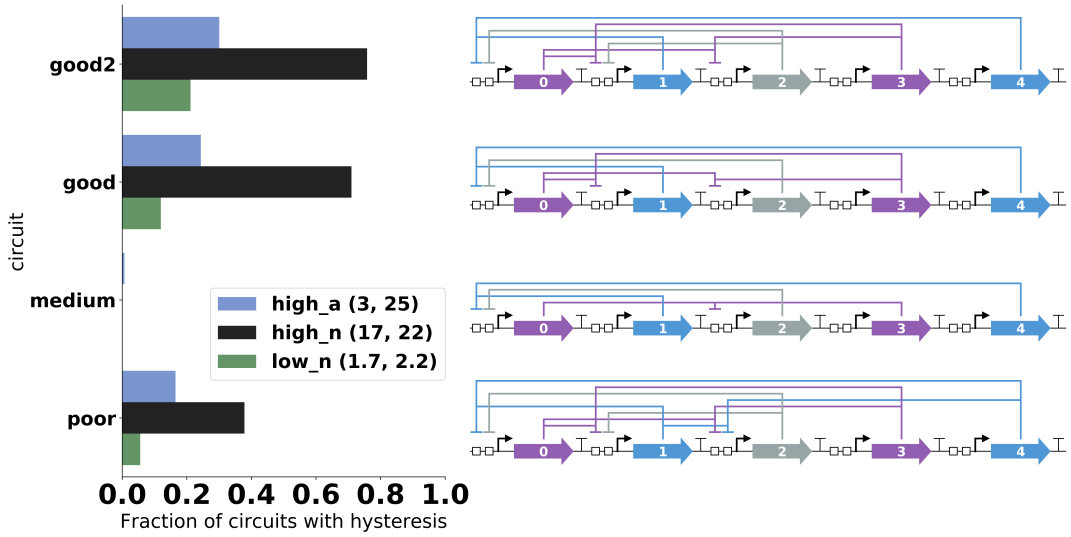


Figure 3.10: **Robustness of hysteresis for circuit networks** Hysteresis summary for the four circuit topologies for high transcription (high a; blue bars), high hill-coefficient (high n; black bars), and experimentally determined parameters (low n; green bars). To classify a circuit as having hysteresis, the area between the rising and falling dose-response curves were calculated (δ) and those beyond a threshold were classified as having hysteresis ($\delta > 10$). At least 10,000 circuits were created using for each parameter sampling and circuit topology. The plot on the left displays the fraction of circuits with hysteresis. The corresponding circuit implementations are on the right. Surprisingly, the 'medium' circuit topology, which implements the flip-flop, did not exhibit hysteresis in this simulation; this may be due to an artifact in how delta was measured via rising and falling dose responses and is not an indication that none of these circuits exhibited bistability.

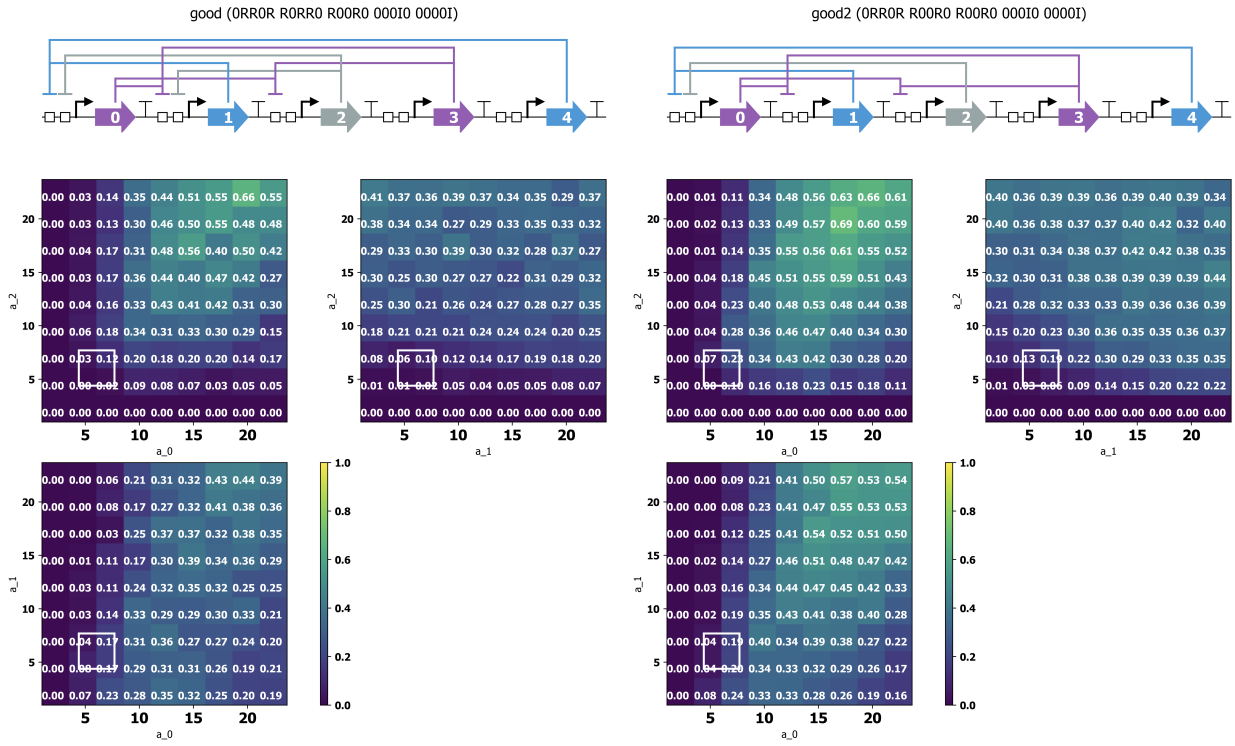


Figure 3.11: **Parameter sensitivities for hysteresis for top-ranked designs** The 3-node circuit network (top) was evaluated for the robustness of hysteresis using a Monte Carlo simulation. To evaluate robustness over parameters, MC simulation across parameter regimes was evaluated and the percentage of circuits exhibiting hysteresis was evaluated for approximately 10,000 circuits. gRNAs 3 and 4 (top panel; right) were used as inducible gRNAs. Rising and falling dose responses were measured for gRNA3. For the Monte Carlo simulation, parameters were as follows: transcriptional strength (a, between 3-25 AU), repression strength (k; between 0.9, 3.65), and hill-coefficient (n; between 1.7 and 2.23). Experimentally predicted values for parameters are hill-coefficient (n; between 1.7 and 2.23) [31]. The white box represents the experimentally predicted parameter regime. Colorbar: Fraction of circuits with hysteresis ($\delta > 10$).

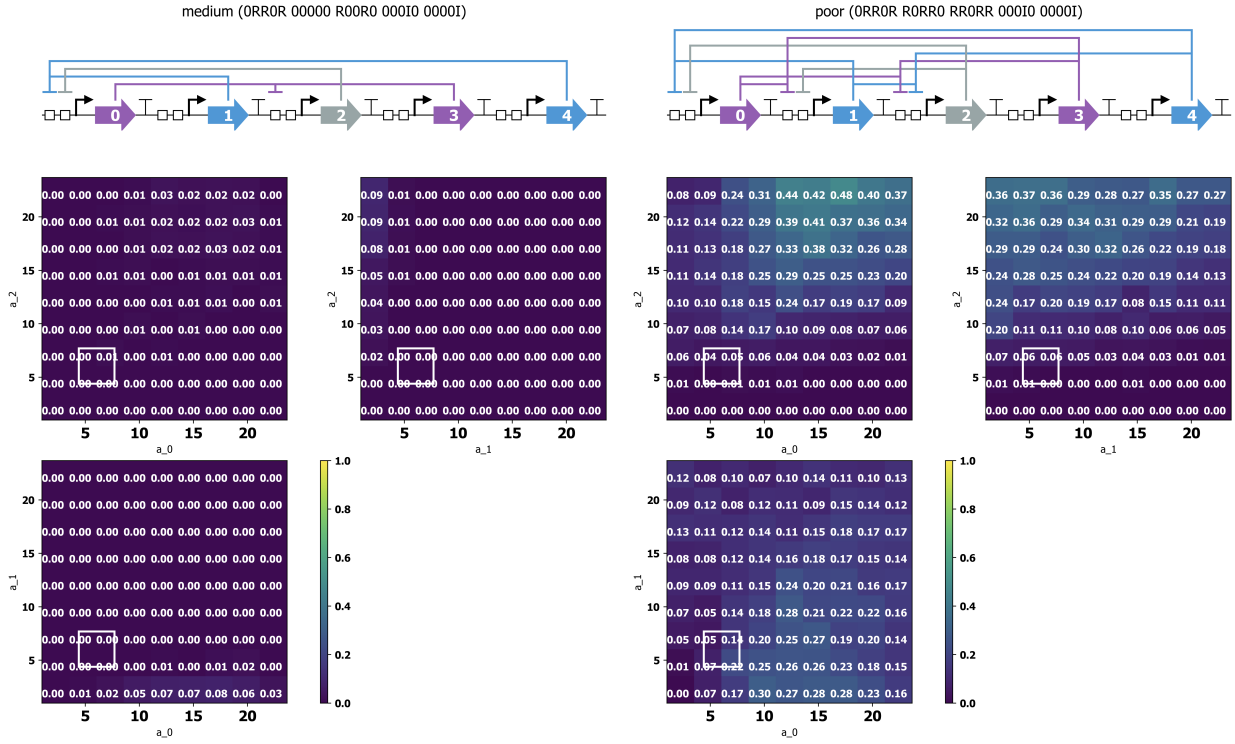


Figure 3.12: **Parameter sensitivities for hysteresis for lowest-ranked designs** Bottom ranked DSGRN designs rarely showed significant hysteresis. Colorbar: Fraction of circuits with hysteresis ($\delta > 10$).

Chapter 4

A MAMMALIAN CRISPR-DCAS9 NOR GATE

Author	Contribution
Justin Vrana	sole author

4.1 Abstract

Here I present a mammalian CRISPR NOR gate based on the bipartite dCas9-KRAB and tripartite dCas9-KRAB-MECP2 transcription factors. The NOR gate is based on the Pol III promoter, using the human U6 promoter (hU6) to drive a small gRNA transcript and was engineered to be very small (535bp) to lend itself to be easily synthesized or packaged into lentiviral vectors. I compose the NOR gate into a 3-layer OR circuit and demonstrate its performance in both hamster (CHO) and human (HEK293) cell lines using co-transfection experiments. In these experiments, I report, for the first time, the ability to repress a Pol III using CRISPR interference. I demonstrate that the NOR gate has excellent performance characteristics, as co-expression of either of input gRNA in the OR circuit completely unrepress a pSV40-GFP reporter plasmid. There was no difference in the response when targeting either of the two input gRNA sites. The NOR gate was engineered to be very small, at just 535bp from promoter to terminator. These results are exciting as they show a new strategy for developing gene circuity in mammalian cells using CRISPR interference Pol III.

4.2 Results

Repression of Polymerase II (Pol II) promoters using CRISPR interference (CRISPRi) in mammalian cells has been previously reported [30, 79–82]. Canonically, Pol II promoters typically express mRNA transcripts that eventually are translated to proteins. In mammalian cells, a gRNA:dCas9 complex can inhibit transcription of Pol II promoters by targeting the region between -50 and 250 around the transcriptional start site (TSS) [82]. In yeast cells, this optimal repression region is between -200 and 50bp of the TSS [55], which is corraborated by our groups results on dCas9-Mxi1 described previously in this document [31].

Our group previously reported (as described in this document) the ability to engineer a synthetic NOR gate by engineering positions around the TATA box of a strong yeast Pol III pGPD promoter. This allows for quickly composing new genetic NOR gates by selecting an input NOR promoter sequence (containing two gRNA target sites) and concatenating it

with either a coding sequence cassette (e.g. GFP) or a ribozyme flanked gRNA cassette. In mammalian cells, however, this engineering strategy may not work since the optimal repression region for Pol II promoters in mammalian cells is well after in the TSS. The -50 to 0 bp region immediately before the TSS does not give much flexibility in engineering a new mammalian promoter. The 0 to 250 bp region following the TSS is in the transcribed region of constructs and in the 5' untranslated region (5' UTR) of coding sequences; the 5' UTR contains important sequences, like ribosome binding sites (RBS), required for efficient protein expression. Engineering this region to contain a selection of variable gRNA target sites may be challenging as editing this region could drastically affect protein expression. Hence, it seems difficult to decouple the gRNA input target sequence from downstream expression. Additionally, Pol II promoters and terminators in mammalian cells are fairly large, on the order of thousands of bp. Long DNA construct sizes make both DNA assembly and costly. In mammalian cells, many genomic integrative strategies like lentiviral transduction become increasingly inefficient as DNA construct size increases. Integration with lentivirus, for example, is most efficient with construct sizes below 5kb [83, 84]. Taking all of these issues together, it appears even with a Pol II CRISPR NOR gate, developing complex circuitry in mammalian cells would be challenging, inefficient, and costly.

An engineered CRISPR Pol III NOR gate would have several advantages over a CRISPR Pol II NOR gate in mammalian cells. Pol III promoters canonically express small RNAs at high levels. The Pol III promoter sequences are very small on the order of hundreds of bp and their terminators are minuscule on the order of tens of bp. RNA expressed off of Pol II promoters are naturally retained in the nucleus and therefore do not require post-processing components like ribozymes or other RNA cleaving motifs to express nuclear gRNAs. Further, Pol III systems are very small lending themselves to be easily synthesized and packaged into lentiviral systems. The U6 promoter, for example, is around 250bp and the U6 terminator is only 10bp; in contrast, the commonly used pEF1 promoter is 1148bp and the WPRE terminator is 528bp. However, there are not many literature examples that show that Pol III can be regulatable [85, 86]. Because of this, Pol III promoters in synthetic biology are

Feature	Pol II	Pol III
Expresses coding sequences	Yes	No
Expresses nuclear RNA	Requires cleavage of 3'-polyA and 5'-cap	Yes
Promoter size range (bp)	10^3	10^2
Terminator size range (bp)	$10^2 - 10^3$	$10^0 - 10^1$
Regulatable	Yes	Yes; this study

Table 4.1: Advantages and disadvantages of using Pol II or Pol III for CRISPR circuits

often just to constructively express small RNA, such gRNAs. Despite the absence of reported CRISPR interference in Pol III promoters, it is not clear that CRISPR interference does not occur in Pol III constructs.

To engineer a CRISPR-based Pol III NOR gate, the human U6 Pol III system was used. Instead of engineering the promoter, the transcript sequence following the TSS was engineered to contain two inverted input gRNA target and PAM sequences immediately following the U6 TSS, denoted as the inputs IN_A and IN_B (Fig. 4.1). The input region is followed by a gRNA expression sequence OUT_C that is flanked by two stem-loop structures ($\text{Csy4}_{\text{hairpin}}$) to which a Csy4 endoribonuclease can bind to and cleave. The Csy4 (Cas6f) endoribonuclease derives from *Pseudomonas aeruginosa* [87] and is involved in native processing of CRISPR RNAs (crRNAs) [87]. The Csy4 protein has been repurposed in other studies to perform multiplex gRNA expression and to cleave 3' and 5' transcriptional caps caused by expression from Pol II promoters[88–91]. While likely unnecessary for expression of single gRNAs from Pol III U6 promoter, as Pol III promoters do not add 5' RNA caps or 3' poly-A tails, the Csy4 hairpins were included in the design so that it permits expression of multiple gRNAs and limit interactions between the leading IN_A, IN_B sequences and the RNA folding of the gRNA scaffold. An optimized gRNA scaffold is used that has been shown to resolve issues of premature termination of Pol III transcribed gRNAs [92, 93].

To repress the NOR gate, a dCas9 transcription factor, dCas9-KRAB or dCas9-KRAB-MeCP2, is co-expressed along with either gRNA sequences that correspond to the input

sequences IN_A or IN_B. Binding of the gRNA:dCas9 causes inhibition of transcription of the output gRNA OUT_C. Either the dCas9-KRAB or tripartite dCas9-KRAB-MeCP2 transcription factors. Krüppel associated box (KRAB) domain has been shown in many studies to be an effective transcription factor for dCas9-mediated programmable gene silencing [53, 54, 94–97]. More recently, an engineered bipartite transcription factor, KRAB-MeCP2, has been shown to greatly improved improve dCas9-mediated gene silencing in mammalian cells [79] over KRAB alone.

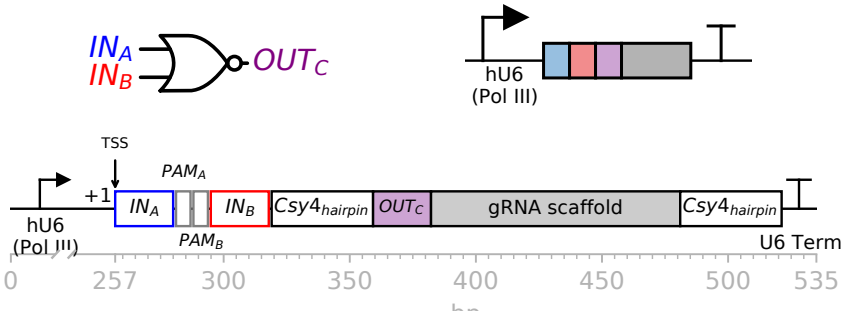


Figure 4.1: **Annotated sequence of the mammalian NOR gate.** Human U6 Pol III promoter drives a gRNA transcript. Two inverted CRISPR gRNA target sites (IN_A, IN_B) immediately succeed the transcriptional start site (TSS). Sequences for two Csy4 RNA hairpins flank a gRNA expression sequence comprised of a 20bp target sequence ("20bp") and an optimized gRNA scaffold that has been shown to resolve issues of premature termination of Pol III transcribed gRNAs [92, 93]. Co-expression of the Csy4 protein cleaves processes the transcribed RNA at the location of the Csy4 hairpins, leaving a small 8bp and 26bp scar sequence on the 5' and 3' ends respectively. The Csy4 RNA processing removes the leading 73bp of sequences (including the IN_A, IN_B sequences) and any sequences following the second Csy4 hairpin. Overall, the construct is only 535bp from the start of the U6 promoter to the end of the U6 terminator making it trivial to synthesize.

To test the performance of the engineered NOR gate, an OR circuit was constructed in Chinese hamster ovarian (CHO) cells (Fig. 4.2) using the dCas9-KRAB-MeCP2 engineered transcription factor. A Pol III NOR gate, designated as $NOR_{1,5,SV40}$ was constructed with input gRNA sequences r_1 and r_5[1] and an output gRNA sequence that targets well-documented and constitutively expressed fluorescent reporter, pSV40-GFP [53, 81]. The gRNA sequence

[1]The r_1 and r_5 gRNA target sequences correspond to the gRNA target sequences explained above in this document for CRISPR yeast circuits and published in Gander et. al 2017 [31]

used to target the pSV40 in Zalatan et. al Circuit components were co-transfected into CHO cells using polyethylenimine (PEI). To evaluate the circuit, certain components were selected to be replaced by the same amount of empty vector (pcDNA). Plasmids constitutively expressing gRNAs from U6 promoters were used as the "inputs" to the OR circuit. **OR10** corresponds to co-transfection of all components except the second input gRNA r_5, which was replaced by dummy plasmid pcDNA, **OR01** corresponds to replacing r_5 with empty vector, and **OR00** corresponds to replacing both gRNAs with empty vector. GFP was quantified using flow cytometry and converted to relative units (rel. u) described in the methods below and similar to previously described methods [98].

When the $\mathrm{NOR}_{1,5,\mathrm{SV40}}$ is unrepressed, as seen in the **OR00** sample, we see 4-fold repression of pSV40-GFP reporter over the reporter only sample, indicating expressed transcript from the NOR gate is functional (Fig. 4.3). This fold-repression is less than reported when using a stably integrated reporter and the exact same SV40 gRNA targeting sequence; Zalatan and colleagues reported 10-fold repression with dCas9-KRAB [53] while Gilbert and colleagues reported a 15-fold repression using dCas-KRAB and 2.5-fold repression using dCas9 only [54]. However, in this study, the reporter is transiently transfected, which may affect the ability of dCas9-KRAB-MeCP2 to repress the pSV40-GFP reporter. The reported 4- to 7.5-fold repression was consistently seen when experiments were repeated in both CHO-MIHAC and HEK293 cell lines (Fig. 4.4) and so represents the maximum repression we can measure using transiently transfected pSV40-GFP reporter and dCas9. Upon expression of either the r_1 or r_5 gRNAs, we see a 100% de-repression of the pSV40-GFP back to previous levels, as seen in the **OR01** and **OR10** circuit states.

We found the dCas9-KRAB-MeCP2 driven OR circuit performed significantly better than the dCas9-KRAB circuit when circuit components were co-transformed into CHO cells (Fig. 4.4). This increased performance of the KRAB-MeCP2 over KRAB is also reported by Yeo and colleagues [80]. Here, we see dCas9-KRAB-MeCP2 causes 7.5-fold repression while dCas9-KRAB results in 2.3-fold repression. When either input gRNA is expressed, there is complete de-repression of the reporter for both dCas9-KRAB-MeCP2 and dCas9-

KRAB as seen in the $OR10$ and $OR01$ states. When the experiment was repeated in human embryonic kidney cells (HEK293), we found a similar trend with both transfection factors causing repression in the **OR00** state and being completely de-repressed in the **OR10**, **OR01**, and **OR11** states. However, equipment failures in the cytometer prevented drawing any strong conclusions from this particular experiment.

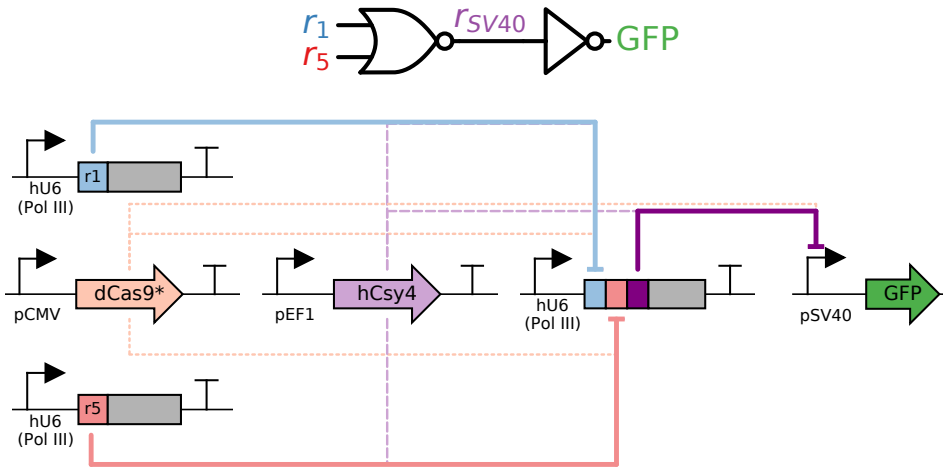


Figure 4.2: **A three-layer mammalian CRISPR dCas9-KRAB-MeCP2 OR circuit** A mammalian OR circuit. dCas9* indicates either the dCas9-KRAB or dCas9-KRAB-MeCP2 transcription factors.

4.2.1 Discussion

In this chapter, I presented a small Pol III NOR gate and performed some initial characterization of it by composing the NOR gate into an OR circuit in HEK and CHO cells. The small size (<550bp) of the construct lends itself to be easily synthesized or packaged into lentivirus for the creation of stable cell lines. Further, it may be possible to string together many NOR gates in series to create complex circuitry, provided there is some insulation between individual gates.

While the results are promising, further characterization is required. Firstly, all experiments involved co-transfection experiments which are notoriously finicky. Decreases in

transfection efficiency can be easily misinterpreted as changes in fluorescence in the cell. To mitigate the effects of different transfection rates, in these experiments, DNA ratios between all components were very carefully maintained as to mitigate the effects of transfection rate. All data was presented as coming from the same exact co-transfection experiment as plotting and comparing data from different experiments would be inappropriate without an appropriate internal transfection control (e.g. a construct expressing an orthogonal fluorescent protein like mKate, iRFP, or mCherry). Instead of co-transfection experiments, performing a proper dose-response characterization using a stably integrated NOR gates would further validate the circuit. Finally, further investigation into the effects of Csy4 are required. This construct was added as part of another project to create libraries circuits for analysis. It is unclear from these experiments what the effect of 5' and 3' cleavage is on NOR gate performance given that gRNAs are expressed from Pol III promoters and do not have 5' and 3' modifications anyways.

The small size of the Pol III mammalian NOR gate lends itself to efficient lentiviral packaging and transduction. Libraries of different NOR gates could be synthesized to create a library of lentiviral particles; transfecting the lentiviral particles into a mammalian cell line containing an integrated dCas9-KRAB-MeCP2 construct would yield randomly generated genetic circuits, which could be analyzed in a high-throughput fashion using RNAseq[2].

[2]I designed and created such a library and ordered it from Twist Biosciences for cloning into a standard vector via golden gate RSII cloning. However, this was around March 2020, immediately before the SARS-CoV2 pandemic after which all mammalian cell lines and this project was terminated; unfortunately because of this, the project never came to fruition. The library consists of sets of both two- and three-input NOR gates, each uniquely barcoded on the 3' end for measurement RNA seq and barcode matching using NGS. The constructs are small enough that NGS can completely read through the barcode and relevant sequences of the NOR gate construct. The number of different gRNA binding sites included in the library was carefully calculated such as to maximize the potential for interesting circuits to form when generating random circuits in cells.

4.3 Materials and Methods

4.3.1 DNA assembly

Backbone and insert fragments were amplified with PCR, gel extracted, purified, and assembled using Gibson assembly [60] using standardized assembly linkers. A derivative of the pcDNA plasmid was used as the backbone for all constructs.

4.3.2 Mammalian cell culture

Human embryonic kidney (HEK) 293T cells were cultured in Dulbecco's modified Eagle's medium (DMEM) without antibiotics and 8% fetal bovine serum (FBS). Chinese hamster ovarian cells (CHO) were cultured in Ham's F12K media with 8% FBS without antibiotics. Cells were cultured in tissue culture-treated T25, T75, and T125 plates in a HEPA-filtered CO2 incubator at 5% CO_2, 37C. To passage, media was aspirated and washed phosphate-buffered saline (PBS) and incubated in TrypLE at 37C for 2-5 minutes. Cells were detached using a pipette and resuspended in fresh medium and transferred to a new plate. Cell lines were routinely tested for Mycoplasma by qPCR every 3-4 months.

4.3.3 Cell transfection

Trypsinized cells were plated into 96-well at concentrations such that they were 70-90% confluent 12-18 hours after plating. Transfections were carried out using polyethylenimine (PEI) using a total DNA:PEI mass ratio of 3 for both CHO and HEK293T cells. PEI stocks were made with linear PEI MW 25,000 and pH was adjusted to 6.7 by HCl or NaOH. Stocks were filter sterilized using 0.22 μm filters and stored at -80C.

4.3.4 Fixing cells for flow cytometry

Cells were maintained for 48-hours following transfection. Cells were then washed with 1X PBS three times and then trypsinized using TrypLE for 3-10 minutes to remove adherent

cells. Cells were then suspended in 4% Formaldehyde in DPBS and incubated for 20 minutes at room temperature. Formaldehyde solution was then removed by washing cells three times by spinning cells at 300xg for 3 minutes and resuspending cell pellets in PBS. On the last wash, cell concentrations were adjusted for measurement on the flow cytometer.

4.3.5 Flow cytometry

Fluorescence intensity was measured with a BD Accuri C6 flow cytometer equipped with a CSampler plate adapter using excitation wavelengths of 488nm and an emission detection filter at 533nm (FL1 channel). A total of 10,000 to 30,000 events were recorded for each sample with and a core size of 22mm using the Accuri C6 CFlow Sampler software. Flow cytometry analysis was performed using Python and the flowcytometrytools Python package.

4.3.6 Calculation of relative units from flow cytometry data

Relative expression units (rel. u) were calculated using the following procedure. (i) Cells were automatically gated on the forward (FSC-A) and side scatter (SSC-A) channels using a 2- to 5-component Gaussian mixture model (GMM; SciPy - Python) and inspected visually for live cells. The component corresponding to live cells was selected and used to gate live cells. Cells were inspected for the presence of doublets in the FSC-A and FSC-H channels, but additional singlet gating appears unnecessary. (ii) After gating for live cells, cells are gated using a fluorescence threshold of 3 standard deviations (s.d.) above cells transfected with a non-fluorescent control pcDNA so that all cells classified as positive would contain less than 1% non-fluorescent cells. Cells above this threshold (FL1-A 7100), were classified as GFP-positive. (iii) For each positive cell population, the mean values of the fluorescent channel were calculated and multiplied by the frequency of the cell positive population. This value was used as the relative expression units (r.u.).

Software FlowCytometryTools (https://eyurtsev.github.io/FlowCytometryTools/) was used to analyze flow cytometry data. Dnaplotlib and custom scripts was used to generate SBOL

gene diagrams [99]. Aquarium was used for protocol execution [100].

DASi software (https://github.com/jvrana/DASi-DNA-Design) was used to design plasmids in this study.

Terrarium (https://github.com/jvrana/Terrarium) was used for automated experiment planning.

Sequences used in this study

4.4 Acknowledgments

I would like to thank the following for providing essential materials for this study. Jesse Zalatan for providing the pSV40-GFP construct and the corresponding gRNA sequence targeting pSV40; the Elowitz lab for sending the CHO-MIHAC cell line used in the study; the Seelig lab for providing HEK293 cell line and other materials used in the study.

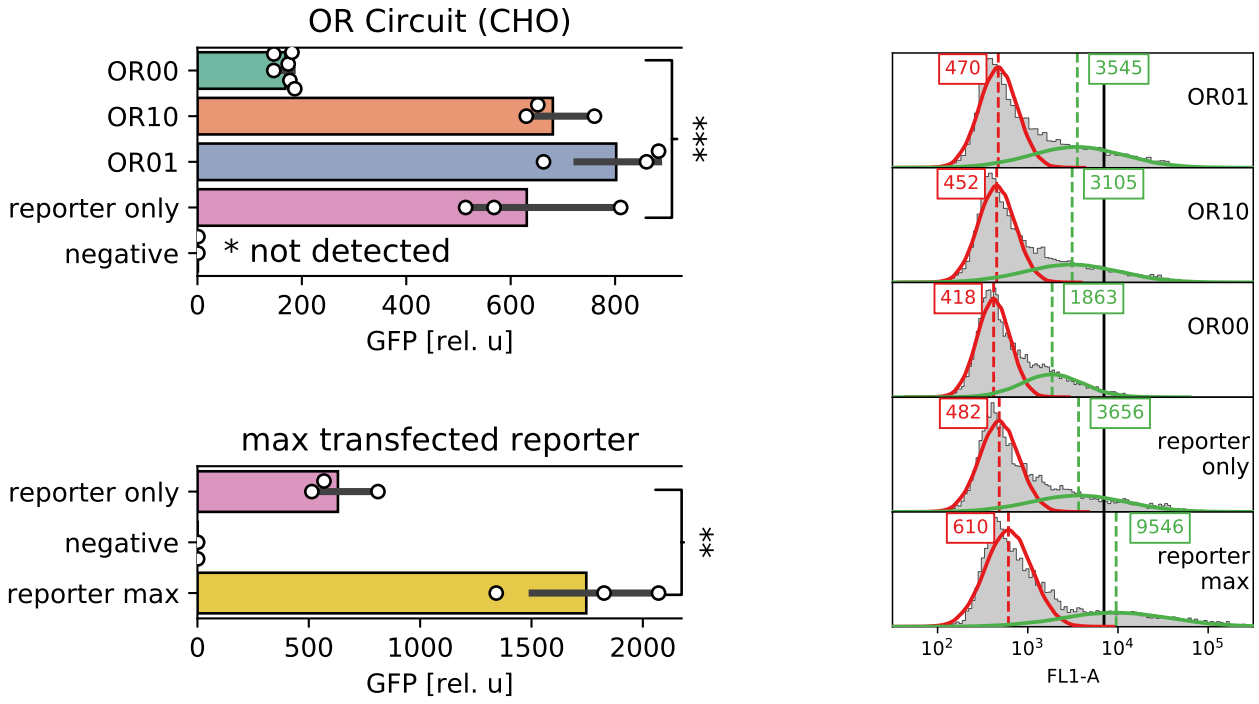
4.5 Appendix

Part	**Description**	**Sequence**
pEF1	promoter	catcgcccacagtccccgagaagttgggggggagggtcggcaatt gaaccggtgcctagagaaggtggcgcggggtaaactgggaaagtg atgtcgtgtactggctccgccttttcccgagggtgggggagaacc gtatataagtgcagtagtcgccgtgaacgttcttttcgcaacgggt ttgccgccagaacacaggtaagtgccgtgtgtggttcccgcgggcc tggcctctttacgggttatggcccttgcgtgccttgaattacttccac ctggctgcagtacgtgattcttgatcccgagcttcgggttggaagtg ggtgggagagttcgaggccttgcgcttaaggagccccttcgcctcg tgcttgagttgaggcctggcctgggcgctggggccgccgcgtgcga atctggtggcaccttcgcgcctgtctcgctgctttcgataagtctcta gccatttaaaatttttgatgacctgctgcgacgctttttttctggcaa gatagtcttgtaaatgcgggccaagatctgcacactggtatttcggt ttttggggccgcgggcggcgacggggcccgtgcgtcccagcgcaca tgttcggcgaggcggggcctgcgagcgcggccaccgagaatcgga cgggggtagtctcaagctggccggcctgctctggtgcctggcctcg cgccgccgtgtatcgccccgccctgggcggcaaggctggcccggtc ggcaccagttgcgtgagcggaaagatggccgcttcccggccctgct gcagggagctcaaaatggaggacgcggcgctcgggagagcgggc gggtgagtcacccacacaaaggaaaagggcctttccgtcctcagcc gtcgcttcatgtgactccacggagtaccgggcgccgtccaggcacc tcgattagttctcgagcttttggagtacgtcgtctttaggttggggg gaggggttttatgcgatggagtttccccacactgagtgggtggaga ctgaagttaggccagcttggcacttgatgtaattctccttggaattt gccctttttgagtttggatcttggttcattctcaagcctcagacagtg gttcaaagttttttttcttccatttcaggtgtcgtga
Csy4	riboendonuclease	ggtgatcattatctggatattcggctgaggcctgatccagagttccc acctgcgcagctgatgtctgtccttttttggcaaacttcatcaggccc tggttgcccagggcggagatcggataggggtaagctttccagacct cgacgaaagccggagccgcctgggagaacgcctgcggatccacgc ttctgccgacgatctgagagccttgctggcaaggccatggcttgag gggctccgggatcacctgcagtttggcgaacccgccgttgttcccca cccaacccttatcggcaggtgtctagagtgcaggccaaatctaat ccagaacggctgcgacggcgactcatgcggcgacatgatcttagcg aggaagaggcccgaaaaagaatccctgataccgtggcccgcgccc ttgacttgccttttgtcacactgcggtcccagagtacggggcagcat ttcagacttttcattcgacacgggccactgcaagttaccgccgaag aaggaggctttacttgttatggactctccaagggaggtttcgtgccc tggttt
SV40NLS	nuclear localization sequence	ggaccaaagaaaaaacgtaaagtg

WPRE	terminator	ctatgttgctccttttacgctatgtggatacgctgctttaatgccttt gtatcatgctattgcttcccgtatggctttcattttctcctccttgtat aaatcctggttgctgtctctttatgaggagttgtggcccgttgtcag gcaacgtggcgtggtgtgcactgtgtttgctgacgcaacccccact ggttggggcattgccaccacctgtcagctcctttccgggactttcgc tttcccctccctattgccacggcggaactcatcgccgcctgccttgc ccgctgctggacaggggctcggctgttgggcactgacaattccgtg gtgttgtcggggaagctgacgtcctttccatggctgctcgcctgtgt tgccacctggattctgcgcgggacgtccttctgctacgtcccttcgg ccctcaatccagcggaccttccttcccgcggcctgctgccggctctg cggcctcttccgcttcttcgccttcgccctcagacgagtcggatctcc ctttgggc
scaffold (F+E)	gRNA scaffold	gtttaagagctatgctggaaacagcatagcaagtttaaataaggct agtccgttatcaacttgaaaaagtggcaccgagtcggtgc
r1	gRNA target sequence	ggaacgtgattgaataactt
r5	gRNA target sequence	gaagtcagttgacagagtcg
r1-PAM	gRNA target sequence	ggaacgtgattgaataactttgg
PAM-r5	gRNA target sequence	cctcgactctgtcaactgacttc
r-SV40	NT1 gRNA target sequence, targeting the TSS of the SV40 promoter	gaatagctcagaggccgagg
Csy4 stem-loop	Csy4 stem-loop	attcctgttcactgccgtataggcagcccttt
hU6	U6 promoter	aaggtcgggcaggaagagggcctatttcccatgattccttcatattt gcatatacgatacaaggctgttagagagataattagaattaatttg actgtaaacacaaagatattagtacaaaatacgtgacgtagaaag taataatttcttgggtagtttgcagttttaaaattatgttttaaaatg gactatcatatgcttaccgtaacttgaaagtatttcgatttcttggct ttatatatcttgtggaaaggacg

tU6	U6 terminator	ttttttctag
barcode	RNA seq handle	atctcccacgtgcgctttctcccttctcc
terminator linker	cloning linker	gcttcaataaaggagcgagcacccgagtaatagtga
p1g1	cloning scar	gggaccgtcaaccctgaaccacaaa
kozak	Kozak	gccacc
start	start codon	atg
tp	cloning scar	tgataccgtcgacctcgagtcaagtaatagtg

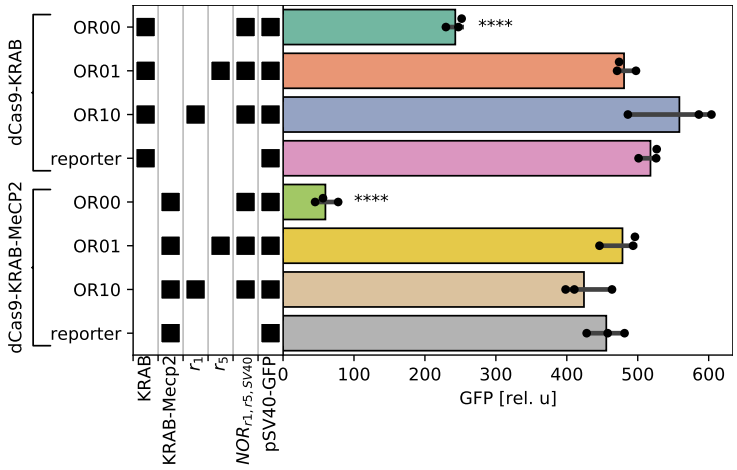
Table 4.2: DNA sequences used in mammalian NOR study



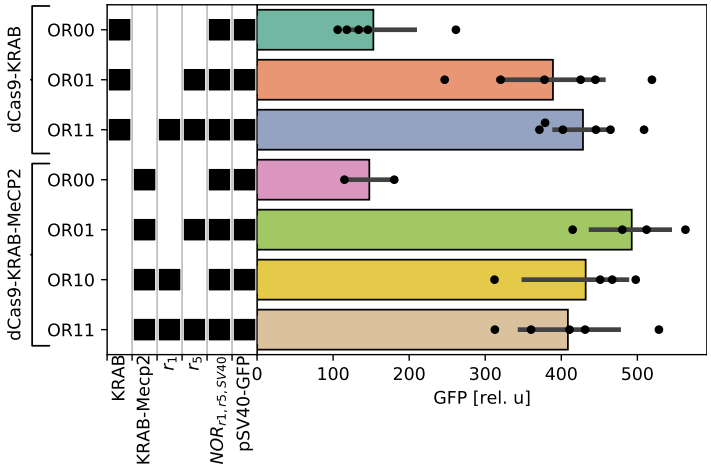
(a) OR circuit performance in CHO cells

(b) Flow cytometer plots

Figure 4.3: **OR circuit performance in CHO cells** (a) Circuit performance was analyzed by co-transfection experiments of up to 6 plasmids consisting of hU6-gRNA-r_1 (r_1), hU6-gRNA-r_5 (r_5), pSV40-GFP ("reporter"), pEF1-Csy4, pCMV-dCas9-KRAB-MeCP2, and hU6-NOR$_{r1,r5,SV40}$ ("NOR"). For each sample, a subset of plasmids was replaced with empty vector DNA. For "OR00", for example, plasmids expressing gRNAs r_1 and r_5 were replaced by empty vector pcDNA at equivalent mass. For "reporter only", only the pSV40-GFP reporter was transfected along with empty vector pcDNA. For "reporter max" the pSV40-GFP was transfected at maximum concentration. Plasmids were carefully measured at equivalent ratios and masses to minimize the effects of changes in transfection levels. (b) Representative flow cytometer plots. Highlighted are Gaussian mixture model estimates of positive (green) and negative (red) populations and their respective means. The black vertical line indicates the selected fluorescence threshold of 3 s.d. above the predicted mean of cells transfected with an empty DNA vector. This threshold was used to calculate relative units (rel. u) as explained in the methods. Repression of the pSV40-GFP reporter by gRNA, as in "OR00" results in a decrease in fluorescence which can be quantified by either calculating relative units beyond the 3 s.d. threshold or by computing the fraction of cells. ** indicates p-value < 0.01 and *** indicates a p-value < 0.001 as calculated by Student's t-test.



(a) OR circuit performance in CHO cells



(b) OR circuit performance in HEK293 cells

Figure 4.4: **OR circuit performance in CHO cell line** dCa9-KRAB vs dCas9-KRAB-MeCP2 OR circuit performance was measured in Chinese Hamster Ovarian (CHO) and Human Embryonic Kidney (HEK293) by co-transfection. Samples were transfected with a subset of seven plasmids, displayed on the x-axis of the left side (Csy4 not displayed, but eas transfected in all samples). Components included in each sample are indicated by a black square on the left; all samples included transfection of pEF1-Csy4 which is not displayed on the left. When a component was not included in the co-transfection, the component was replaced with an equivalent mass of empty vector (pcDNA) to minimize differences in transfection. In HEK239 cells, cytometer instrument error causes loss of the "dCas9-KRAB OR10" samples. ******** indicates p-value < 0.001 calculated by the Student's t test

Plasmid	**Source**
dCas9-KRAB	Addgene #110820
dCas9-KRAB-MeCP2	Addgene #110821
pcDNA3	
pSV40-GFP	Zalatan lab
$\text{NOR}_{r1,r5,r_{\text{SV40}}}$	hU6,r1-PAM,PAM-r5,Csy4 stem-loop,r-SV40,scaffold (F+E),Csy4 stem-loop,barcode,tU6
hU6-r_1	hU6,r1,scaffold (F+E),Csy4 stem-loop,barcode,tU6
hU6-r_5	hU6,r5,scaffold (F+E),Csy4 stem-loop,barcode,tU6
pEF1-SV40NLS-hCsy4	pEF1, kozak, start, SV40NLS, hCsy4, tp, WPRE

Table 4.3: DNA constructs used in mammalian NOR study

Chapter 5

SEMI-AUTOMATED EXECUTION OF LABORATORY WORKFLOWS

Author	Contribution
Eric Klavins	Core Aquarium system development (version 1, 2)
Yaou Yang	Core Aquarium system development (version 1)
Justin Vrana	Python API to Aquarium, protocol development and coding, protocol testing
Ben Keller	Core Aquarium system development (version 2)
Devin Stickland	protocol development and coding, protocol testing
Many others*	protocol development, protocol testing, execution

5.1 Abstract

Automation has been shown to improve the replicability and scalability of biomedical and bioindustrial research. Although the work performed in many labs is repetitive and can be standardized, few academic labs can afford the time and money required to automate their workflows with robotics. We propose that human-in-the-loop automation can fill this critical gap. To this end, we present Aquarium, an open-source, web-based software application that integrates experimental design, inventory management, protocol execution, and data capture. We provide a high-level view of how researchers can install Aquarium and use it in their own labs. We discuss the impacts of Aquarium on working practices, use in biofoundries, and opportunities it affords for collaboration and education in life science laboratory research and manufacture.

5.2 Introduction

As the scale of scientific research expands, systems to support replicable methods are increasingly important [10]. Critical to replicability is the question of how experiments are described and how closely these descriptions are followed. Working practices for conducting biology experiments fall on a continuum between artisanal and highly standardized. On one end, idiosyncratic decisions made on the fly by a few people appear throughout the experimental design. At worst, these decisions are poorly documented, while at best they introduce extra experimental factors that make results challenging to compare and replicate. On the other end, researchers follow well-established protocols and do not deviate from them. This is due to either top-down enforcement, as with clinical research, or to the fact that doing so is simpler and more reliable, as with kits for common procedures such as plasmid DNA isolation. Similarly, record-keeping varies from hand-written laboratory notebooks and manually-curated digital records to fully structured electronic notebooks.

A consequence of less structured approaches is that many experiments cannot be repeated by different researchers [101, 102]. These types of issues are typically described under the

terms replicability (reproducibility of results) [103–106]. While replicability in science is a complex and multifaceted issue [107, 108], enforcement of standardization of experimental work can result in more replicable data collection [109, 110]. However, the reality is that maintaining this standardization and record-keeping is laborious and researchers often fail to enforce strict standards on themselves at the bench [111].

Robotic automation has been proposed as a solution for replicable large-scale experimentation by delivering consistent performance of delicate high-throughput procedures [112]. However, commercially available liquid handling robots and general-purpose manipulators have high upfront costs and are laborious to reconfigure or reprogram [11]. While robotic systems are continually improving, labs solely reliant on robots to carry out experimental work are rare. Many experiments involve tasks for which a human operator is well suited and more cost-effective than currently available robots; for example, tasks involving delicate hand-eye coordination or variable inputs and protocols. Hence, it seems likely that human researchers will be involved in benchtop experimentation for some time, and thus the potential for non-standardized execution and sub-optimal record keeping will remain a concern.

To address this challenge, we built Aquarium—a web-based application that integrates experimental design, inventory management, protocol execution, and data collection. Aquarium supports flexible development and deployment of standardized workflows, composed of modular protocols, that drive on-screen, step-by-step instructions for human technicians. During execution, experimental data and metadata are captured in forms or uploaded as files. The software automates computations involved in preparing and tracking samples through protocol execution. Aquarium also provides features to plan complex experiments involving many samples and protocols.

Aquarium integrates two key software innovations: Aquarium workflow language (AWL) for defining custom laboratory workflows and Krill, a protocol language for describing replicable laboratory instructions. AWL is a dataflow programming language [113] that represents a laboratory workflow as a network of modular work units linked by inputs and outputs.

Borrowing concepts from visual programming languages, such as Scratch (Resnick et. Al. 2009), protocols are represented graphically as blocks that can be wired together to create workflows. Krill is a Ruby domain-specific language (DSL) that complements AWL by capturing granular instructions for a protocol as computer code. In addition to complex procedural steps such as if-then statements, loops, and calculations, Krill has methods to facilitate sample flow management and render instructions for technicians working at the bench. Through AWL and Krill, Aquarium provides interactive web-based interfaces to build executable protocols, design experimental workflows based on these protocols, manage the execution of protocols in the lab, and automatically record the resulting data.

Aquarium also features a Python application programming interface (API), called Trident, that provides a common interface for other applications and scripts to interact with Aquarium, for example in planning complex workflows or extracting detailed datasets. These three programmatic interfaces, combined with inventory management and human-centered execution, make Aquarium a comprehensive, open-source software platform that facilitates low-cost scaling of laboratory research while retaining replicability and flexibility.

5.3 Results

5.3.1 Planning laboratory work with Aquarium

From the perspective of the researcher, planning laboratory work is the primary interaction with Aquarium. Researchers design **plan**s using a graphical user interface (GUI) that resembles a sketch board (Fig. 1). Plans are built from workflows, essentially stereotyped series of procedures, in which materials pass from an initial state to a final state. Within plans each input sample passes through a series of work modules termed **operation**s (Fig. 5.2b), to produce desired output samples and data. For example, a plan that ends with a sequence-verified plasmid stock (Fig. 1) might include a series of operations such as PCR, DNA assembly, bacterial transformation, and plasmid DNA purification. A plan may represent a fixed workflow that always executes in the same way, or it may be extended as it

is executed. Thus enabling the use of Aquarium for either manufacturing or exploratory research and development.

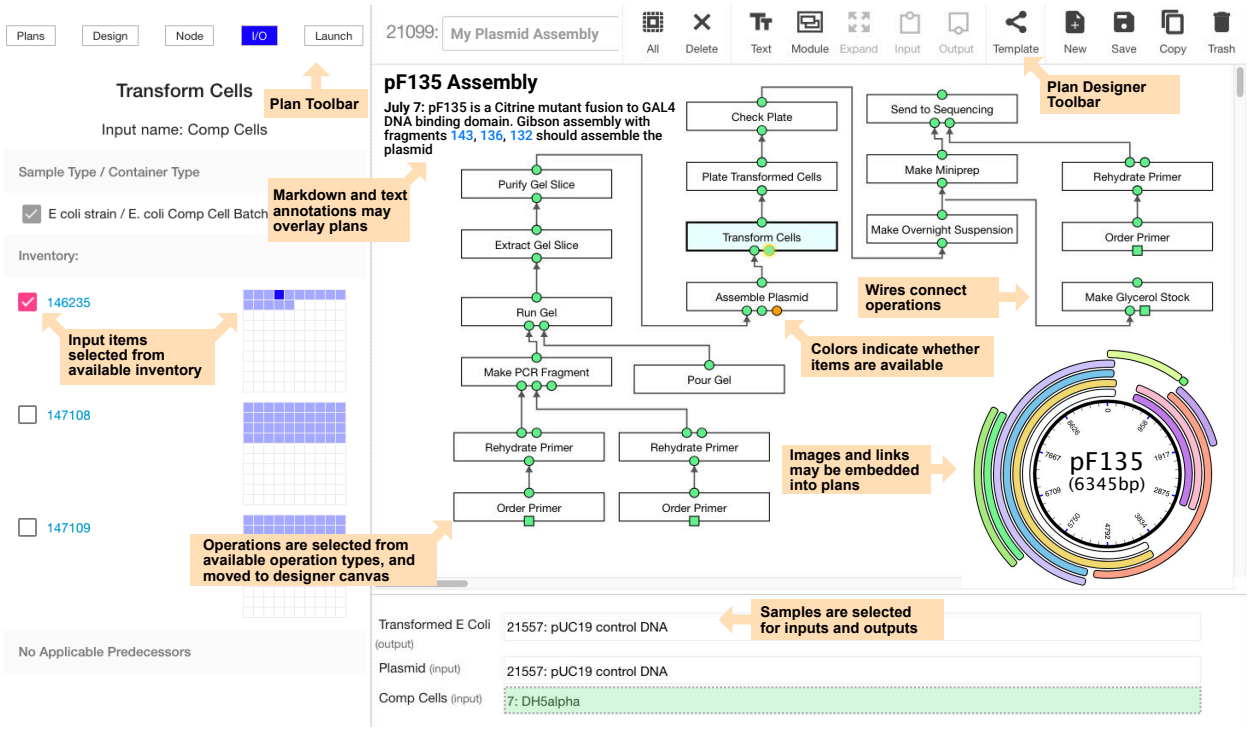


Figure 5.1: **Aquarium planner interface** Experimental plans can be created by dragging operation types (not shown) onto the designer canvas (right side) to create operations. Selecting a given operation input or output node users are prompted with the select from a list of compatible up or downstream operations to create a custom workflow. Available inventory for each operation input is selected in the input view (bottom) and the I/O view (left). Designer tools are available for creating templates and modifying/copying plans. Additionally, there are several plan tools (top left) available for investigating input/output specifications, managing existing plans, and launching plans. The designer also features annotation capabilities, allowing embedded text (such as Markdown; https://daringfireball.net/projects/markdown/), images, or links.

Operations correspond to units of work that can be performed on one sample, by one person, within a single work session. Each operation will generally output a sample that can be stored or used in a variety of other operations. Operations are wired together such that the output of one operation is automatically routed to and triggers the execution of

one or more subsequent operations (Fig. 5.2b). Each operation is defined by valid inputs and outputs as well as a detailed laboratory protocol, written in Krill, that renders on-screen instructions to guide technicians. An Aquarium plan can encode arbitrarily large and complex programs of work that progress automatically. As the plan is executed, the state of inventory is automatically updated and data is captured, stored, and made available through the GUI as well as the Python API.

5.3.2 Executing laboratory work

Aquarium contains its own laboratory information management system (LIMS) that tracks lab inventory. Through the LIMS, changes in inventory are recorded automatically as a part of protocol and workflow execution, rather than requiring manual updates. Hence the workflow planner and Krill can reliably use the LIMS as an up-to-date representation of the laboratory.

In Aquarium, a physical object is referred to as an **item**. Each item has a recorded location, and may have associated data (Fig. 5.3). An item is an instance of a **sample**, which is a class of physical objects in the laboratory defined by a set of descriptors determined by a **sample type**. The information fields used to define each sample type are chosen by the user and then apply to every sample of that sample type. For example, a user may define a 'plasmid' sample type, including fields for information on sequence, length and selectable markers, which would have to be defined for each plasmid sample in the database. Using sample types ensures that inventory descriptions are standardized, while allowing the flexibility of custom definitions based on user needs.

Each item belongs to a user-defined **object type**. One key parameter for the definition of an object type is the default **location wizard** for items of this object type. Location wizards correspond to storage locations such as fridges and freezers and are represented as matrices with unique numbered positions for items within boxes, organized into rows and shelves (Fig. 5.3b). Other than the location wizard, the name of each object type is it's most important defined parameter and will indicate the physical state of the sample, reflecting how it manifests or is used in the laboratory. For example, a plasmid sample might have items associate with it belonging to a number of object types such as 'Plasmid miniprep', 'Gibson assembly reaction product', or '*E. coli* overnight culture'. As items are generated in the course of laboratory work, they are automatically assigned ID numbers as well as locations according to the location wizard defined by the item's object type.

Once the items required to initiate an operation in a plan are available in the lab, the

inputs to the operation are satisfied and the operation can be executed. In the manager subsystem, an executable operation is batched into a **job** with operations of the same type to be executed together (Fig. 5.2). A job may include operations from many different plans and researchers, but can only be created from operations of the same type. A strict execution policy governs how and when operations can be batched into jobs and executed in the lab (Fig. 5.5). Running a job launches a graphical user interface that displays step-by-step instructions to guide a technician through the steps of the operation protocol (Fig. 5.4).

5.3.3 Rendering instructions using the Krill protocol language

The Krill protocol language produces detailed, context-specific instructions for how materials and data should be handled for each operation (Fig. 5.4). To accomplish this, Krill provides methods that allow arbitrary computations to be rendered dynamically, so that specific instructions presented to the technician can be made to reflect not only the number of items being processed, their locations and ID numbers, but also the results of calculations such as pipetting volumes based on the molarity of a solution. Thus, a Krill protocol describes a procedure in enough detail that it can be replicated by another person or lab by following specific instructions on a tablet or computer screen. Krill methods include those to execute complex calculations, retrieve and generate data, add and remove inventory, display videos and photos, retrieve user input, create interactive timers, output audio alarms, and send emails, among other functions. Krill extends Ruby [114], a popular, dynamic, object-oriented language used in web development. To facilitate development, Krill provides ways of creating libraries of reusable code. A version control system allows a lab to record changes to their protocols over time, or revert their protocols to past versions.

The Krill protocol is supplemented by two functions that facilitate the proper execution of the protocol (Fig. 5.4). The cost model calculates how much an operation may cost at the time of execution. Cost models can use any information from Aquarium to perform cost calculations, but typically calculations use properties of samples or items used in an operation. For example, an operation that orders a synthesized piece of DNA, may use the

length and sequence of the DNA and check with a vendor website to establish an accurate monetary cost. Some protocols may be labor-intensive, and so operation cost models can include an estimation of the 'labor rate' and the average length of time required to complete the protocol. Additional features in Aquarium allow the generation of monthly spending reports and budget tracking for each user or user group. The precondition defines conditions required for a protocol to be run; the default precondition is always true. Preconditions are a critical part of an operation's execution policy (Fig. 5.5) and can be used to institute more complex experimental workflows. For instance, enforcing a 12-hour delay to wait for an *E. coli* plate to grow. Like the cost model, precondition code may use any of the other subsystems to establish running conditions. For example, an operation that runs a colony PCR on a bacterial plate may halt operation if there is data associated with the plate indicating there was contamination, or if the plate is less than 12 hours old and thus not ready to be run. Finally, the documentation contains human-readable markdown text that describes how the protocols are used and executed.

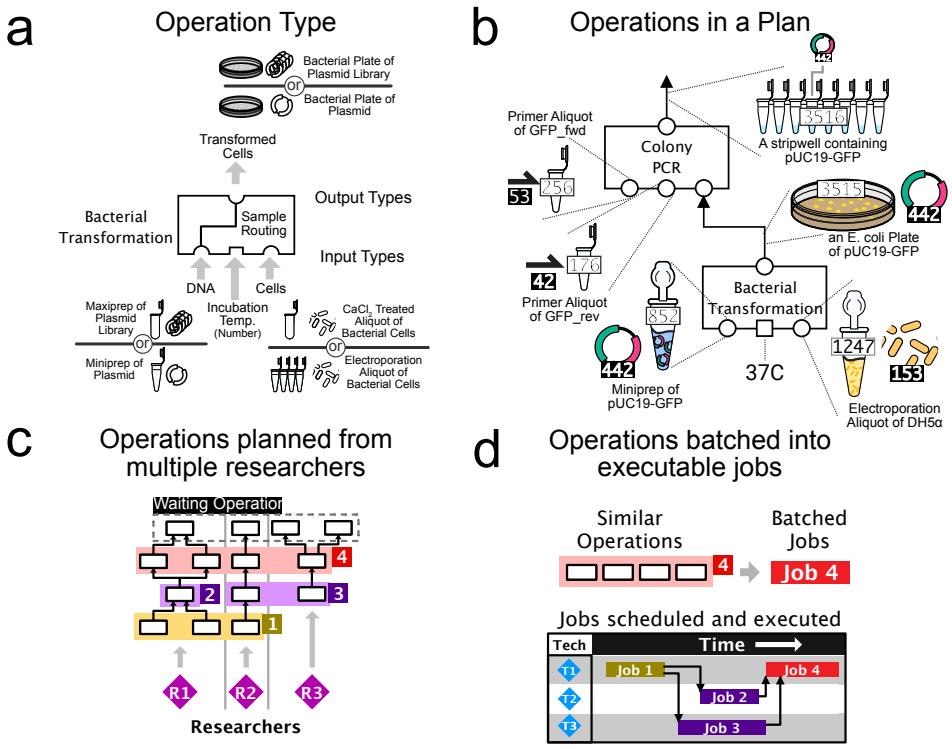


Figure 5.2: **Depictions of operation, operation type, plan, and job models that comprise an Aquarium Workflow Language** (a) An example of a "Bacterial Transformation" operation type is displayed. Operation types define specific ways in which input samples and items can be processed to produce outputs. Each operation type contains specifications for its input and output types. For example, the "DNA" input of a hypothetical bacterial transformation operation type may be satisfied by a 'maxiprep of plasmid library' or a 'miniprep of plasmid'. Input and output types are entirely customizable and may include any number of sample type and object type specifications. Sample routing, if provided, ensures the input and output samples are mapped correctly upon operation execution; here the input "DNA" sample will be mapped to the "Transformed Cells" output sample. Non-inventory inputs (i.e. parameters) can also be defined as inputs to operation types. (b) An example of a bacterial transformation connected to a colony PCR operation. Operations types are instantiated to operations when their input and output types are satisfied by items. Here a bacterial transformation uses specific items in the LIMS and a parameter (37°C) to produce a bacterial plate. After executing the transformation, the output plate is wired to the colony PCR operation. The colony PCR outputs the amplicon to an empty well in a stripwell. Notice that the operation type sample routing ensures sample information (here pUC19-GFP, sample 442) is maintained throughout the series of tasks. (c) Operations from several different researchers and different plans can be batched together into jobs if they have the same operation types. For instance all 'Colony PCR' operations from all users can be run as a single job. (d) Once operations have been batched into jobs, jobs can be run divorced from Aquarium plans because all necessary information for execution is included in the job. These jobs can then be performed concurrently by separate technicians.

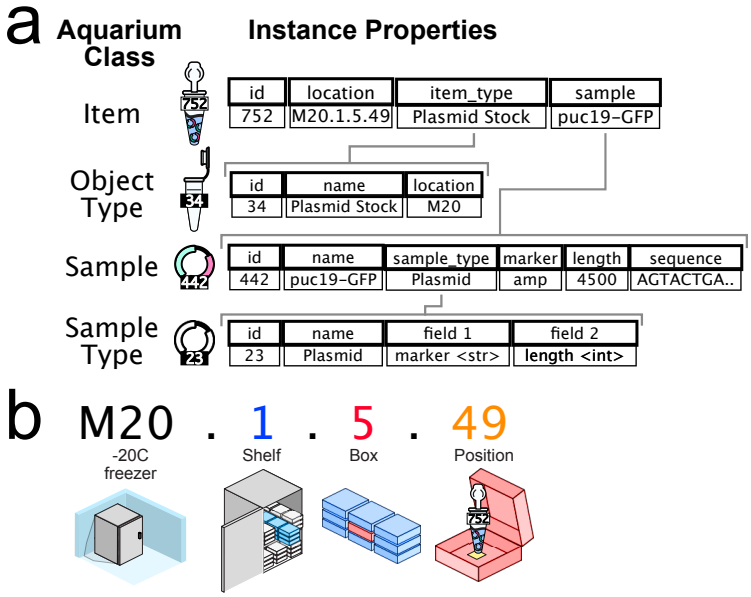


Figure 5.3: **Aquarium inventory models** The procedures of a given research lab will require handling of multiple different sample types, representing things like DNA plasmid, various cell lines or chemical reagents. (a) For instance samples of type "Plasmid" may be defined by a marker, length or sequence. Properties for sample types are defined by the user allowing for custom definitions of inventory. In Aquarium, there are no hardcoded concepts of "Plasmids" or any other form of laboratory inventory. Once a sample type has been defined samples can then be added to the database. Items are physical manifestations of samples and always have a location, as well as the ability to carry data associations. Each item is of a given type, known as an object type defining relevant physical properties relevant to laboratory handling. (b) Each item produced automatically gets assigned a location. How this location is assigned is reconfigurable in Aquarium. Here, a -20C freezer is designated "M20" and has three dimensions (shelf, box, and position). The location designation and capacity along the dimensions are customizable. Which types of items go into which locations is defined in the item's object type. Proper management of item locations in Aquarium is critical as this feeds into protocol execution, which may use (or alter) item location during execution.

5.3.4 Operation execution policy

An operation has a *status*, which describes its current state ('running', 'done', 'error', etc.). The statuses of operations are governed by a state diagram which ensures strict execution of operations in an experimental plan (Fig. 5.5). All operations begin in the 'planning' state and eventually end up in the 'done' state if the operation completed with no errors, or the 'error' state if the associated job was unable to be completed (e.g., because a piece of equipment broke down). Upon moving to the 'done' state, operations often will produce their output items, allowing their next operations (i.e. successors) to eventually be scheduled. Upon launching a plan, operations move from 'planning' to 'waiting', or to 'pending' if the operation has no predecessors (known as a 'leaf'). Some operations have a special designation known as 'on-the-fly, and are only scheduled when their successors are batched. While rarely used, 'on-the-fly' operations are useful for performing 'look-ahead' actions when operations highly dependent on the specifics of their successors' batching. As an example, there may be a "Pour Agarose Gel" operation that depends on its successor operations "Run Gel"; how many gels are poured depends on how the "Run Gel" operations are batched, hence a look-ahead is required and the "Pour Agarose Gel" is designated as "on-the-fly." When items are made available for an operation's inputs, it moves from 'waiting' to 'pending.' From 'pending,' operations of the same type can move to 'scheduled' when they are batched together, and move to 'running' when the batch job is executed. In some cases, an operations precondition code will fail, which will send operations to the 'delayed' state. Once the precondition passes, the operation can return to 'pending' and be scheduled, run and completed. Precondition code is run for waiting operations before the operation transitions to pending (being 'stepped'), and enables delays or inventory checks to be done before operations are scheduled.

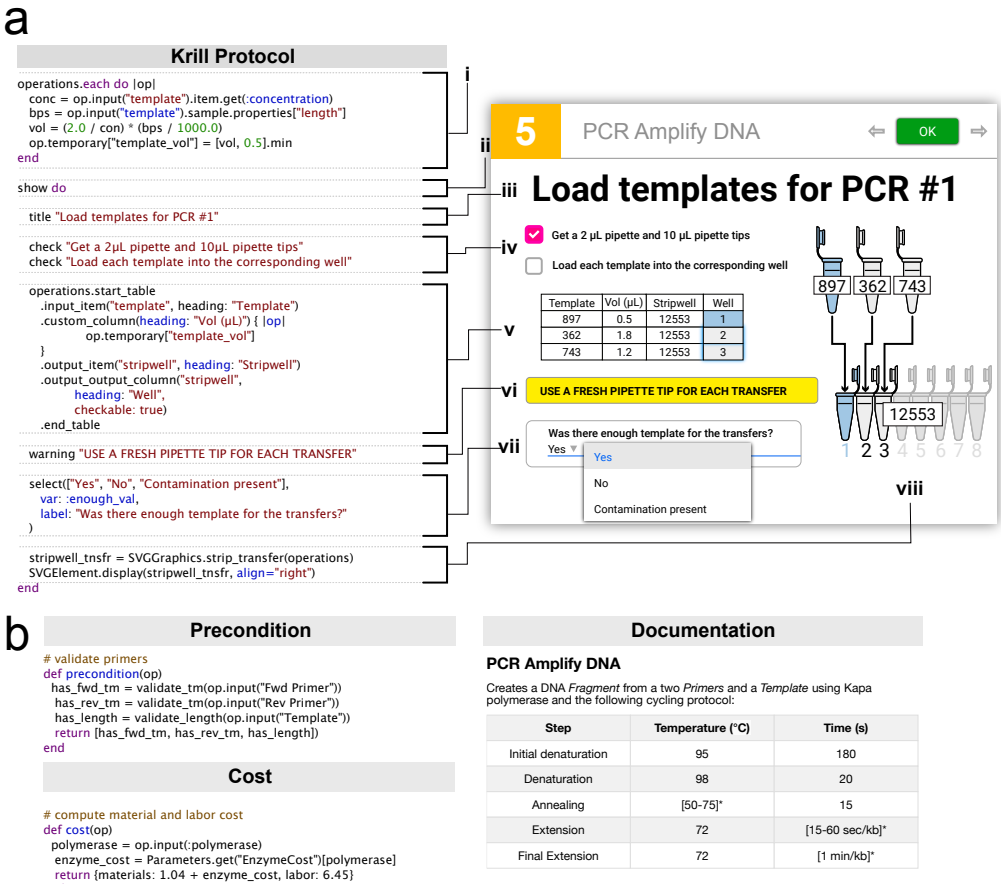


Figure 5.4: **Krill** An example of Krill protocol code and its corresponding rendered protocol instructions. Operation types have four sets of code that govern operation behavior and scheduling: Krill protocol, precondition, documentation, and cost. (a) Shown here is a snippet of Krill protocol code for the load template step in a PCR amplification protocol with lettering highlighting aspects of the code. Operations are batched into a single job; when the job is executed, Krill code can access the input and output information of all batched operations. (i) In this simple example, the template volume is calculated for each operation before generating instructions for loading template DNA into wells. (ii) A new protocol step is rendered with a show block. (i - vii) Within the show block, various elements are rendered for the technician. (iii, iv) A title is displayed and has two checkboxes. Checkboxes must be checked before proceeding to the next step, forcing the user to be attentive. Operations are iterated to display a table that uses the computed template volume, inputs, and outputs of the operations. (v) Tables can be interactive and may include text inputs or checkable boxes (vi) A visible warning is displayed. (vii) A selection input instructs the user to select from a list of options; numerical, textual, and file upload inputs are also possible through Krill. (viii) Finally, an SVG graphics element can be rendered on the fly using operation information. (b) Optional precondition code governs when operations can be scheduled into jobs. Cost model computes monetary costs prior to plan launch and documentation provides readable instruction about the underlying protocol.

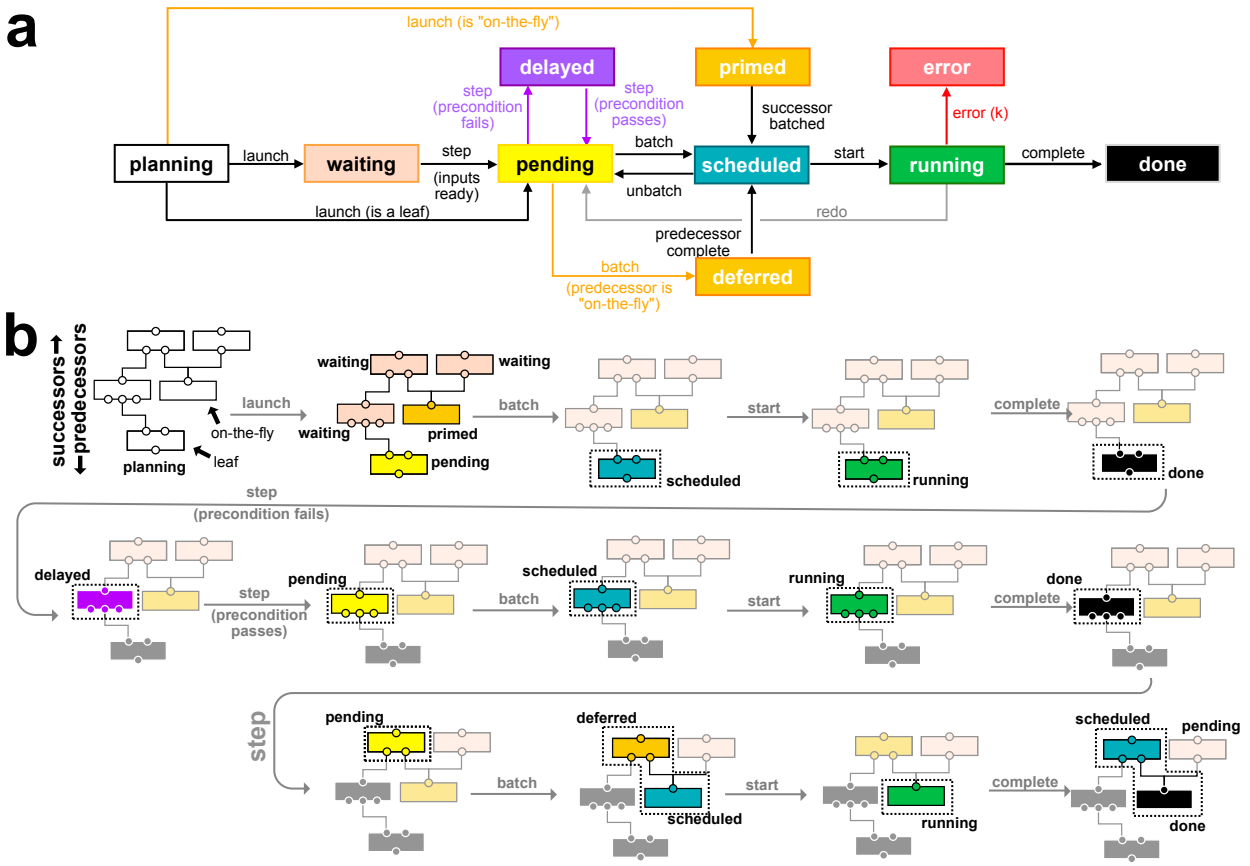


Figure 5.5: **Operation execution policy** (a) transition diagram for operation state. (b) Shown in the example, is an execution procedure for a generic plan containing five operations, with one 'on-the-fly' operation. Boxes indicate operations and colors indicate statuses. Gray arrows indicate actions such as 'batching', 'starting', 'completing' or 'stepping'. Operations whose statuses have changed from the previous action are highlighted with a dotted box.

5.3.5 Recording and accessing experimental data

Aquarium records data in several ways. By default, Aquarium automatically logs protocol metadata during job execution. These metadata include operations batched within the job as well as the identity of submitting users and IDs of source plans for each operation. Job logs also capture technician identity, job start and end time, timestamps for each step in a protocol, job error records, and inventory handled. In addition to the job metadata, Aquarium has generic data associations that record data as attachments to inventory items, operations, or plans. Data associations may include numerical (e.g., DNA concentration), text (e.g., experiment notes), or file uploads (e.g., sequencing results). Data associations can be created manually through the GUI or automatically by a protocol during a job. During execution, protocols may include specific steps instructing the technician to record or upload data (Fig. 5.4 -vii). Post-execution, data can be accessed by researchers through the GUI, or by scripting via Aquarium's Python API, known as Trident. Trident allows custom Python applications that power visualizations, interfaces, reports, or machine-learning workflows to communicate easily with Aquarium.

5.3.6 Interacting with Aquarium

There are five major interfaces in Aquarium: designer, plans, manager, samples and developer. The designer tab provides access to the AWL interface, in which users can draft and launch plans, selecting from available operation types. The plans interface offers a summary of launched plans including up to date sample and status data. The manager tab is used to batch operations into jobs and run them. Within the manager interface all operations are accessible, grouped by category, operation type and status. Launching a job from the manager interface starts the on-screen instructions used by technicians (Fig. 5.4). The samples interface provides searchable access to inventory. The developer tab provides access to an integrated developer environment (IDE) where Krill code for protocols can be written and tested directly in the web browser (Fig. 5.6).

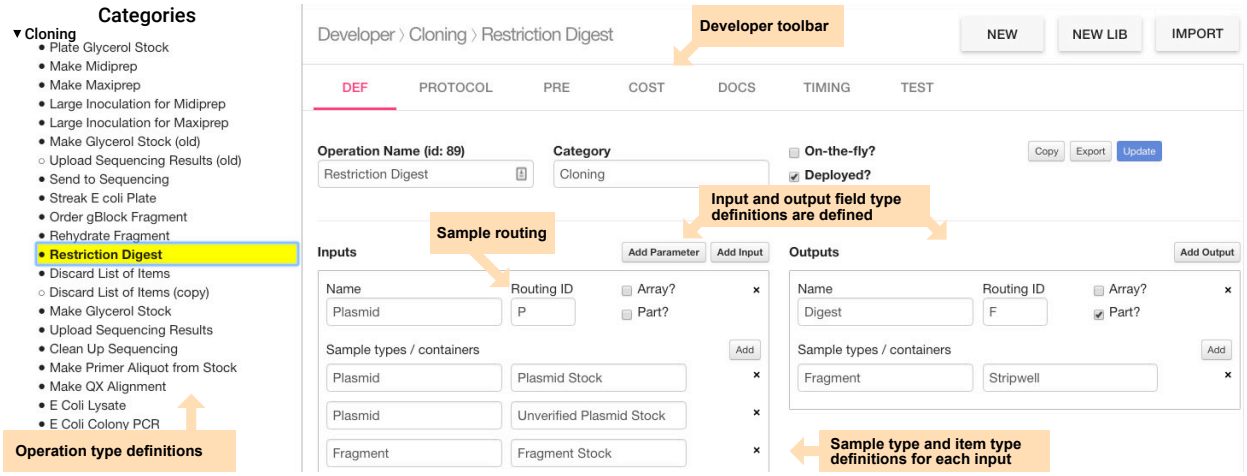


Figure 5.6: Aquarium integrated developer environment

New operation types are developed through the operation type IDE. Input/output specifications are defined for each operation, along with sample type and object types specifications for each input or output. Several tools are available (top) for editing Krill protocol, precondition, cost, and documentation code. A built-in protocol testing environment is available (top right) to speed workflow development.

5.4 Discussion

5.4.1 Specialization of roles

Aquarium facilitates, but does not require, a division of personnel roles roughly corresponding to the different front-end interfaces, thereby facilitating the standardization of laboratory workflows as a low-cost and flexible alternative to robotic automation systems. The module composition approach implemented by AWL is intended to allow for flexible workflow design, reflecting the reality of discovery-phase research, while gaining the benefits of standardization, including replicability, and efficiency gains from batched jobs. Laboratory roles can further be divided into lab managers, scheduling and assigning jobs, and technicians, executing jobs. The following role descriptions are based on our experience using the system, while recognizing that individual members of laboratory personnel have often adopted multiple or blended roles.

Researchers, including graduate students and postdocs, use Aquarium's LIMS to define new samples and the Aquarium workflow language Aquarium Workflow Language (AWL; Fig. 1) to design and launch plans. Once a researcher is ready to launch a plan, costs are computed with the operation cost models, and the researcher assigns the total to a budget that Aquarium uses to automatically track spending and generate reports. After launching, plans become visible within the plans interface, where the researcher can see the status of each operation in the plan, as well as access collected data. Some power users in the researcher role entirely bypass the Aquarium browser front-end and instead add sample definitions, submit plans and retrieve data through the Trident API. We have found that these tools allow researchers to spend minimal time at the lab bench, and more time reading literature, planning experiments, and analyzing data.

Developers use Aquarium's IDE to specify operation types and associate code (Fig. 5.4). In our experience Aquarium protocol drafting typically begins with a pre-existing paper-based or digital protocol as references, with the developer often working with an experimentalist for guidance. Once a protocol has been tested in the IDE, it can be deployed, making it available to add to plans and run in jobs. Developers also work with researchers and managers to develop the cost model, documentation, and preconditions to create cohesive workflows (Fig. 5.4d).

Managers batch operations and schedule and launch jobs, in the process deciding how many operations to include in each job, when each job should be executed and which technician should run the job.

Technicians execute jobs at the lab bench in accordance with the on-screen instructions provided by the protocol code (Fig. 5.4). Instructions typically include item retrieval and storage, sample preparation and handling, operation of laboratory instruments, and data uploading. Technicians may be guided to directly upload data files from cameras or other equipment, or asked to create data based on prompts (e.g., answering whether or not a band of a given length is present on a gel).

As well formalizing personnel roles, Aquarium facilitates a conceptual shift to thinking

about all laboratory work, including both manufacturing and experimentation, as composed of modular units, with the steps of each modular unit standardized. Standardization of laboratory methods can be beneficial for replicability 10,11 and Aquarium provides a means to ensure that standardized procedures are both established and followed. This arrangement can also reduce experimental bias and shield sensitive sample information.

5.4.2 Aquarium use cases

Aquarium has been used for a number of applications beyond the work of academic research groups. These include biofoundries, service laboratories and laboratory skills training.

Biofoundaries are facilities providing laboratory services to the synthetic biology research community, generally including plasmid assembly and strain construction 1. Aquarium's built-in abstraction barrier between design and execution, and system for efficient task management are well suited to support biofoundries. Aquarium has supported a biofoundry at the University of Washington, the UW BIOFAB, that was first developed for internal use in 2014 and then made publically accessible in 2016. Between 2014 and 2020, the UW BIOFAB has run over 30,000 jobs, serving 319 different users. BIOFAB technicians have assembled 23 million base pairs of DNA using 8.8 million base pairs of fragment DNA amplified in-house, and have built over 5,700 different yeast strains. This work has supported synthetic biology research efforts of the Klavins lab and collaborators [31, 115–118], as well as other users with no shared research interests. The UW BIOFAB first implemented cloning and yeast construction services, but has since moved on to offer plant cultivation and transformation, mammalian cell culturing, protein engineering, and next-generation sequencing, and other workflows.

Operated by private companies or public institutions, service laboratories support clinical diagnostics, agricultural soil and crop analytics, and forensics. The impacts of the global pandemics (such as COVID-19) highlighted the importance of low-cost, flexible tools that can support the rapid scaling of laboratory services both in terms of throughput and geographical reach. Aquarium was recently used to support an HIV-resistance screening workflow for use

in the developing world [119, 120], taking advantage of Aquarium's graphical technician interface, data collection management and options for rapid deployment into new devices and locations.

Given the instructional efficacy of the technician interface, we have also found utility for using Aquarium as an education tool, teaching university laboratory courses with the software. Aquarium's technician interface (Fig. 5.4) delivers step-by-step instructions at the lab bench, reducing the need to front-load learning of methods, similar to Just-in-Time Teaching [121]. It has also been used to support undergraduate laboratory training courses at the University of Washington and elsewhere.

5.4.3 Comparison with other software

Aquarium is part of a growing ecosystem of laboratory research software that we believe will become central to the working practices of researchers over the next decade. Similar to Aquarium, existing software platforms like those provided by Benchling, Riffyn, Teselagen, and Transcriptic have fully-featured LIMS capabilities that connect to protocols and workflows in meaningful ways. However, as far as we are aware, Aquarium is unique in its support for human-centric workflow execution which allows labs to leverage existing equipment and personnel. This is in contrast to robotic automation approaches, like those provided by Transcriptic, that integrate workflow execution primarily via laboratory robotics, which allows high-throughput experimental automation. Other platforms specialize in other aspects of the laboratory research process, such as Riffyn, which provides sophisticated tools for connecting and integrating workflow processing to data analytics; Benchling and Teselagen integrate aspects of biodesign to LIMS and workflow processes. Unlike these tools, Aquarium has limited capabilities for data processing and biodesign. Instead, Aquarium focuses on flexible workflow planning and execution, leaving design and data analytics to other software better suited to that task, such as those mentioned above. Aquarium's open-source Python API and flexible LIMS invites future integrations of Aquarium with other software systems.

5.4.4 Future development of Aquarium

We support a growing Aquarium user community and are aware of at least eight groups that have set up and operated independent Aquarium servers for applications ranging from plant transgenics, to microbial strain construction to biomedical diagnostics. Aquarium is distributed under the MIT license to promote adoption by, and contributions from, users in any setting, whether academic, commercial or educational.

An online hub (https://www.aquarium.bio/) for sharing and peer-curation of Aquarium workflows supports the growing user community. Aquarium workflows currently can be exported and published as Github repositories. Current development plans include simplification of the Krill protocol language to lower barriers and allow for wider use of Aquarium in life science research labs.

While Aquarium provides a way to formalize scientific workflows and their execution so as to allow researchers without high-level knowledge of the protocol to perform experiments reliably, it does not provide guidance on experimental design choices. However, there have been many recent advances on computer-aided design (CAD) tools for science [122–127]. Using Aquarium and its Python API provides a way to execute experimental plans developed by CAD software and return results in a machine-readable format. In the future, one can imagine combinations of such systems that mediate automatic design and submission of experiments, execution through Aquarium, automated extraction and analysis of results, and rapid redesign.

5.5 Materials and Methods

5.5.1 Software availability

Aquarium is distributed under the open-source MIT license. Aquarium, documentation, and installation instructions are freely available (https://www.aquarium.bio/) along with links to Dockerized versions of the software. Code is maintained on Github (https://github.com/aquariumbio/aquarium). Aquarium's Python API (Trident) is also under the open-

source MIT license and is hosted on the open-source python repository at PyPI (https://pypi.org/project/pydent/) and its documentation and installation instructions are also freely available (https://aquariumbio.github.io/trident/).

5.5.2 Software implementation

Aquarium is implemented as a browser-based Ruby-on-Rails application (https://rubyonrails.org/), with an AngularJS (https://angularjs.org/) and HTML5 front-end.The current implementation of Krill leverages Ruby (https://www.ruby-lang.org/en/), which is a popular, dynamic, object-oriented language used in web-development. Data models are implemented using a MySQL relational database (https://dev.mysql.com/). Aquarium is distributed as a Docker (https://www.docker.com) image (https://hub.docker.com/repository/docker/aquariumbio/aquarium), along with Docker Compose (https://docs.docker.com/compose/) scripts (https://github.com/aquariumbio/aquarium-deployment) that can be used to orchestrate backend, relational database, and frontend services. The Trident API is implemented in Python using open-source libraries, and is available as a Python package via pypi.org (https://pypi.org/project/pydent/).

5.5.3 Funding

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), the Department of Defense, or the United States Government.

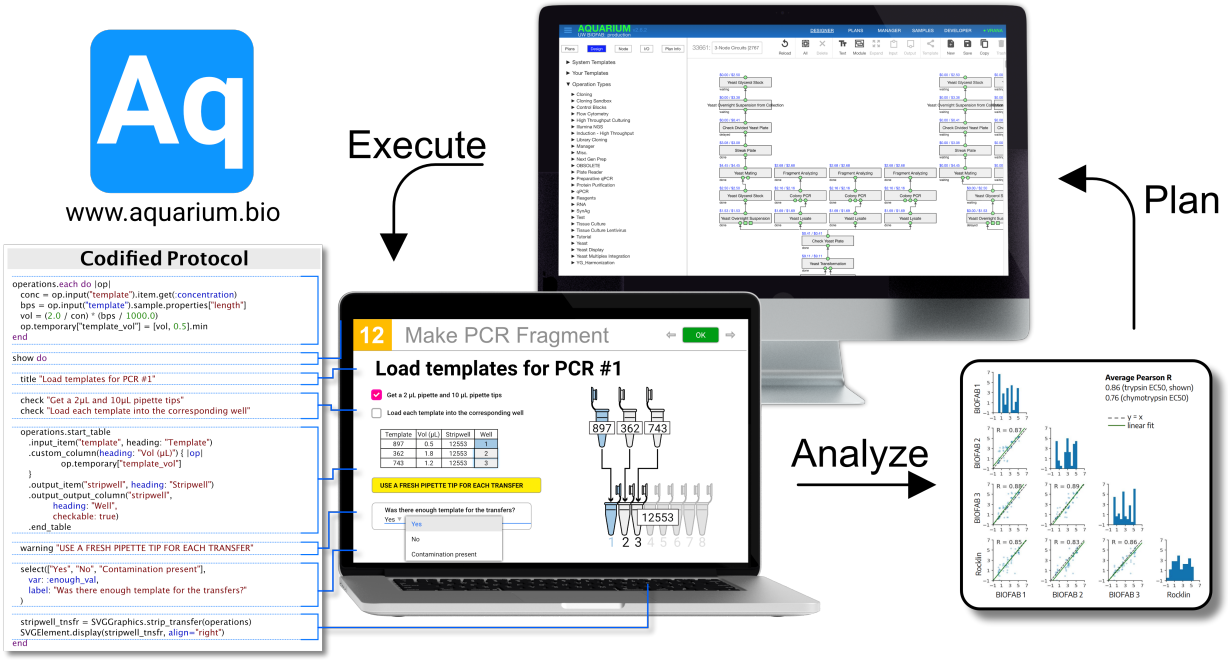
5.5.4 Contributions

Justin Vrana, Orlando de Lange, Devin Strickland, and Ben Keller wrote and revised the manuscript. Eric Klavins and Yaoyu Yang conceptualized the software and wrote most of

the application code. Ben Keller, Abe Miller, Garrett Newman, Phuong Le, Justin Vrana contributed to the application code. Abe Miller and Ben Keller prepared documentation and installation scripts. Tileli Amimeur, Nick Bolten, Leandra Brettner, Cameron Cordray, Miles Gander, Sarah Goldberg, Samer Halabiya, Seunghee Jang, Yokesh Jayakumar, Eriberto Lopez, Jon Luntzel, Cannon Mallory, Abraham Miller, Garrett Newman, Michelle Parks, Sundipta Rao, Ayesha Saleem, Devin Strickland, Chris Takahashi, Justin Vrana, Yaoyu Yang, and David Younger developed Aquarium workflows. Cami Corday, Samer Halabyla, Aza Allen, and Michelle Parks executed and tested workflows and contributed to project ideas while managing the experimental laboratory.

Chapter 6

ALGORITHMS FOR AUTOMATED EXPERIMENT PLANNING



Author	Contribution
Justin Vrana	Project conception, algorithm development, code implementation, lab execution, DNA plasmid construction, yeast plasmid construction, design to BMF conversion
Bree Cummins (MSU)	Circuit topology designs (DSGRN)
Rob Moseley (Duke)	Circuit designs

6.1 Towards building national infrastructure for synthetic biology

The work described in this chapter was part of a massive $80-million led by the Defense Advanced Researcher Projects Agency (DARPA), called the Synergistic Design and Discovery (SD2) group. The project aims are to massively accelerate scientific discovery and design of robust biological systems. Researchers across the United States in fields of mathematicians, synthetic biology, artificial intelligence, biology, and computer science participated in this project towards this goal. The primary philosophy of the program was to develop advanced software and computing infrastructure to drive new methods to accelerate design, experimentation, and the learning process. The massive diversity and scale of the program necessitated the use of individual organizations within this project that accomplishes similar groups of tasks: design, analysis, experimentation, and infrastructure. Organizing groups in this way force the development of abstraction barriers between interacting groups.

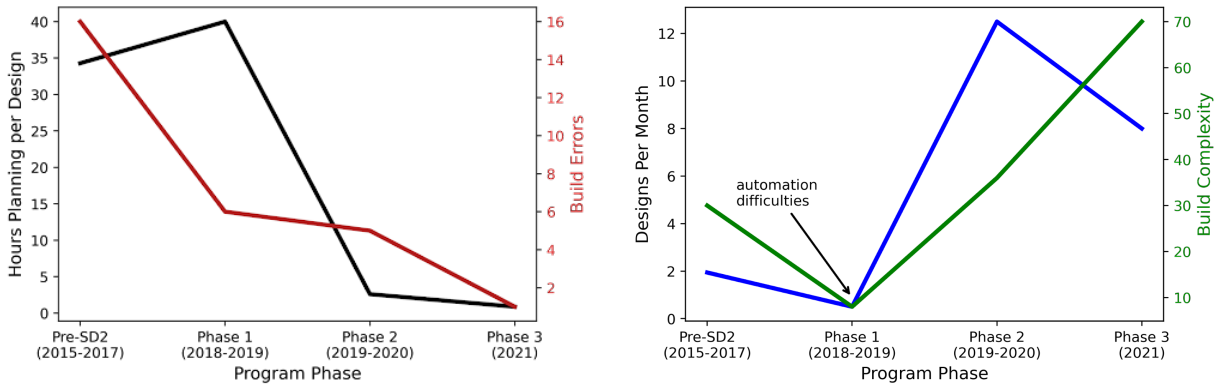
Our group was responsible for a large portion of the experimentation for this massive program. The work I present in this chapter is responsible for the automated conversion of abstract cell designs into executable steps that could be performed by our laboratory on the University of Washington-Seattle campus. This software is almost entirely responsible for the design-to-build pipeline for the yeast group within the program. Towards the start of the program, we received hundreds of strain designs involving complex genetic circuitry and new parts from program participants, primarily mathematicians from Montana State University and Rutgers. What we received from designers were very abstract representations of circuits, with no information regarding types of parts or DNA sequences. From this, it soon became clear how infeasible it would be to have a human researcher struggle to interpret such abstract designs and somehow manage and execute protocols to implement the designs in cells. To give perspective on the scale of designs, in one project we received 100 circuit designs, intended to be implemented in *S. cerevisiae*, with each design requiring up to six different genomic integrations and many of these requiring the assembly of new DNA to perform the integrations. While accomplishing the task of building 100 complex yeast strains

is certainly possible for full-time human researchers to accomplish, doing so is both grueling and not scalable. In addition to the practical difficulties of planning and performing hundreds and thousands of protocols, there are additional difficulties in interpreting designs when the *designer* and *experimenter* are different groups of people. This calls into question what a *design* means and what is the sufficient amount of information required to actually construct a given biological design. If designs are not specific and detailed enough (sequences, strains, etc.), they cannot be constructed in the laboratory. Conversely, too much specification puts an unnecessary burden on the designer or design software; it seems infeasible for every design software to plan down to the individual protocol steps. How does one connect biological designs to actual experiments that implement those designs?

Thinking about these problems, began to draw analogies to other manufacturing fields. When using a 3D printer or a CNC-milling machine, users provide three-dimensional digital representations of what they want to be made. These machines have a software layer that converts designs into a series of machine steps that creates the 3D object in physical space. Specifically, Cartesian 3D-printers and CNC machines convert 3D designs into G-code, a language that tells machine controllers exactly what motors move and how to move them.

With this analogy, I thought it was possible to create new software that would be able to interpret biological design files and output specific laboratory instructions that implement biological designs. This software proved to be extremely challenging and rewarding to make. The primary challenge is how dynamic and diverse different biological samples and protocols can be. The analogy between a 3D printer and a laboratory quickly breaks down as developing a new cell strain in the laboratory involves dozen of different machines (PCR, incubators, electrophoresis, flow cytometers, etc.) and is often a long multi-day or multi-week process. Biological samples themselves have highly diverse properties. These properties often change as a result of performing certain protocols or as a result of natural biological processes. An *E. coli* strain may confer antibiotic resistance after a DNA transformation protocol. It will multiply, evolved, die, or may become contaminated with other strains. This is unlike a 3D printer; while the printing material may melt or solidify, it otherwise remains the same.

Taken together, automatically generating experiment instructions for biological designs is difficult.



(a) Efficiency improvements

(b) Throughput improvements

Figure 6.1: **SD2 program build metrics since implementing CAPP software** Program metrics for the DARPA-led SD2 program since implementing biological CAPP software. Metrics are estimated for the University of Washington-Seattle BIOFAB laboratory. On the x-axis of plots are program phases. CAPP software was first implemented in phase 1 of the program and improved in phases 2 and 3. On the left is a comparison of the estimated number of human labor hours involved in experiment planning per biological design. This metric combines the sum total of hours involved in notebook taking, brainstorming, and planning of experiments. These metrics were estimated by collating data from notebooks and calendars of researchers. The pre-SD2 phase is estimated from Gander et. al 2017 [31]. On the right axis is the estimated number of build errors (strains that fail quality control) during the build processes. The right plot depicts speed ups the number of designs completed per month, along with an estimate of "build complexity" calculated as follows $(n_{\text{DNA assemblies}} + 1)(n_{\text{yeast integrations}} + 1)$ per design. In phase 1 on implementing the software, we experienced many automation difficulties revolving around integrating various software; after initial difficulties, we saw substantial improvements in throughput.

The software I present in this chapter is the first example, as far as I am aware, of a computer-aided process planner (CAPP) applied to biology. CAPPs are used in other fields to bridge the gap between computer-aided design (CAD) tools and manufacturing facilities. The software I present, called Terrarium, converts a simple file called the Biological Manufacturing File (BMF) into a network of executable protocols. Aquarium [128] is used to execute the protocols. To deal with the huge diversity in biological samples and protocols,

Terrarium leverages metadata from Aquarium and develops a model of the laboratory. Using this model, Terrarium attempts to propose workflows similar to workflows previously run by Aquarium. Since a vast majority of previously run workflows in Aquarium were designed by human-researchers, the idea here is to recapitulate what a human researcher would do if given the same planning task.

Upon implementing this software in the SD2 program, we saw drastic improvements in both the number of strains we could build and a drastic reduction in the number of build errors (Fig. 6.1). As of writing this document, the SD2 program is continuing and publications involving this work will likely continue to spin out in the upcoming years. I imagine and hope this work will inspire fully automated synthetic biology pipelines in the future. What is entailed in this chapter below is primarily material for a future publication describing this BioCAPP software.

6.2 Overview of Terrarium

Industrial-scale engineering of organisms is becoming increasingly data-driven, often requiring hundreds or thousands of genetic variants. To meet the demands of designing workflows that construct new strains, I created Terrarium, a software package that automates a workflow design using data from past experiments to inform the design of future workflows. The software borrows heavily from concepts of "smart manufacturing," treating the scientific workflows as a re-configurable manufacturing process that produces biological products and data. The critical feature of Terrarium is that it adapts to changes in inventory and the implementation of scientific protocols, meaning the workflows it designs are always valid. The software has been effective at automating the design of yeast strain construction workflows using simple strain specifications with little human intervention. Additionally, it can be employed for other organisms or generic scientific workflows. The project is a significant step in defining how experiments and laboratory operations can be digitally represented and how their metadata can be used to inform the future design of experiments.

6.3 Introduction to Computer-Aided Process Planning in Synthetic Biology

Cells can be genetically reprogrammed to sense and respond to their environment [31, 116, 129–142], create useful chemical compounds such as biofuels or biologics [143–147], and create new materials [148–150]. Cell engineering is a complex process that requires multi-level design and manufacturing on DNA sequence, protein, cell, or multi-cellular levels [8]. With advances in cell engineering and increases in the scale of experimentation, there is an increase in reliance on computer-aided design (CAD) tools to design various aspects of biological cells. For example, DNA sequence design tools help in the design of primers [124–126] and aid in the effective molecular assembly of DNA into larger more complex constructs [123, 127, 151]. Sequence and design repositories (such as SBOL [152], ICE [153], NCBI [154]) help in the grouping, categorization, and retrieval of genetic components for reuse in complex genetic designs. Some tools automate genetic part composition in sophisticated ways (Cello [43, 137], GenoCAD [155, 156], Eugene, DeviceEditor). Beyond DNA sequence level design tools, other tools like Rosetta allow *de novo* design of proteins with a myriad of interesting properties affecting ligand binding, protein-protein binding, stability, and many other properties [122, 139, 140]. Advances in cell modeling and computing have allowed the creation of cellular and multi-cellular modeling software like TinkerCell.

Despite advances in biological CAD tools, there has been far less published research into how to connect these tools to actual processes in the laboratory. Actual experimental implementation still remains a bespoke process, relying on the expertise of benchtop researchers [112, 128, 157]. While robotic automation is becoming increasingly more prevalent with the utilization of microfluidics, general-purpose liquid handling robotics, and other automated systems [112, 158–166], there still exists many non-automated steps required in routine laboratory work. Cell transformations, for example, are a majorly manual process as there are very few reports of automated systems that automated this process. Generally speaking, most lab procedures require high mental effort and planning by human researchers. Samples must be labeled, arranged and organized; protocols must be chosen, read, comprehended

and understood; researchers have to be familiar with all equipment and machines involved. The result of this is that substantial effort is dedicated to planning and executing of scientific protocols [112]. Furthermore, the general lack of tools and automation in this area of scientific workflow planning calls into question the ability of other labs to replicate published research findings [112, 167, 168].

In manufacturing disciplines, the endeavor of determining which processes to perform given design specifications is known as *process planning*. Whether performed manually by experts or automatically by software systems, process planning is a necessary prerequisite to manufacturing. In modern manufacturing systems, CAD tools are connected to computer-aided manufacturing (CAM) systems using computer-aided process planning (CAPP) systems [169, 170]. CAPP systems provide the essential linkages between design specifications and manufacturing processes [169–176]. The primary function of CAPP systems is to determine an economical selection of processes to manufacture products given design specifications. This process often involves the interpretation of design specifications, selection of machine and cutting tools, determining which operations to perform, evaluation of current inventory, determination of machine parameters, and optimizing plans to reduce lead times and costs. CAPPs are routinely used in concert with computer numerical control (CNC) machining, 3D printing, milling machining, and laser lithography machines to physically generate industrial products. The primary benefit of using a CAPP is that is it creates an abstraction between the design and the manufacturing processes, allowing for the specialization and division of labor, which results in significant reductions in manufacturing lead time and cost [176].

In the field of chemistry, planning algorithms are becoming increasingly more prevalent. Machine learning algorithms and artificial intelligence systems are being used to automatically generate high-throughput experimental plans and create new synthesis routes for compounds [177–180]. Some of these tools perform retrosynthesis planning, the process of finding an efficient chemical synthesis route for a given chemical structure [163, 178, 180–182]. Retrosynthesis planning is highly similar to the description of CAPP tools in manufacturing.

At its most basic description, a retrosynthetic planner takes a design (chemical structure) and converts the design into a series of executable steps (synthesis plan). Such tools create economical synthesis plans by taking into account thing likes material costs, reaction efficiencies, and what compounds can be accessed or purchased from vendors. The recent prevalence and interest in these types of planning algorithms is a testament to their utility.

Despite their usefulness in other disciplines, the concept of CAPP tools has yet to fully enter biology. A common trend in synthetic biology is the development of an automated design, build, test, learn (DBTL) loop [157, 183]. Many publications have focused on automating design for synthetic biology or the test-learn linkages [157, 184]. Far fewer studies have focused on how to bridge the gap between design and build. A few systems have attempted to bridge this gap [123, 127, 183], however, none of these systems are general-purpose, often relying on the use of specific robotics platforms or only automating a narrow range of procedures, such as DNA cloning. A general purpose CAPP that can fully bridge the gap between a biological design and manufacturing has not yet been reported.

In this article, we report the first implementation of a general-purpose biological CAPP tool, Terrarium. The software extends our group's previous software, Aquarium [100, 128] which is a software tool that performs laboratory inventory management and automates protocol execution. Users can specify biological designs by creating a custom file called a biological manufacturing file (BMF). Terrarium converts BMFs into executable networks of protocols, called workflows. Workflows are uploaded to Aquarium where they are then used to generated human-readable instructions to execute the workflows. During execution, metadata on protocol execution time, inventory usage, experimental errors, success rates, materials, and labor costs are collected into a database. Terrarium uses this execution data from the database to update an internal digital model of the laboratory, which it uses to predict lead times and costs of workflows. This laboratory model is used during the process planning to create valid, economical, and efficient workflows. Terrarium and Aquarium forms a software suite that automates the design-to-build linkage, with Terrarium acting as a CAPP and Aquarium acting as a computer-aided manufacturing (CAM) system.

We designed the software so that new biological designs, such as engineered yeast strains, could be easily parsed into a simple BMF which could then be automatically be processed by the CAPP-CAM software. The intention is that the input files are highly flexible, allowing for a variety of CAD tools to interface with the system, hence bridging the design-build gap by employing the CAD-CAPP-CAM software paradigm seen in other manufacturing systems [20, 174, 185, 186].

Here we present descriptions, algorithms, code, and example usages for our software. To validate the efficacy of our tool, we used the software to manufacturing genetically engineered *S. cerevisiae* strains that employ a CRISPR-dCas9-Mxi1 genetic circuitry. To test that the software provides an effective abstraction barrier between design and build, we stipulated that all designs be provided by an outside group unfamiliar with laboratory procedures and the inner workings of the lab. One hundred different genetic circuits were designed and converted to biological manufacturing files (BMFs) by our group. BMFs were used by our software to automatically generate experimental workflows that manufacture the designs. The automatically generated workflows involved over 380 intermediate yeast strains and 86 plasmid assemblies, which totaled over 2000 automatically planned protocols. The workflows the system generated involved a wide variety of protocols that included primer design, primer ordering, PCR amplification, DNA assembly, DNA synthesis, bacterial culturing, bacterial transformation, DNA purification, yeast culturing, and yeast transformation. After executing the workflows, we experimentally tested and validated the resulting yeast strains to confirm genetic circuit function. Further, we show that this software is not limited to just yeast, as the software is able to automatically generate workflows for other organisms, such as mammalian stable cell lines. These results represent a substantial step towards fully developing automated DBTL pipelines.

6.4 Process Planning using Terrarium

Automated process planning refers to the conversion of a design into a set of executable steps that produce the design. The primary goal of Terrarium as a computer-aided process

planner (CAPP) is to automatically generate an economical and executable series of laboratory steps for a given biological design. To effectively create executable laboratory steps, an understanding of what types of protocol and inventory are available in the laboratory is required. Experienced human researchers do this intuitively using their experience and familiarity with the laboratory, albeit in *an hoc* way.

The paradigm behind Terrarium is that the user only provides a bare minimum amount of information about the design, with the software interpreting the design and producing a workflow that is customized to the user's laboratory (Fig. 6.2). After processing the design, the system produces a series of human-readable laboratory instructions so that the designs can be successfully constructed. This is accomplished by leveraging another software, Aquarium [100], to serve as a computer-aided manufacturing system (CAM). Terrarium serves as a computer-aided process planner (CAPP), which provides an automated linkage between designs and manufacturing.

Biological designs are defined in a Biological Manufacturing File (BMF) using the JSON (JavaScript Object Notation) standard [187]. A BMF defines three components: (i) the laboratory configuration, (ii) relationships between biological sub-components in a given design, and (iii) biological definitions, which include the type of biological sample (yeast, bacteria, DNA, sequences, etc.). By design, BMFs are often simplistic. For example, new yeast strains can be defined as providing the parent strain and a set of DNA sequences to be integrated into the genome (Box 6.4). BMFs can also be multi-level and complex. Complex BMFs can be automatically generated by parsing other design formats, such as the Synthetic Biology Open Language (SBOL) [152]. This allows the system to be connected to other upstream computer-aided design (CAD) tools, like Cello [43, 137], as long as they produce files that can be converted to a BMF. This is often the case if the design files are in a hierarchical design format specification such as the commonly used SBOL format.

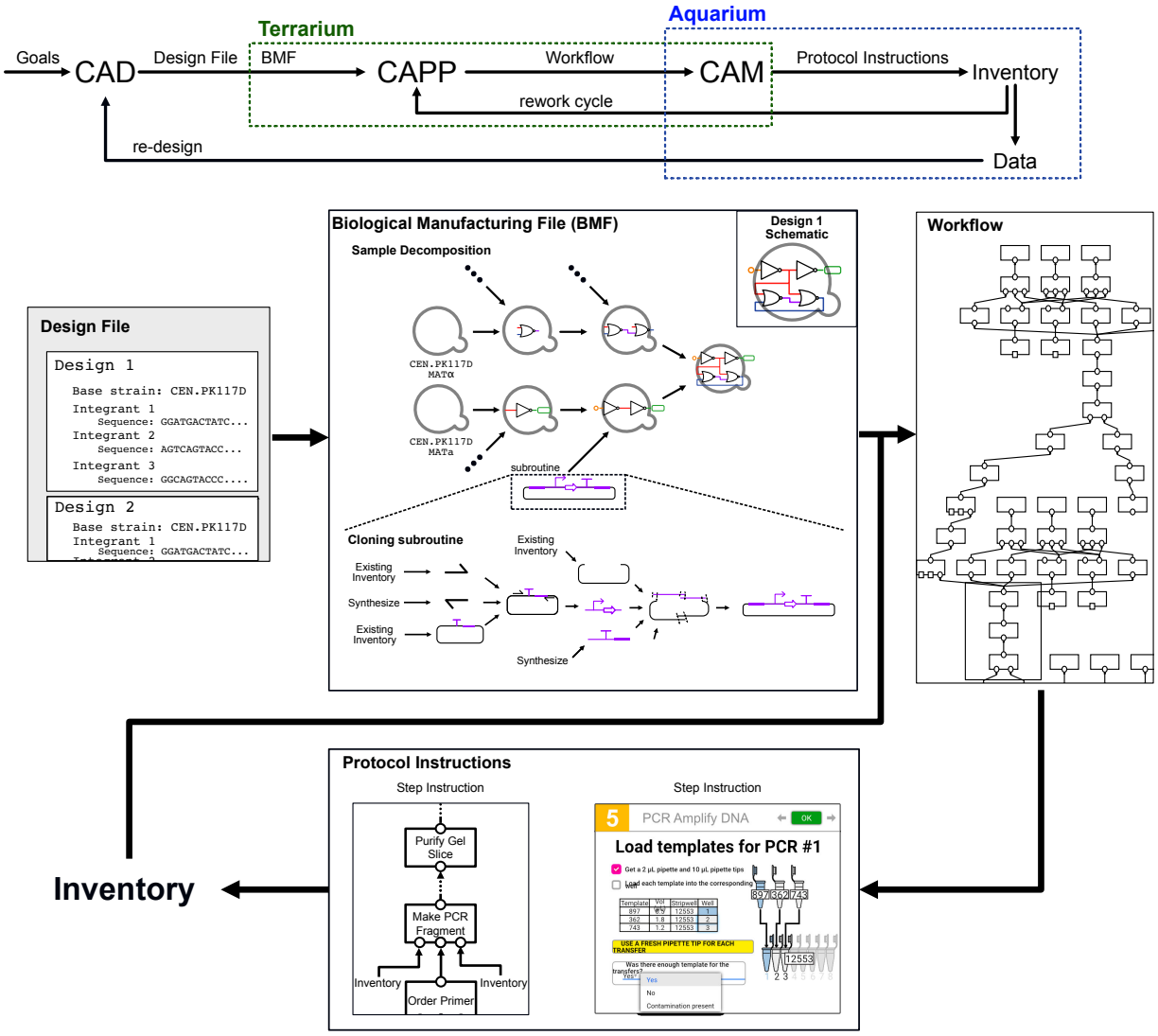


Figure 6.2: **Overview of CAD-CAPP-CAD integration** (top) Terrarium and Aquarium are used as computer-aided process planner (CAPP) and computer-aided manufacturing (CAM) tools. Design files coming from computer-aided design (CAD) tools are converted into a Biological Manufacturing File (BMF), which represents relationships between biological samples. Using lab inventory and other lab information, Terrarium (CAPP) converts BMFs into networks of operations, called workflows. Workflows are executed by Aquarium (CAM) by providing human technicians with detailed and graphical protocol instructions. As protocols are executed, Aquarium updates its own inventory database and other metadata related to the execution runs. Terrarium can use this data on the next iteration of workflows it generates.

Box 6.4: Biological Manufacturing File (BMF) for yeast

```
{
  "__bmf__": true,
  "config": {
    "lab": {
      "url": "http://0.0.0.0",
      "name": "production"
    },
    "constraints": {}
  },
  "goals": [
    {
      "plan_id": "myplan",
      "output": {
        "sample": "CEN.PK2-MATA|yeVenus",
        "object_type": "Yeast Glycerol Stock"
      },
      "dependencies": [
        ["CEN.PK2-MATA", "CEN.PK2-MATA|yeVenus"],
        ["pGPD-yeVenus-tCYC1", "CEN.PK2-MATA|yeVenus"]
      ]
    }
  ],
  "definitions": [
    {
      "name": "CEN.PK2-MATA|yeVenus",
      "sample_type": "Yeast Strain"
    },
    {
      "name": "CEN.PK2-MATA",
      "sample_type": "Yeast Strain"
    },
    {
      "name": "pGPD-yeVenus-tCYC1",
      "sample_type": "Plasmid",
      "fields": {
        "sequence": "AGGGCATGGATCAGGATGATAGAAAAGAGATGG...",
        "cyclic": true
      }
    }
  ]
}
```

Terrarium (CAPP) processes a BMF by converting the file into an executable workflow for the Aquarium (CAM) to execute (Fig. 6.2). This process occurs in four steps. First Terrarium uses the lab configuration information provided in the BMF to access the laboratory inventory systems (LIMS) of the CAM system. Inventory information and metadata from any previously executed workflows is used to generate a model of the laboratory. This model provides information about how to connect protocols (called *operations* here) together in the lab and can make predictions for lead time, costs, labor, and predictions of experimental

errors for proposed workflows.

The second step is to map biological definitions provided in the BMF to either existing inventory or to sets of operations that can produce the defined samples. For cases in which a DNA sequence is defined, Terrarium uses an optional process planning subroutine, called DASi, to automatically propose molecular biology reactions that assemble the provided DNA sequence.

In the third step, Terrarium constructs a directed graph of all possible workflows that could generate the goal samples in the BMF. This large workflow graph includes any available inventory, desired goal inventory provided by the BMF and any intermediate operations that could produce goals from inventory. This process makes use of any dependency definitions provided in the BMF to maintain a strict ordering of operations used in workflows. Because operations can often be connected in many different ways, edge weights are applied to the graph that corresponds to how often two operations have been connected in past workflows, with high scores corresponding to rare connections and low scores corresponding to highly likely operation connections. These edge weights are used in the next step to find optimal workflows.

In the final step, spanning trees (ST) between all available inventory and goal samples are computed. This procedure is a special variant of the Steiner-Tree problem [188, 189]. The Steiner tree problem is an NP-complete networking problem [189] whose goal is to find a minimum spanning tree (MST) between a set of terminal nodes. Since operations can have multiple inputs and operations that are missing required inputs are not executable, an additional constraint is added to the Steiner tree problem in that any solution is required to have all valid operations. Because of the size of the graphs involved and the likely NP-completeness of the variant of this problem, a finding true minimum tree is not feasible, and so a heuristic algorithm is employed to identify spanning trees. These spanning trees refer to candidate workflows for producing the designs given by the original BMF.

Candidate workflows can be evaluated using the laboratory model Terrarium generates. Monte Carlo simulations based on experimental error rates, recorded execution times, and

estimated material and labor costs are used to further evaluate and rank workflows (Fig. 6.3). The user can then choose and optionally manipulate the automatically generated plans. Workflows are sent to the CAM (Aquarium [128]) as a JSON file that includes all information about inventory, sample definitions, operations, and any parameters required for execution. After submission, designs can be successfully manufactured.

6.4.1 Workflow and protocol execution using Aquarium

Laboratory protocols are typically represented in text format as a series of step-by-step instructions (Fig. 6.2). Often included are descriptions of the data or materials required to execute the protocols and the types of lab samples and data the protocol produces. Laboratory protocols are analogous to manufacturing operations, which describe the process in which materials are converted and transformed into other materials. Both laboratory protocols and manufacturing operations have concepts of inputs, outputs, and inventory types. Biological protocols are represented as a series of step-by-step instructions that generate data and/or creates new biological samples. Protocols have defined inputs and outputs that define the bounds of their operation. Based on the types of these inputs and outputs, protocols can be connected into networks called *workflows*.

Instructions to the scientist and benchtop researchers are displayed through a browser interface (a process called *protocol execution*). How and what instructions are displayed are governed by computer code, which can automatically generate readable step-by-step instructions based on their inputs and the current laboratory state. During the execution of a protocol, data and metadata are logged and tracked to a database. Metadata recording start and end times, number of errors, and successes are all recording during protocol execution. Protocols can be parameterized to generate instructions in different ways. The protocol execution process is reported in detail by our group previously [100, 128].

6.4.2 Simulation of laboratory workflows

Metadata generated by the CAM system is used to create a graphical model of laboratory behavior, which allows for useful predictions, such as lead time and material/labor costs. In downstream planning algorithms, these calculated predictions are used to compare and score workflows. To produce a statistical model of the laboratory, data is gathered from a subset of previously run workflows, which are represented as a set of directed acyclic graphs (DAGs), with each node corresponding to a protocol that was executed and each incoming edge to the node corresponding to that protocols dependencies. For each node and edge in the graph, summary statistics are calculated for each node and edge

The graphical model is used to predict lead times and costs for new proposed workflows. To make the predictions, a Monte Carlo simulation is used to simulate stepping through the entire workflow. During the simulation process, accumulated errors, costs, and lead times are sampled from the previously computed distributions. During simulation, there is a non-zero probability of a protocol entering an 'error' state. This probability is estimated from prior data and represents the probability of a particular protocol failing in the actual laboratory. When the simulated protocol enters the error state, errors are backpropagated to their predecessors with a non-zero probability and the simulation retries complete the errored protocols. During simulation, accumulated material cost, labor cost, and lead time are calculated for each Monte Carlo sample (Fig. 6.3).

6.5 Automated Manufacturing of DNA

In some cases for the exact sequence provided in a BMF is not available in the laboratory as inventory. In these cases, DNA sequences must be created in the laboratory by a combination of DNA subcloning or DNA synthesis from vendors like IDT or Twist. For this type of planning, Terrarium uses a planning subroutine to automatically generate DNA cloning assembly plans from sequences. Similar to the outer CAPP algorithms, the inner DNA planning algorithms are designed to economically utilize available inventory to create an

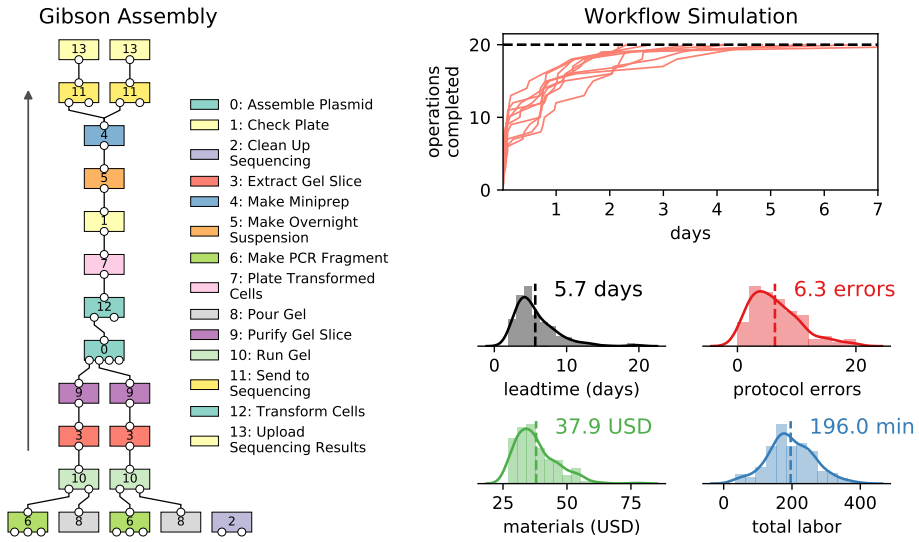


Figure 6.3: **Simulations and predictions of workflow lead time and costs** Monte Carlo simulation of workflow execution is used to evaluate workflows and determine the lead time, labor costs, material costs, and estimated number of experimental errors. Depicted on the left is a simple Gibson DNA assembly automatically generated by Terrarium. The left upwards arrow indicates that operations at the bottom of the workflow are dependents to operations above them. On the right are simulated progress for 10 MC simulations. The lower right depicts the estimates of lead time, error rate, and costs as histograms from 200 MC simulations. Operations are simulated to have three main states "waiting", "running", "done", and "error". During simulation, Terrarium steps through each operation state using the dependent operation states. A learned error rate specific for each operation may cause an operation to enter the "error" state, in which case the error back-propagates downwards partway through its ancestors. How this error propagates is determined by learned error rates determined from previously executed workflows. As an example, if sequencing validation fails (operation 15 "Upload Sequencing Results") error is often back-propagated through to the DNA preparation operation (operation 4 "Make Miniprep"). This is because simply repeating the same sequencing reaction is unlikely to yield different results, and so prior users of CAM (Aquarium[128]) often error out all three operations to produce a new DNA preparation sample from a new bacterial colony; this type of behavior is captured in the laboratory model Terrarium creates.

assembly plan that will produce a given DNA sequence. The input to this software is either as a string of characters, a GenBank file, an SBOL file, or another BMF. The set of algorithms and software is provided as a separate standalone software called DASi. Unlike other cloning automation software, DASi does not require further design specifications beyond the DNA sequence to effectively create DNA assembly plans.

The DASi algorithm follows the same paradigm as the outer Terrarium planning algorithms, albeit with additional steps and the use of additional software for designing DNA sequences. Briefly, DNA sequences available in inventory are used to generate a graph of all possible DNA assemblies. A path through the graph corresponds to a valid DNA assembly. A course simulation of molecular biology reactions is used to calculate material costs and estimate assembly efficiency every assembly at every potential assembly junction. Finally, shortest path algorithms (or shortest cycle for cyclic plasmid sequences) are used to determine the best DNA assemblies. Once an assembly is returned from DASi, the output file can be converted into a BMF which can be processed by Terrarium into a validated workflow. BMF from DNA assemblies and can be combined with other larger BMFs (such as yeast construction BMFs) simply by concatenating the biological dependencies and definitions of the BMF file.

DASi and Terrarium generate economical and complete workflow solutions for assembling DNA (Figs. 6.4, 6.3). DASi provides workflow solutions using the flexible homology-based Gibson assembly method [60]. For DASi to work, users only provide the complete output DNA sequences that are to be assembled and laboratory inventory information. This provides a similar design-to-build abstraction barrier that most DNA synthesis vendors (IDT, Twist, etc.) provide; users do not need to know the inner workings of how DNA is manufactured. DASi and Terrarium extend the capabilities of using DNA synthesis vendors by providing solutions to assembling DNA using inventory physically found in the laboratory. The DASi and Terrarium algorithms produce a series of sub-cloning operations that will produce a given DNA sequence. If it is determined to be economical, the software will also suggest DNA by ordered from outside vendors (Fig. 6.4; operation 9 "Order gBlock Fragment").

Otherwise, if it is cheaper or more efficient to perform other reactions, such as polymerase chain reaction (PCR), those will be planned in lieu of DNA synthesis. For DNA synthesis, DASi creates estimates of lead time and material cost using an additional configuration file; these configurations are taken by collating information from synthesis vendors like IDT or Twist. DASi calculates "synthesis complexity" which is used to prevent synthesis failures by the vendor. Synthesis complexity involves calculating windowed GC%, DNA hairpin, and DNA repeats for sequences to be synthesized. In cases where primers are to be designed for downstream PCR reactions, an algorithm using Primer3 [124–126] is used to automatically generate primer sequences. This algorithm automatically generates overhangs to be used for homology-based DNA assembly; eventually, when interpreted by Terrarium, primers are created via an operation that requests primers to be synthesized by outside vendors (Fig. 6.4; operation 8 "Order Primer"). DASi automatically shares common DNA fragments and materials between similar DNA assemblies (Fig. 6.4; inset operation 0 "Assemble Plasmid"). The final assemblies the algorithm produces take into consideration all material costs involved in the assembly and predicted efficiencies for each junction and estimated success rate for individual reactions.

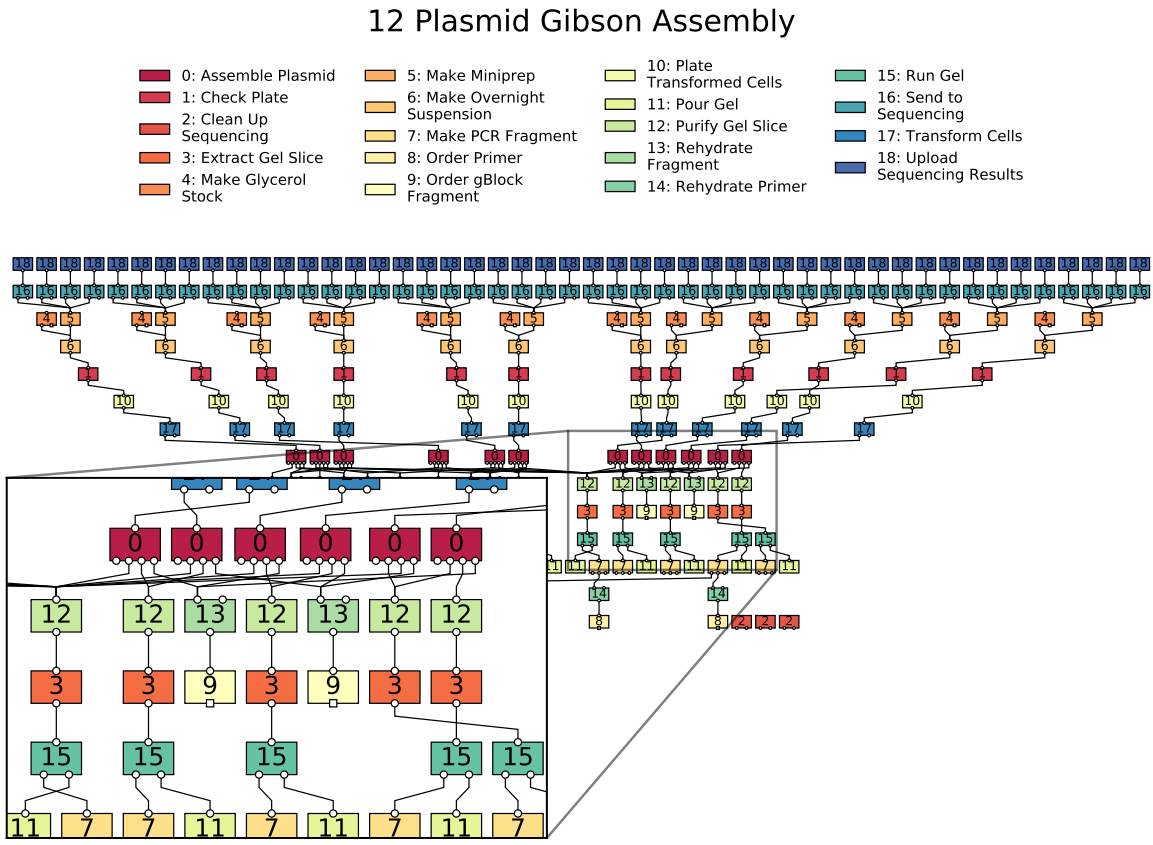


Figure 6.4: **Generated workflow for construction of 12 DNA plasmids using Gibson Assembly** Example of an automatically generated Gibson assembly of twelve different plasmids created by Terrarium and DASi. Dependent operations are on the bottom and final operations are at the top. Twelve DNA sequence designs were provided to the software as well as access to the laboratory inventory system. Plasmids share several common inserts and backbone sequences. In the figure inset a variety of operations are used to accomplish the DNA assembly, ranging from ordering primers, amplifying and purifying DNA and ordering DNA for synthesis. Crossing 'wires' between "Assemble Plasmid" operations indicates that common DNA fragments are being used in multiple assemblies. Inputs with no incoming wires indicate a sample was sourced from the lab inventory.

6.6 Automated Manufacturing of Yeast Strains

To create new yeast strains using Terrarium, a simple example is to provide a BMF file with definitions for goal strain, the parent strain, and the integrative DNA sequence (Box 6.4, Figs. 6.5,6.13). Terrarium correlates information in the BMF with laboratory information provided by the LIMS & CAM. Because of this, biological definitions must correspond to entries found in the LIMS, otherwise, Terrarium would be unable to interpret biological designs. In some cases, new intermediary biological samples are required for manufacturing, in which case, the LIMS is simply updated with these new intermediary samples. When interpreting the BMF, Terrarium uses data on how similar biological samples were created from previous workflows. In the laboratory in which we conducted this study, yeast samples are most often created from a yeast transformation protocol in which an integrant is linearized by a restriction enzyme (PmeI) and genomically integrated into the yeast genome by a lithium acetate protocol (LiAC) transformation[61]. Using this data from the laboratory, Terrarium correctly discovers a workflow path using this procedure (Figs. 6.13,6.7,6.8) by employing the necessary protocols involving DNA preparation, yeast culturing, transformation, plating, and selection. BMFs can be created that employ multiple integration events (Fig. 6.6). These larger BMFs are interpreted by Terrarium to create large complex yeast construction plans that can be executed by the CAM (Figs. 6.7, 6.8). To validate strains, colony PCR operations are used to validate the insert; while some quality control (QC) operations can be automatically planned, the nature of these QC operations often fall outside the scope of the Terrarium planner and require human intervention and interpretation.

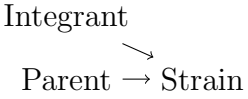


Figure 6.5: **BMF dependencies in simple yeast strain constructions** Dependencies in a simple yeast strain construction forms a simple directed graph between parent, integrate, and output strain

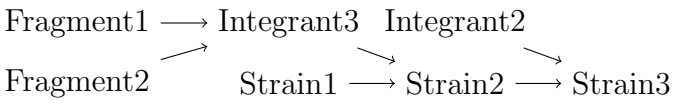


Figure 6.6: **BMF dependencies for yeast with multiple integrants** Dependencies for yeast strain construction can be multi-level, involving multiple integrations or DNA assemblies.

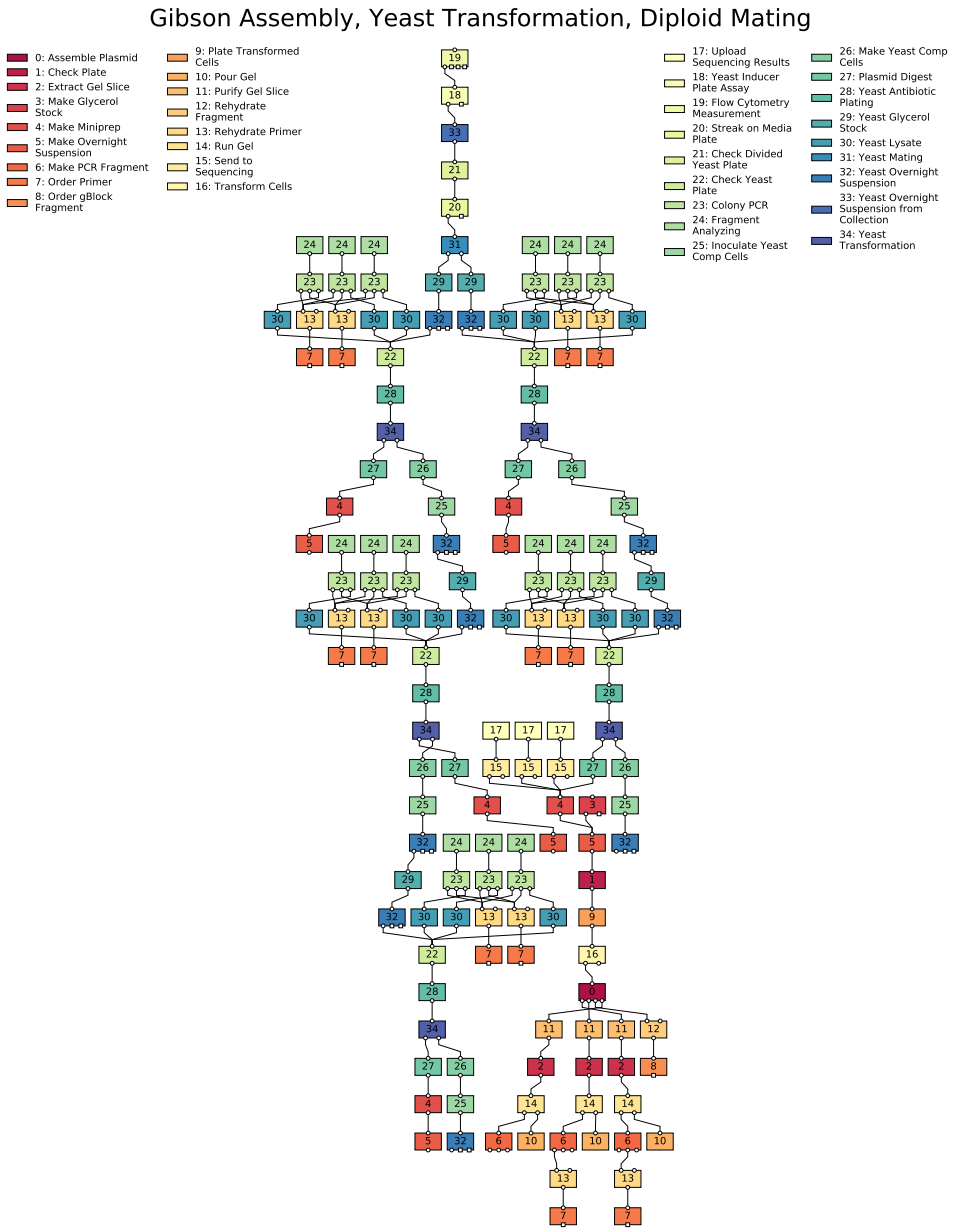


Figure 6.7: **Generated workflow for yeast strain construction involving DNA assembly, yeast transformation, and diploid mating** An example of an automatically generated workflow for the construction of a diploid yeast strain involving five genomic integrations, one Gibson assembly, three PCRs, and one yeast mating. Quality control (QC) operations like colony PCR and DNA sequencing are only partially planned by the system and final plans must be inspected and adjusted by a human researcher before submission.

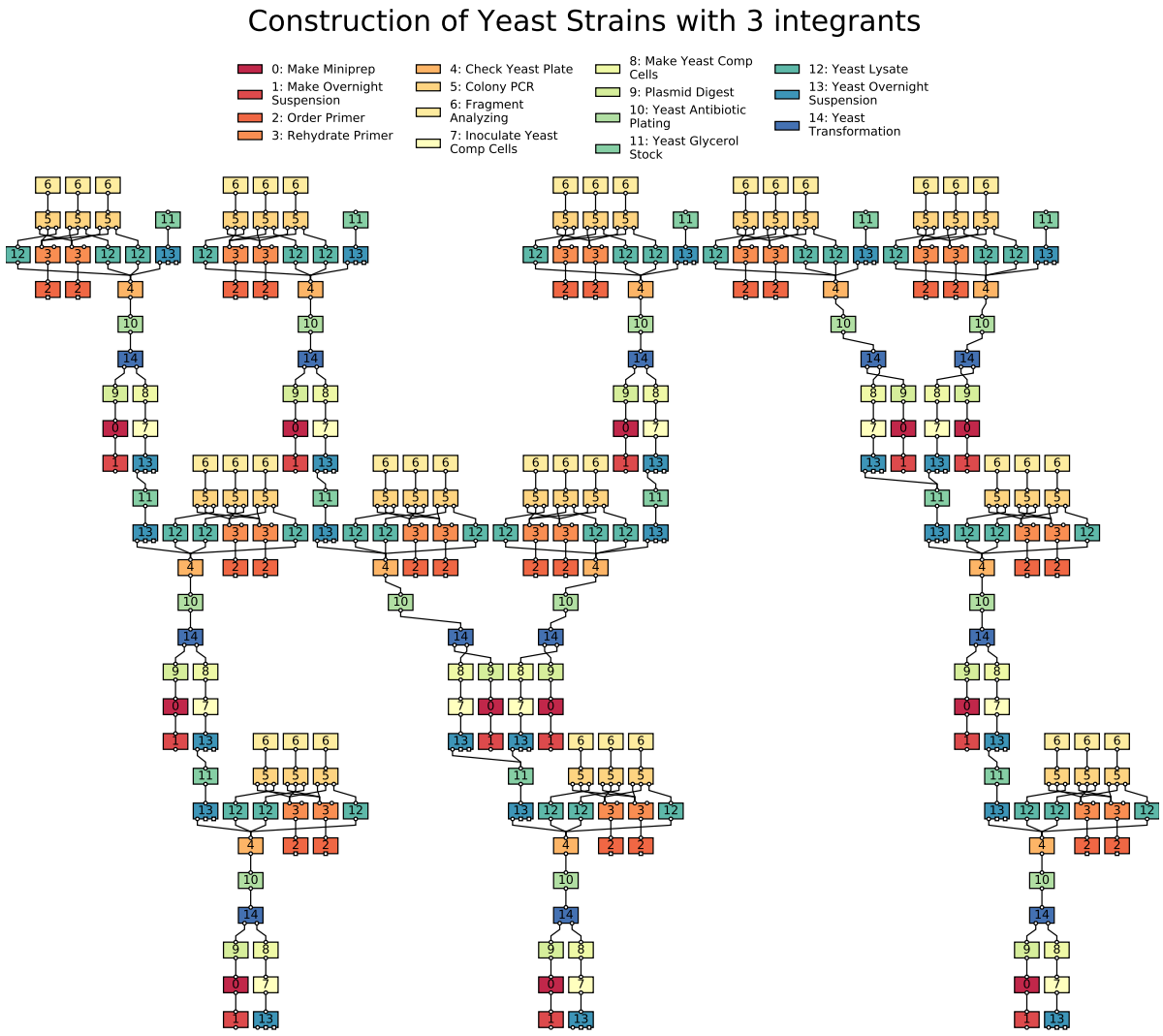


Figure 6.8: **Generated workflow for construction of three-integrant yeast strain** Yeast strain construction involving three sequential integrations of plasmids. This workflow is only a small portion of a larger 900-operation workflow generated by Terrarium and DASi. The BMF that generated this workflow is large and was automatically generated by custom scripts by converting SBOL files to BMF.

6.7 Automating Design-to-Build for Engineered Yeast Strains

We challenged our software to automatically manufacture 100 genetic circuits in *S. cerevisiae* designed by outside collaborators unfamiliar with the inner workings of the laboratory. We received designs as abstract circuit topologies and converted them into BMFs using custom scripts (Section 7.1). We used a CRISPR dCas9-Mxi1 genetic circuit platform to implement genetic circuit designs [31]. These designs consisted of up to six different genomic integrations per strain, some of which required the creation of DNA assemblies. Terrarium successfully created experimentally valid workflows that constructed the strains (Fig. 6.9). The software generated a multi-level and hierarchical plans that included over 2000 different operations consisting of PCR, primer synthesis, DNA synthesis, Gibson assembly, yeast transformation, DNA preparation, *E. coli* culturing, yeast mating, and *S. cerevisiae* culturing. Workflow plans were executed by laboratory technicians via Aquarium. During the course of execution, over 380 intermediary strains and 86 new plasmids were generated. To validate the constructed strains, we used colony PCR to confirm the genomic integrations. We confirmed the performance of the circuits by using a chemical induction protocol followed by flow cytometry (Figs 6.9, 6.10). The completed genetic circuit strains, which included circuit feedback loops, exhibited a linear amplifier response; the performances of some of these circuits are displayed in Fig. 6.10. Importantly, this process demonstrated that abstract designs could be successfully translated by software and automatically manufactured at a separate facility. The design of the entire workflow was automatically generated by Terrarium/DASi, from the design of individual DNA primers to the yeast transformations.

6.8 Adapting the System to Other Organisms

Because Terrarium uses a data-driven approach to create workflows, it can be easily used for other organisms and workflows. Terrarium pulls data on previously run workflows from CAM (Aquarium) to generate a laboratory model; this model includes information on the connectivity of operations in workflows. During the planning process, it uses this informa-

tion to generate valid workflows that are similar to workflows found in the CAM system. As an example, it can be used to generate workflows to generate stable cell lines, provided there have been previous examples of similar workflows that have been executed previously as seen in Fig. 6.11. Here, A BMF defining a new mammalian cell line can be specified with a lentivirus and parent strain as dependencies. The lentivirus definition itself has a dependence on a DNA sequence that is to be used in that packaging of the lentivirus. In this example, Terrarium automatically plans a DNA assembly and plans to have it packaged in lentivirus and used in lentiviral transduction to produce the final strain. The algorithmic strategy behind this type of solution is somewhat trivial. In this example, there is only one type of lab operation that takes a mammalian cell line and lentivirus and produces a new different mammalian cell line, the "lentivirual transduction" operation. Using data from other workflows previously executed, workflows that involve lentiviral transduction typically have other operations related to lentiviral transduction (for example "Harvest Lentivirus", "Package Lentivirus", etc.). The optimal workflow involves these other related operations. Since the provided DNA sequence does not exist, it performs the DNA sub planning routine to create an optimal DNA assembly which is then used in lentiviral packaging. This whole procedure is captured in the core Terrarium algorithm, which in its essence solves the minimum spanning tree between available inventory and goal samples specified by the BMF, using operations/protocols and intermediary nodes to produce the tree. The core algorithm of Terrarium is not specific to any organism or particular workflow; rather it is driven by data from previously executed workflows. Terrarium can therefore be used for any organism.

6.9 Materials and Methods

6.9.1 Plasmid construction

Backbone and insert fragments were amplified with PCR, gel extracted, purified, and assembled using Gibson assembly [60] using standardized assembly linkers. Backbones contained a high-copy E. coli origin of replication and ampicillin resistance for propagation. The yeast

expression cassettes were flanked upstream and downstream by approximately 500 bases of chromosomal homology to the yeast genome and PmeI restriction sites for linearization before transformations. Plasmids were sequence-verified using Sanger sequencing.

6.9.2 Yeast strain construction

CEN.PK113-7D strains were generously provided by Ginkgo Bioworks. Strains were constructed using genomic integration from linearized DNA. Integrative plasmids were linearized using PmeI digestion (37C, 30 min) to cut upstream and downstream of the chromosomal homology. Unpurified, linearized DNA was transformed into yeast cells using a standard lithium acetate protocol [190]. Strain selection was performed on solid synthetic-complete (SC) using auxotrophic or antibiotic markers. Diagnostic colony PCR was performed to verify integration into the proper locus. Strains were picked from single colonies and stored long-term at -80^{o}C in a sterilized 30% glycerol and media solution. Strain retrieval was performed by plating glycerol stocks onto solid media plates (YPAD) grown for 2-3 days at 30^{o}C and picking single colonies for liquid culture. All yeast cultures and assays were grown at 30^{o}C shaking at 275 RPM.

6.9.3 Primer design

Primers were designed using Primer3 v4.1.0 [124–126]. Primer designs were created using a simple Python wrapper found at https://github.com/jvrana/primer3-py-plus.

6.9.4 Protocol execution

Protocols were executed by technicians following on-screen instructions generated by Aquarium [128]. Occasionally, the same protocol was executed by multiple technicians at the same time or sequentially (one technician leaving and another starting where the other left off). Technicians were trained beforehand. Various skill levels and experience. During execution, technicians were unaware of project goals and were instructed to exactly follow the steps and

instructions provided by Aquarium.

6.9.5 Genetic circuit and topology design

A software tool, Dynamic Signatures Generated by Regulatory Networks (DSGRN) was used to design all genetic circuit topologies [191]. Briefly, the tool scores the dynamics of circuit topologies, represented as directed graphs, by assuming each node in a circuit network exhibits a step-like response behavior. This step-like behavior is very computationally efficient to compute and so the tool examines a massive number of different circuit topologies for a user-provided behavior.

6.9.6 Software availability

All open-source software is available on Github. The open-source CAPP software can be found at https://github.com/jvrana/Terrarium. Standalone DNA cloning planning software (DASi) can be found at https://github.com/jvrana/DASi-DNA-Design. Aquarium software is available at https://github.com/aquariumbio/aquarium along with the Python API interface to aquarium https://github.com/aquariumbio/pydent.

6.10 Discussion

6.10.1 Advantages of top-down planning

There have been a number of reported software packages that perform algorithmic optimization of molecular biology procedures [8, 123, 127, 156]. For example, **j5** (now part of Teselegen) determines semi-optimal cloning procedures from a list of available reactions (e.g. SLIC, Gibson, CPEC) to efficiently assembly sequences [123]. **RavenCAD** [127] is a combinatorial procedure that creates hierarchical assembly plans for restriction-type cloning from predefined 'parts' (such as promoters, terminator, coding sequences, etc.). Central to these types of algorithms is the abstract concept of a genetic 'part', a user-defined DNA sequence that has some meaning or intended function in more complex designs. Parts are

often stored in convenient reusable physical libraries, often referred to as *toolkits* or *toolboxes* [40, 192, 193]. Toolkits are often designed around a very specific type of cloning, such as RSII based restriction cloning (Golden-Gate) [192]. These types of optimization/planing procedures, we refer to as *bottom-up* optimization/planning, as optimization occurs with pre-defined definitions of parts and sequences.

Unlike other DNA assembly algorithms, DASi contains no reference to or concepts of 'parts'. It is a purely sequence-based optimization procedure and requires as inputs only the available inventory and goal sequences (plus optional optimization parameters). The available inventory and goal sequences can be *any* sequence. We call this a *top-down* as we start from our goal DNA sequence and determine the optimal reaction to produce the sequence. This type of top-down optimization eliminates any burden on human designers or software to define a list of 'parts' beforehand or maintain lists of parts. This also eliminates requirements for upstream design software to produce meaningful definitions of 'parts'.

An example of the advantages of top-down planning is DNA synthesis. DNA synthesis vendors (Twist, IDT, etc.) receive DNA sequences and use that information to perform the necessary reactions to synthesize the provide sequences. While users may use 'parts' in their designs and abstractions, this information is extraneous to DNA synthesis. The result of this is much greater flexibility on DNA sequence designs as the manufacturing process is almost completely decoupled from the design[1]. Other good examples are recent developments in automated retrosynthesis planners [178, 180, 194], which propose chemical reaction routes for a given top-level compound.

6.10.2 Adapting CAPP to changing laboratory

In a conventional laboratory, protocols are continuously updated and improved over time as new research, results, and literature come into play. Similarly, new protocols are introduced

[1]There are limits to this decoupling. For example, DNA sequences must have sufficiently low 'synthesis complexity' (limiting sequence repeats, poly-G sequences, etc.) so that vendors can reliably manufacture the sequences.

and old less-relevant protocols are removed from the lab. This would make it challenging for any planning software that relies on the specifics of a protocol for planning. By design, our CAPP software is unaware of the internal implementation of protocols/operations. Instead, it utilizes a statistical representation (error-rate, cost, connectivity, etc.) of an operation that is identified by its prior use in actual workflows. These representations can be continuously updated as the lab itself changes over time. Further, data accumulated from the lab can be filtered by user, workflow type, dates, or other queries to produce more useful representations of the lab.

6.11 Conclusion

In this chapter, I outlined software and algorithms for creating a general-purpose computer-aided process planner (CAPP), called Terrarium. This is, to my knowledge, the first implementation of a CAPP for biology. Terrarium serves as an automatic linkage between design tools and manufacturing. By combining it with Aquarium, which can serve as a computer-aided manufacturing (CAM) system, I demonstrated that we can generate automated CAD-CAPP-CAM pipelines to process biological designs into engineered biological cells in the laboratory. As the scope of synthetic biology increasing, automating linkages between design and implementation is going to become increasingly important.

In the area of DNA synthesis, we have seen a general trend of decoupling between the design of DNA sequences and the actual chemical processes that synthesize the DNA. This allows scientists and researchers to design hundreds of thousands or millions of different DNA sequences without having to consider how the manufacturing process works. Companies like Twist or IDT specialize in the conversion of these digital DNA sequences into actual physical DNA samples. In a similar way, I envision tools like Terrarium becoming an essential component to automated systems that manufacture complex biological samples, like engineered cell lines. Rather than implementing the laboratory procedures themselves, I imagine a future where researchers could order hundreds of thousands of engineered *cells* and remain, for the most part, divorced from the implementation for how these engineered cells are constructed.

We've taken this exact strategy (albeit on a smaller scale) with our group's involvement in the DARPA-led Synergistic Design and Discovery (SD2) program. We received hundreds of abstract cell designs from collaborators and converted them into physical strain designs. These collaborators (being mathematicians) had no experience in the laboratory and therefore treated our system as a *blackbox*, similar to how most researchers treat Twist or IDT when they order DNA. Without the type of automation software I described in this chapter, these collaborators would be unable to implement their circuit designs; the software was solely responsible for the conversion of designs into physical realizations of those designs.

While Terrarium algorithms have been tested in production, the system has several limitations. First, it is heavily intertwined Aquarium, merely because Aquarium is the only known software that completely integrates laboratory information management system (LIMS) with protocol execution. Second, Aquarium relies on humans to execute protocols; while this makes the system highly flexible, it limits the throughput of laboratory work. Terrarium is capable of routing and planning thousands of operations, it is not feasible for humans following Aquarium to execute these protocols in a high-throughput way. To fully realize the potential of Terrarium, it is important to eventually implement automated platforms into Aquarium's execution procedures. Lastly, Terrarium only has limited error-correction capabilities. Many scientific protocols have high error rates (DNA assembly and transformations being examples). Terrarium can, at best, replan workflows (known as a "rework cycle") once it encounters an error. It has no capabilities to analyze results and adapt to experimental data. Often, errors must be correct manually.

To finally conclude, Terrarium is the first reported CAPP for biology and represents a significant step in the development of fully automated design-build-test-learn pipelines. It is currently the subject of a draft publication.

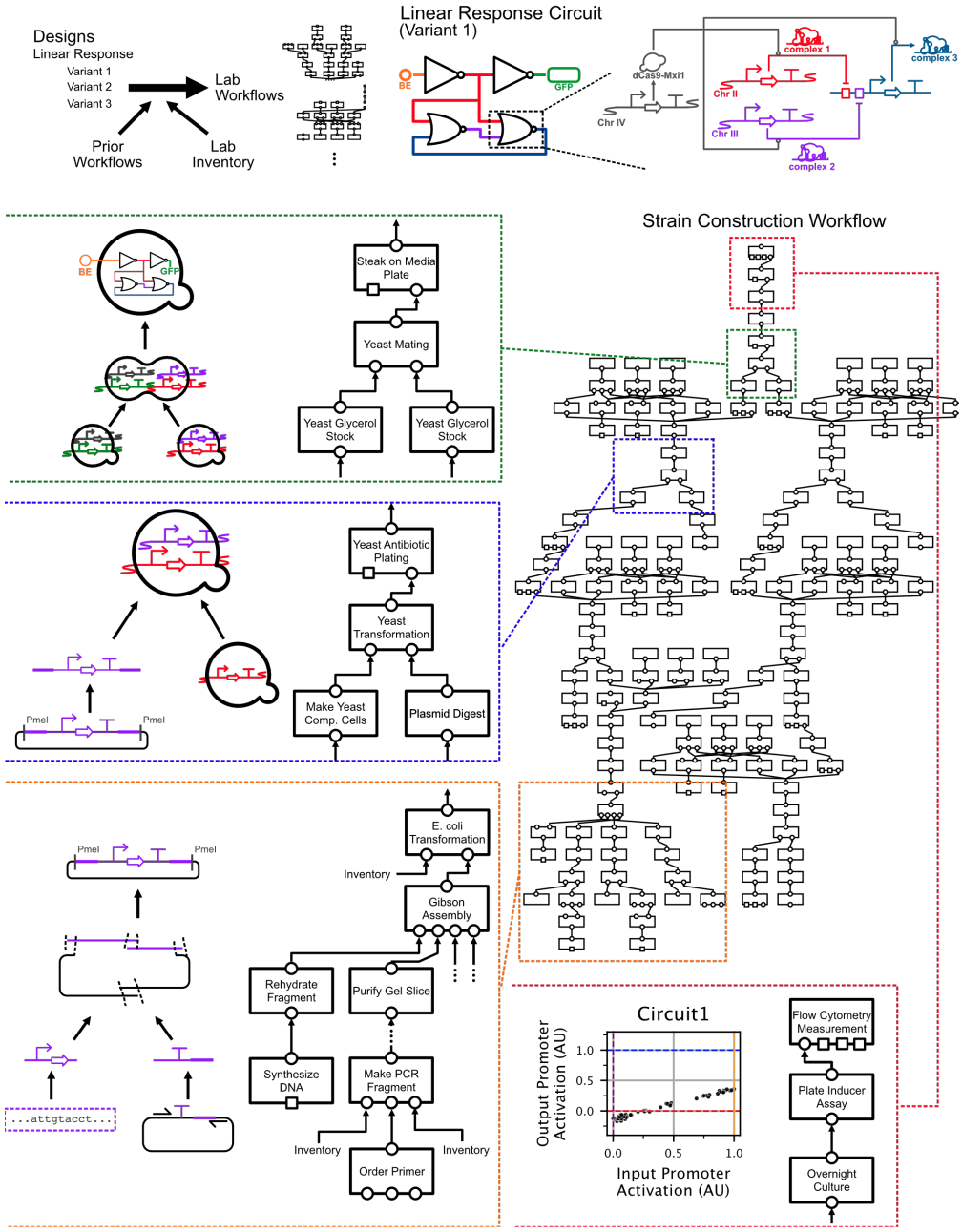


Figure 6.9: **Automated construction of yeast genetic circuits** An example of converting genetic circuit designs to an executable experimental workflow. Many abstract designs were received as graphical networks and converted into genetic circuits using CRISPR-dCas9-Mxi1 circuits[31] using custom scripts (top right). These genetic circuit designs were converted to BMF and processed to created complete workflows (center right). Workflow included a variety of operations including yeast mating, yeast transformation, and Gibson assembly (center left). Flow cytometry analysis was used to measure the circuit characteristics (lower right).

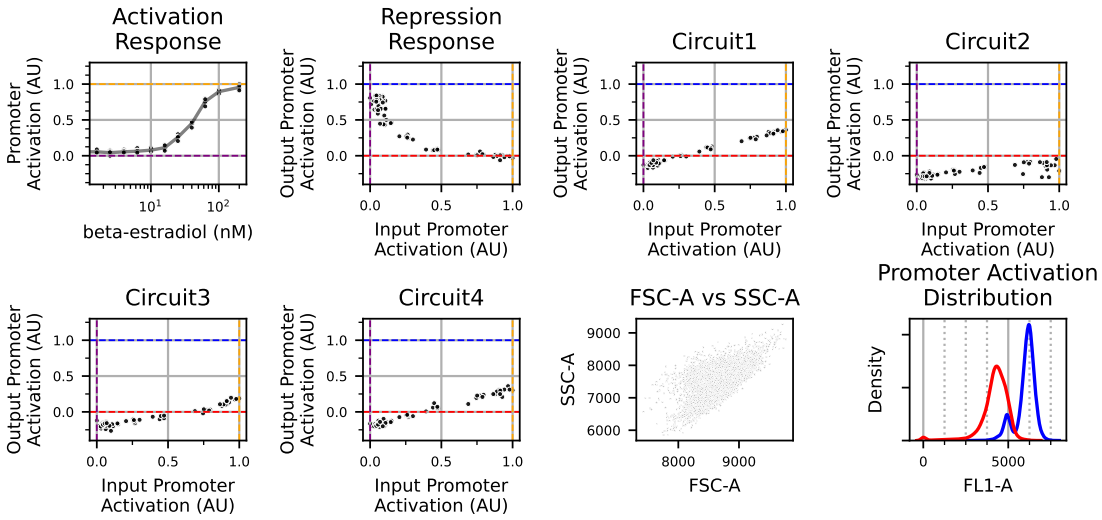


Figure 6.10: **Selected dose-response plots for yeast genetic circuits** Genetic circuits were constructed from BMFs, where were converted to workflows by Terrarium and Dasi (CAPP) and executed by Aquarium (CAM). CRISPR dCas9-Mxi1 genetic circuits were used from a prior publication[31]. An inducible beta-estradiol promoter was used to control input gRNA expression. Dose-response curves for four of these circuits are displayed along with inducible fluorescent control (top-left). Inducible promoter control (top-left) was used to estimate input promoter activation (AU) for the four remaining circuits. Output promoter activation was normalized to the highest and lower fluorescence readings as determined by a constitutively expressing fluorescent reporter (GFP) and non-fluorescent strain. Forward and side scatter (FSC-A vs SSC-A) are shown; cell gating is not shown. Horizontal blue and red lines on all plots correspond to the highest and lowest flow cytometer readings found in the lower right plot. Vertical purple and yellow bars correspond to the maximum and minimum input promoter activation as determined by the inducible fluorescence control on the upper-left.

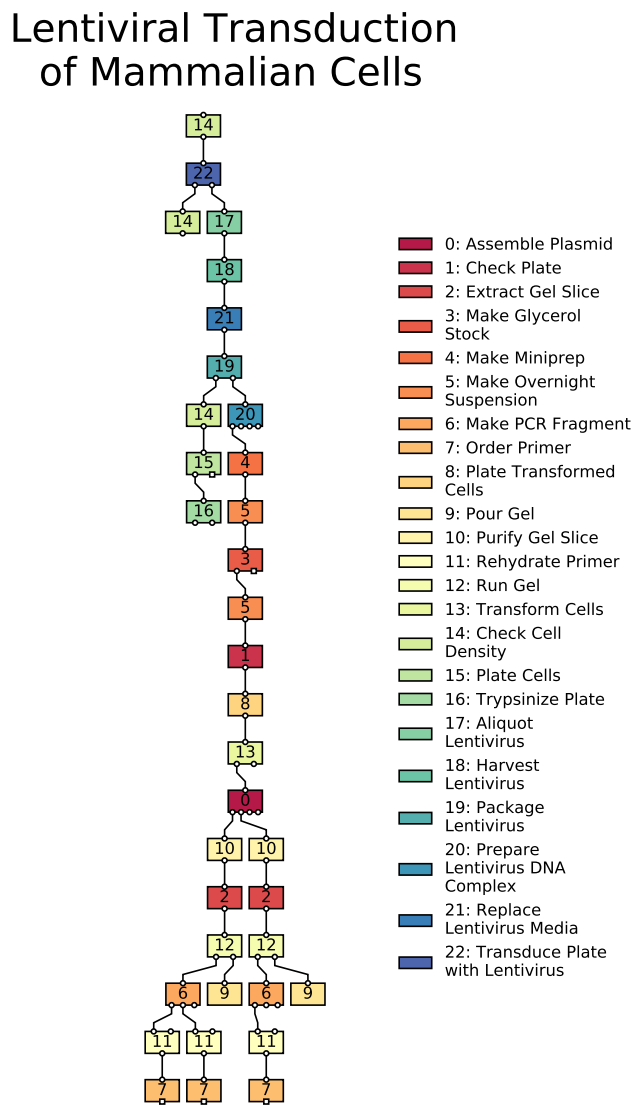


Figure 6.11: **Generated workflow for construction of mammalian stable cell line** An example of an automatically generated plan for stable cell line creation in mammalian cells. Towards the bottom, a small Gibson assembly creates a new plasmid. The resulting plasmid is packaged into lentiviral particles by co-transfection with lentiviral packaging, envelope, and transfer plasmids (op. 20). The lentivirus is used to transduce cells to begin creating a stable cell line (op. 22). Not displayed are additional operations required for DNA sequencing for the Gibson assembly. Unlike other examples presented here, this workflow was only generated *in silico* and was not executed.

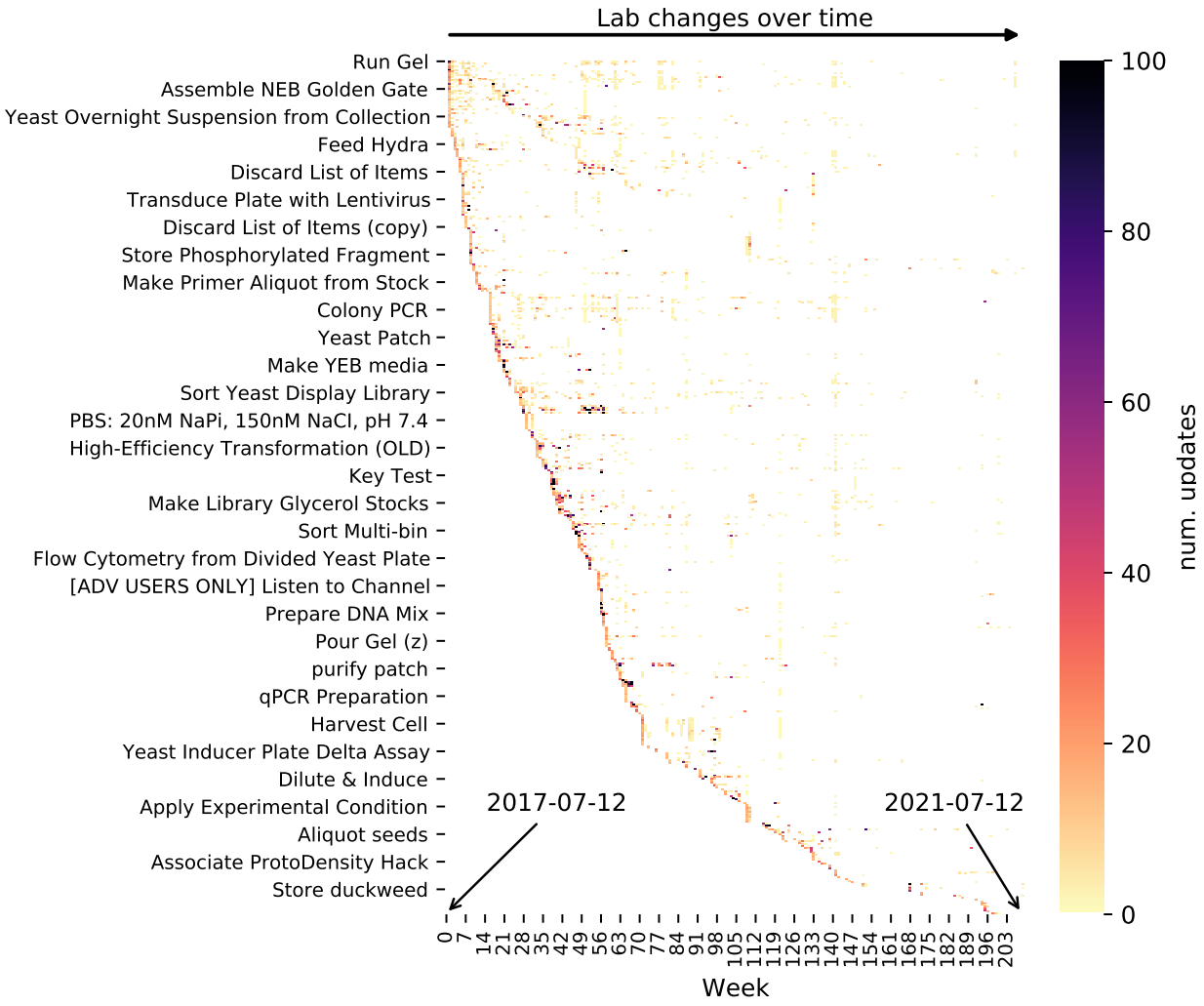


Figure 6.12: **Plot of a dynamically changing laboratory** A lab changes over time with the addition of new protocols and updating of existing protocols. Depicted is a timeline-heatmap of operation (protocol) updates of a University of Washington lab over the course of 4 years. On the y-axis are 495 operations with some of the operation names displayed. The x-axis represents the week since implementing Aquarium (starting in 2017-07-12). An update consists of a change in the operation code or operation definition. For visual purposes, the number of updates is clipped to a maximum of 100 updates but the actual range is [0, 445]. Visually, operations are sorted by their creation date so that the left boundary can be interpreted as the introduction or creation of protocols in the lab. Over just 4 years 36905 protocol updates have been recorded.

6.12 Extended Data and Figures

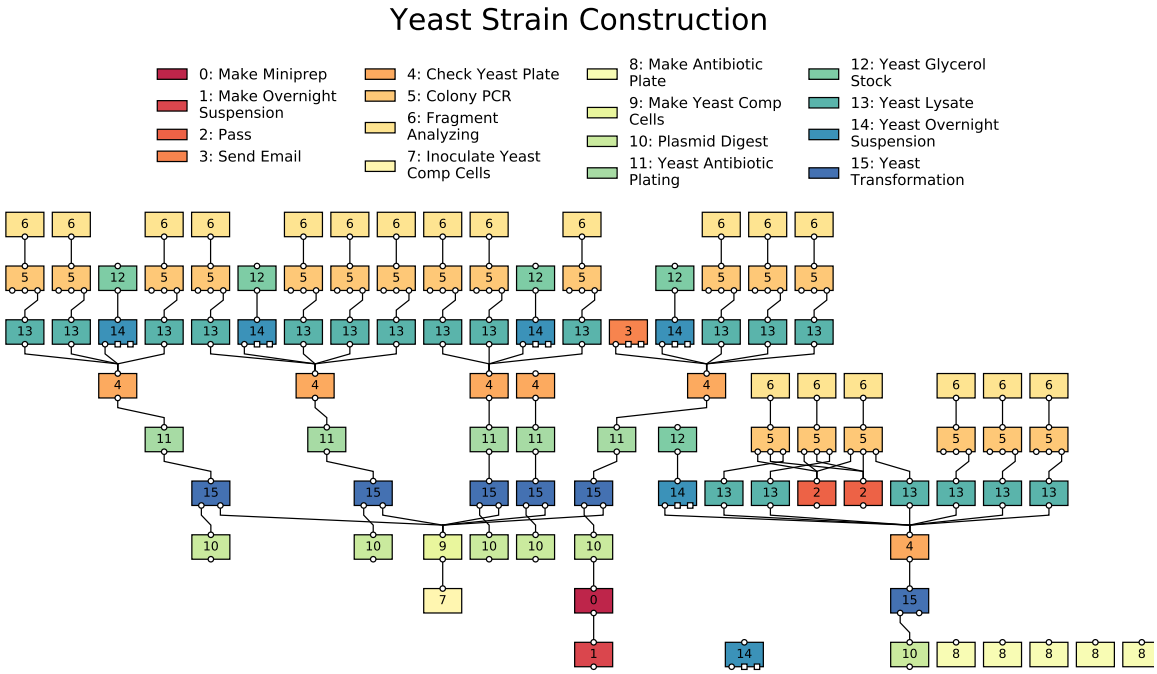


Figure 6.13: **Generated workflow for yeast strain construction**

Box 6.12: Biological Manufacturing File (BMF) for DNA


```
{
  "__bmf__": true,
  "config": {
    "lab": {
      "url": "http://0.0.0.0",
      "name": "production"
    },
    "constraints": {}
  },
  "goals": [
    {
      "plan_id": "dna_assembly",
      "output": {
        "sample": "pGPD-yeVenus-tCYC1",
        "object_type": "Plasmid Glycerol Stock"
      },
      "dependencies": [
        ["frag1", "pGPD-yeVenus-tCYC1"],
        ["frag2", "pGPD-yeVenus-tCYC1"],
        ["primer1_f", "frag113434"],
        ["primer2_r", "frag113434"],
        "...",
      ]
    }
  ],
  "definitions": [
    {
      "name": "pGPD-yeVenus-tCYC1",
      "sample_type": "Plasmid",
      "fields": {
        "sequence": "AGGGCATGGATCAGGATGATAGAAAAGAGATGG...",
        "cyclic": true
      }
    },
    {
      "name": "primer1_f",
      "sample_type": "Primer",
      "fields": {
        "sequence": "AGGAGGCTAGTATATAGGAGGGAGGCCCAA",
      }
    },
    {
      "name": "frag113434",
      "sample_type": "Fragment",
      "fields": {
        "sequence": "GGGCGAGCCTAGCATTTATAGCATA...",
      }
    },
    {
      "name": "gblock5234",
      "sample_type": "Fragment",
      "fields": {
        "sequence": "GGGCGAGCCTAGCATTTATAGCATA...",
        "synthesized": true
      }
    },
    "...",
  ]
}
```

Chapter 7

ALGORITHM DETAILS FOR AUTOMATED PROCESS PLANNING

In this chapter I describe additional details on the algorithms used in Terrarium and DASi.

7.1 Algorithms I: General Computer-Aided Process Planning for Laboratory Operations

Laboratory models and definitions A digital twin in manufacturing refers to the use of a computerized model of a manufacturing factory for the purposes of simulating and optimizing either the planning or manufacturing processes [195–199]. Essential to the digital twin concept, is an automatic coupling between objects in the physical world and those digitally represented (Fig. 7.1). To create a digital twin of the laboratory, we leverage

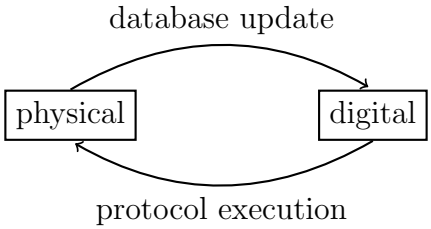


Figure 7.1: The digital twin concept

our group's previously published software Aquarium [128] to manage inventory and protocol execution such that, protocol execution results in updates to the digital representation of lab inventory. This allows the use of algorithms for automated data collection, workflow planning and protocol execution in Aquarium. In order to perform automated planning using Aquarium, we introduce some core mathematical definitions of laboratory work, which we borrow and modify from other manufacturing literature [174]. We use these definitions to outline algorithms for computer-aided process planning below.

In other manufacturing disciplines, machine products and parts are grouped by feature similarity (manufacturability, shape, size, material, etc.) to aid in process planning (referred to as Group Technology (GT) [200]). We take a similar approach in defining different types of objects used in a laboratory. Every type of inventory in the laboratory is defined by a 3-part tuple (t, s, c) representing "sample type", "sample", and "container". As an example, a 1.5mL eppendorf tube of a pUC19-GFP plasmid may be defined as ($t =$ "plasmid", $s =$

"pUC19-GFP", $c =$ "1.5mL eppendorf tube"). Inventory can be implemented using database tables but is not discussed in this document[1].

Definition 7.1.1 (Laboratory State)**.** A laboratory 'state' is the set $\mathbb{S} \in \mathbb{Z}^{+3}$ representing all available inventory in the laboratory. Inventory consists of tuples (t, s, c) representing "sample type", "sample", and "container". We also define a set of query functions $q(\mathbb{S}) = s; s \subseteq \mathbb{S}$ that selects inventory. For example, we denote functions as in $q_{t=\text{yeast}}(\mathbb{S})$ to select all inventory with the "yeast" sample type.

Definition 7.1.2 (Laboratory Operation)**.** A laboratory 'operation' is a mapping $\phi : \mathbb{S} \mapsto \mathbb{S}$ that alters the laboratory state (by adding or removing inventory tuples (t, s, c)). We specify a particular state at a given time t as in S_t. Laboratory state transitions can be denoted using a ordered sequence of operations $(\phi_1, \phi_2, \ldots, \phi_n)$ as in:[2]:

$$S_0 \xrightarrow{\phi_1} S_1 \xrightarrow{\phi_2} S_2 \xrightarrow{\phi_3} \ldots \xrightarrow{n} S_n$$

Definition 7.1.3 (Operation type)**.** We further specific operations ϕ has having an 'operation type' defined as ϕ^{type}. Operation types refer to the type of protocol the operation accomplishes, such as a "polymerase chain reaction" or a "yeast transformation". For brevity, we refer to a specific operation's type by ϕ^{type} instead of involving a new function $f(\phi_i) = \phi_i^{\text{type}}$ that retrieves the type from the operation.

Definition 7.1.4 (Laboratory workflow)**.** A 'workflow' is a directed acyclic graph $W = (V, E)$ where V are the set of nodes $V = (v_1, v_2, ..., v_n)$ corresponding to operations $(\phi_1, \phi_2, ..., \phi_n)$ and E are the edges connecting operations together $E = (e_1, e_2, ..., e_m)$.

---

[1]Inventory tuples (t, s, c) can also be presented using unique numbers (such as $(100, 131, 3410)$) and implemented using database tables, typically with further information (such as "pUC19-GFP (id=131) contains an Ampicillin resistance marker"), as our group has shown in our prior publication[128]. This is typically referred to as a Laboratory Information Management System (LIMS), of which there are many.

[2]Practically speaking for Aquarium[128], there is also a mapping between each operation ϕ to computer code that generates a list of instructions that can be interpreted by a human or robot to physically execute actions in the laboratory. However, the nature of these instructions is not relevant for the mathematical definitions. We assume the instructions, if executed faithfully, result in the given alteration of lab state S.

Definition 7.1.5 (Operation to operation connectivity)**.** We define the connectivity function that determine whether connecting two operations together is a valid connection:

$$\Xi(\phi_i^{\text{type}}, \phi_j^{\text{type}}) = \begin{cases} 1, & \text{if valid} \\ 0, & \text{otherwise} \end{cases} \tag{7.1}$$

A workflow is valid if every edge in the workflow is valid $\Xi(\phi_i^{\text{type}}, \phi_j^{\text{type}}) = 1, \forall(\phi_i, \phi_j) \in E$. The connectivity function c captures the relationship between different operation types.

Definition 7.1.6 (Operation to inventory connectivity)**.** We define a connectivity function that determines whether a inventory (s, t, c) is a valid input for a given operation ϕ

$$\xi\big((t_i, s_i, c_i), \phi_j^{\text{type}}\big) = \begin{cases} 1, & \text{if valid} \\ 0, & \text{otherwise} \end{cases} \tag{7.2}$$

This function broadly captures the input properties of an operation type. For example, an *E. coli* strain is not a valid input to an operation that performs a yeast transformation, while a linear DNA fragment is a valid input to a yeast transformation.

Data driven estimation of the connectivity function (Ξ) Creating and encoding new operations can be a time-consuming and laborious process. From a programming standpoint, it is useful to define operations in such a way that they are modular and can be reused in different contexts. However as the number of available modular operations grows in a laboratory, the number of available ways to connect operations into workflows grows exponentially. As an example, the UW-Seattle laboratory that implements the Aquarium system includes 79 modular operations that cover molecular biology and yeast strain construction operations that can be connected in 22,300 different ways; that is the full network of possible connected operations contains over 22,000 edges. In order to generate goal-based workflows, the CAPP algorithm must filter through this large network to find valid workflows. Naively choosing workflows from the complete network often generates workflows that are often not sensible. The primary cause of this is that defined operations often do not completely capture how

they are intended to be used in a workflow; this is a practical limitation to the complexity and diversity of protocols available to a laboratory. To circumvent this, the CAPP algorithm employs a data-driven approach to generating workflows. Briefly, the algorithm collects data from a set of previously run workflows, $D = (W_0, W_1, ..., W_n)$. Every operation in every workflow in D is used to calculate the probability of connecting of connecting two operation types in series $p(\phi_i^{\text{type}}, \phi_j^{\text{type}})$ by simply counting all $(\phi_i^{\text{type}}, \phi_j^{\text{type}})$ edges and dividing by the total number of edges $(\cdot, \phi_j^{\text{type}})$.

$$p(e_{ij}) = \frac{\sum_W \sum_{(\phi_i, \phi_j) \in W} (\phi_i^{\text{type}}, \phi_j^{\text{type}})}{\sum_W \sum_{\phi_j} (\cdot, \phi_j^{\text{type}})} \tag{7.3}$$

For downstream planning in later algorithms, the probability $p(e_{ij})$ is used in the calculation of an empirically derived cost function empirically derived cost function:

$$c(p) = \frac{1}{p^x + \epsilon} + \beta \tag{7.4}$$

where p is the probability, ϵ is a small number controlling the maximum penalty when $p = 0$, x is the penalty scaling factor for rare connections and β is the minimum edge cost for an edge with $p = 1$.

Algorithm for finding minimum spanning tree The objective of the computer-aided process planner (CAPP) is to find a valid workflow that generates a provided goal sample, as provided by the Biological Manufacturing File (BMF). The general procedure to do this is to generate a large directed graph representing all possible valid workflows and find the minimum spanning tree between any goal samples and available inventory. The challenge with doing this is that (i) all operations in a valid workflow must follow strictly the sample dependencies outlined in the BMF and (ii) every operation in the final workflow must have all of its inputs satisfied by either inventory or an incoming edge from another operation.

To begin, we construct our graph using the biological dependencies outlined in the BMF against the network of all possible operations that could be connected in a laboratory. We define the sample dependencies from the BMF as directed graph D and the network

of all possible operations in the lab as directed graph O. D contains inventory labels $[(s_0, t_0, 0), (s_1, t_1, 0), ...]$ corresponding to "sample" and "sample type" definitions for each biological sample, as defined in Def. 7.1.1. Unlike inventory defined in Def. 7.1.1, samples from the BMF have no physical containers, and so the container is set to 0 in the last position of each tuple. To create O, we iterate through all connectable pairs of operation type using the connection function $\Xi(\phi_i^{\text{type}}, \phi_j^{\text{type}}) = 1$ as a filter. From there we define a new noncommunative graph product, $\Diamond$ defined as follows:

Definition 7.1.7 ($\Diamond$)**.** The graph product, $H = G_2 \Diamond G_2$, creates a new vertex set of H by the cartesian product $V(G_1) \times V(G_2)$. These vertices can be enumerated as $(a_1, a_2), (b_1, b_2), ...$ and so one where where a_1 , b_1, ... are vertex labels of G_1 and a_2 , b_2, ...are vertex labels of G_1. Vertices of H, $(a_1, a_2), (b_1, b_2)$ are adjacent if and only if $(a_1 = b_1 \vee a_1 \sim b_1) \wedge a_2 \sim b_2$, where "$i \sim$ j" means i is a parent of j.

Applying $\mathcal{G} = S \Diamond O$ creates new vertices that correspond to assigning biological samples to operations. The adjacency condition ensures that ordering present in O is also maintained in $\mathcal{G}$, while a less stringent dependency condition for S is maintained. This allows workflows that utilize a particular sample for a path through the operation network. To finalize the graph, the weight defined in Eq. 7.3 is applied to all edges.

To find a valid workflow through H, we classify some terminal nodes as "start" nodes or "end" nodes. For "start" nodes identifying assign all available inventory samples (tuples of (t, s, c) Def. 7.1.1) to all operations in H. We iterate through all pairs of inventory (t, s, c) and all operations ϕ and apply the connectivity function $\xi((t_i, s_i, c_i), \phi_j^{\text{type}})$, any valid inventory is assigned to the corresponding node in H. Likewise, for "end" nodes we take all goal sequences provided by the BMF (nodes in D) and assign them to nodes in H. This defines the problem in terms of a Steiner tree problem, whose solution is to find a minimum spanning tree (MST) between "start" and "end" terminal nodes. We use the applied edge weight defined in eq. 7.3 to guide the solutions towards valid workflow that are similar to workflows previously run. We apply an additional penalty (50,000) to solutions that contain

invalid operations, or those operations that are missing satisfied inputs.

Creation of BMF from design files for yeast To create BMF files for yeast strain construction, we apply a special procedure called the *design decomposion*. Design decomposition refers to the procedure of taking a top-level organism design and creating a hierarchical representation of that design for use in down-stream planning. In this study, we define cell designs as a base strain and a set of DNA sequences that are to be integrated into the base strain's genome. For yeast, we use homology-based integration strategy and so for each integrant, we select from a list of standardized integration plasmids (with standard genomic loci and homology regions) and assign each integrant to its own unique integration plamid. We then decompose the set of plasmids into a partially ordered set ("poset"). This can be best thought of as drawing a 'path' through a Hasse diagram (Fig. 7.2 a diagram of all possible subsets) such that each edge in the diagram represents an integration event: In

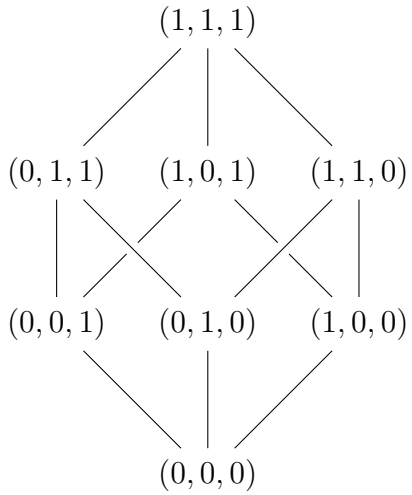


Figure 7.2: **Yeast transformations as a Hasse diagram** Each position in the set represent a separate integrative plasmid. A path through from bottom "(0, 0, 0)" to top indicates a valid path to produce a given yeast strain "(1, 1, 1)"

this study, we used a specific decomposition procedure appropriate for general cell transformation or genomic integration procedures. However, how exactly to design DNA such

that integration or transformation events occur is highly application and organism-specific. For homology-based integration in yeast, for example, there must be sufficient homology between DNA sequences flanking the integrant and the host genome for successful integration to occur. Furthermore, appropriate auxotrophic or antibiotic markers often are included in the design so that successful transformants may be selected. Other strategies may include using multi-locus CRISPR integration as previously reported [201, 202]. Given the variety of ways to perform transformations and integrations, we consider this type of design work to be outside the scope of a general lab-based computer-aided process planner presented in this study, leaving this type of design work to more capable and appropriate software. Rather, the purpose of the CAPP in this study is to automate workflows *following the DNA design* process.

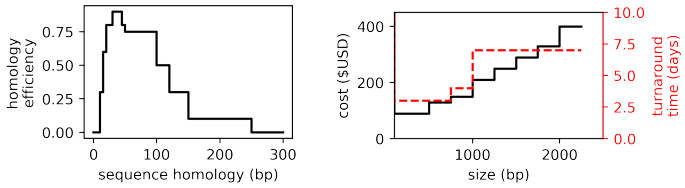
7.2 Algorithms II: DNA Assembly

Molecular biology reactions In brief, the objective of the molecular biology planning algorithm is to construct a least-costly workflow that, if executed, would clone the desired goal DNA sequence(s). The workflow consists of a series of molecular biology reactions (called here *operations*) that act on one or more DNA species (e.g. primers, linear fragments, cyclic plasmids) to produce other DNA species. For example, a goal sequence might entail adding a GFP expression cassette to the pUC19 cloning vector [203]. If provided with the end-sequence (that is the desired pUC19 + GFP cassette sequence), is possible to compute backward to all possible molecular biology reactions that could produce the provided

To automated molecular biology planning, a family of algorithms is used to construct possible cloning paths for provided goal DNA sequences and then compute the least-costly path(s) given cost and efficiency estimations, as well as the availability of current inventory. We represent molecular biology reactions as the set of possible reactions $\mathcal{R}$ such that each member of the reaction set $r_0, r_1, ... \in \mathcal{R}$ if a function $r_0(s_1, ...) \rightarrow s_2$ that transforms a set of sequences s_i to a new set of sequences s_{i+1}. For example a PCR reaction r_{PCR} take three sequences (one template and two primers) to produce a linear amplified sequence $\text{REACTION}_{\text{PCR}}(s_0, p_0, p_1) \rightarrow s_2$. Additionally, each reaction also has a cost c and efficiency $e \in [0, 1]$ function associated with it representing the materials costs associated with the reaction and the estimated probability of a successful reaction. For simplicity, we define a new reaction function $r(\cdot) \rightarrow (s, c, e)$ that returns the new sequence(s), cost and efficiency. We define the following reaction functions

$$
\begin{aligned}
&r_{\text{PCR}}(s_0, s_1, s_2) \rightarrow s_3, c, e \\
&r_{\text{primer annealing}}(s_0, s_1) \rightarrow s_3, c, e \\
&r_{\text{homology assembly}}(s_0, s_1) \rightarrow s_2, c, e \\
&r_{\text{synthesis}}(s_0) \rightarrow s_0, c, e \\
&r_{\text{primer order}}(s_0) \rightarrow s_0, c, e
\end{aligned}
\tag{7.5}
$$

The c, e returned depends on the sequence identity of the provided sequences to the function. To represent an invalid molecular reaction, the reaction function merely returns an $e = 0$. For example, $r_{\text{primer order}}(s_0) \rightarrow s_0, c = 25, e = 1$ for small sequences below 50bp (a valid primer order), while it would yield $e = 0$ for sequence of length 1kb (too long for primers). Similarly, for DNA synthesis, a combination of length and sequence complexity is used to adjust the efficiencies. For PCRs, efficiency is based on length, for example, lengths beyond 4kb begin to have a lower efficiency that scales with its length. A majority of these cost and efficiency functions are trivial and uninteresting. However, below are the junction homology and DNA synthesis costs and efficiency plots which often have the largest effect on overall DNA assembly cost (Fig. 7.3).



(a) junction efficiency (b) DNA synthesis costs

Figure 7.3: **Homology and synthesis costs for DASi** Plots of costs associated with default parameters for DNA sub-planning. The homology efficiency function is hand-crafted to be most efficient 20-60 bp. Synthesis complexity is estimated using windowed GC% content and by estimating the number of inverted repeats and direct repeats. Synthesis costs are collated from DNA synthesis vendors like IDT and Twist. These parameters can be adjusted and customized in DASi's configuration file.

Creating cloning paths DASi plans homology-based DNA assemblies. To create a cloning path, we create a direct graph of molecules $A = (V, U)$ $V \in (s_0, s_1, ...)$ and assign all possible reactions $(r_0, ...)$ between all sets of sequences and filter by reactions that result in $e > 0$. A vast majority of evaluated reactions result in $e = 0$ and so in the actual code implementation, this procedure is performed much more efficiently than the simplistic

brute-force method described here. Practically speaking, most goal sequences are between 10^3 and 10^4 bp long, and so the number of every possible candidate molecule that could produce an astronomical number of evaluations; for a 5kb plasmid, evaluating just a two molecule assembly would by 5000^4 (evaluating every basepair position for each end of the two molecules), with a vast majority of these resulting in inefficient or non-functional DNA assemblies. To get a list of candidate molecules, the BLAST+ algorithm [204] is used against available inventory sequences. The algorithm begins by calculating perfect (no-mismatch or misalignment) local sequence alignment between goal sequences S_{goal} and available inventory sequences $S_{\text{inventory}}$. The algorithm uses BLAST+ to generate a new set of sequences S_{align}, that is we apply the alignment function $f_{\text{BLAST+}}(S_{\text{goal}}, S_{\text{inventory}}) \rightarrow S_{\text{align}}$. We then perform an *expansion* function $f_{\text{EXPAND}}(S_{\text{align}}) \rightarrow S_{\text{candidate}}$ to produce a set of candidate molecules. The expansion function f_{EXPAND} takes all the 5' and 3' locations of the alignments relative to the S_{GOAL} and mixes these locations to produce new candidate molecules. Candidate molecules that would be impossible to produce from the $S_{\text{inventory}}$ are then filtered out. What remains is a list of molecular products that could produce the goal sequence. For areas of the goal sequence that do not contain alignments, DNA synthesis can be employed to *span the gap*.

Finding assembly by shortest cycle Once a list of candidate molecules is generated $S_{\text{candidate}} \in (s_0, s_1, ...)$ reactions can be matches to all subsets of molecules as described above. Once reactions are matched, reactions are simulated to produce a new group of molecules $S_{\text{reactants}}$. Candidate molecules are updated $S_{\text{candidate}} \leftarrow \cup(S_{\text{candidate}}, S_{\text{reactants}})$. We then create nodes and edges between two molecules s_0, s_1 if the efficiency of the reaction $r_{\text{homology assembly}}$ is greater than 0. We assign cost and efficiency to nodes for molecules that were produced in $S_{\text{reactants}}$ and use the c, e from $r_{\text{homology assembly}}$ to assign c, e to the edges. The score of a path is the sum of all costs along nodes and edges multiple by the product of

all efficiencies along the nodes and edges:

$$c_{\text{path}} = \prod_{i=1}^{n-1} e_{i,i+1} e_i \sum_{i=1}^{n-1} c_{i,i+1} + c_i \tag{7.6}$$

where $e_{i,i+1}$ is the efficiency of the i-th edge in the path and e_i is the efficiency of the i-th node in the path; $c_{i,i+1}$ is the cost of the i-th edge in the path and c_i is the efficiency of the i-th node in the path.

Handling cyclic alignments The alignment procedure takes into careful account whether inventory sequences or goal sequences are cyclic; the BLAST+ algorithm does not handle cyclic sequences. To handle this, DASi concatenates the cyclic sequences and performs the BLAST alignments, and then carefully adjusts the alignment back to cyclic coordinates.

Sequence complexity calculations To estimate sequence complexity, DASi employs a highly optimized sequence complexity function using 2D convolutions. Briefly, a sequence of size n is one hot encoded into $n \times 4$ based on base pair identity on each position. A small kernel (e.g. 20) is used to scan sequences. Sequence repeats are easily identified by counting unique numbers in the resulting array. Hairpin can be identified by repeating the process with a reverse complement sequence and counting the number of collisions between the two convolutions. This in combination with rolling window statistics on GC content, is used in a function to estimate sequence complexity. Because this calculation is very fast, it can be used to create optimal synthesis partitions for very long synthesized molecules in a way similar to the BOOST algorithm described by the Hillson et. al [151]. Code for sequence complexity calculations can be found on a public Github repo.

Primer design For all reactions r_{PCR} that require new primers, Primer3 is employed Ṗrimers are first design to efficiently align to the template sequence and then, if necessary, expanded on the 5' end for homology-based assembly.

Extending sub planning to other cloning strategies Currently, our software implementation only implements homology based DNA assembly. Homology-based cloning, such as Gibson Assembly, allows is flexible as it results in a scar-less DNA sequence [205]. However, other cloning methods, like restriction-based cloning (e.g. RSII/Golden-Gate [206]) are often more appropriate for high-throughput cloning. It is possible to extend our software implementation to include restriction-based cloning, such as RSII cloning (like Golden Gate [206]). The restriction cloning implementation, arguably, would be similar to our current implementation; instead of identifying junctions based on sequence identity, restriction sites would be identified and only those sequences with complementary overhangs would be allowed to form an assembly junction.

Also of note, is that the CAPP software implementation does not preclude the use of other DNA cloning automation tools; all that is required for sub planning DNA assembly is generating a DNA sequence decomposition graph (hierarchical representation of the DNA assembly process). The J5 software tool, for example, outputs CSVs of DNA fragments, plasmids, and primers, from which a sample decomposition graph can be created. However, our implementation is the only known implementation that can effectively utilize lab inventory during the sub planning process.

Raw DASi output file The raw output of DASi is a large JSON file containing all information about molecular reactions, sequences, alignments, costs, and efficiencies. This file can be parsed to a much simpler BMF or used to produce a graphical PDF report. Below a snapshot of such a file can be found in Box 7.2.

Box 7.2: Abbreviated DASi output JSON file


```
{
 "metadata": {
  "program": "dasi",
  "version": "0.2.1",
  "execution_trace": {
    "compile": {
      "__name__": "compile",
      "__spec__": "FullArgSpec(args=['self', 'post_process_kwargs'], varargs=None,
          varkw=None, defaults=(None,), kwonlyargs=[], kwonlydefaults=None,
          annotations={'post_process_kwargs': typing.Dict})",
      "start": "2021-07-23 14:23:46.285666",
      "end": "2021-07-23 14:23:50.127993",
      "args": {
        "post_process_kwargs": null
      },
      "algorithm": "standard optimization"
    },
    "optimize": { ... }
    }
}
*** SECTION OMITTED ***
"designs": {
  "ff9411eb-8e3d-4554-a37c-f3e522c0e840": {
    "compiled": true,
    "run": true,
    "success": true,
    "assemblies": [
      {
        "cost": {
          "material cost": 289.75,
          "assembly efficiency": 0.46,
          "max synthesis complexity": 2.22
        },
        "summary": [
          {
            "reaction_index": 2,
*** SECTION OMITTED ***
"molecules": [
  {
    "__name__": "PLASMID",
    "__index__": 0,
    "__type__": "molecule",
    "__meta__": {},
    "sequence": {
      "bases": "AATGCAGCGAGGTCGGTCTACGTCG...",
      "length": 7009,
      "name": "<unknown name>",
      "id": "<unknown id>",
      "customFields": {"topology": "circular"},
      "isCircular": true
*** SECTION OMITTED ***
"reactions": [
  {
    "__name__": "Assembly",
    "__index__": 0,
    "__type__": "reaction",
    "used_in_assemblies": [
      {"design_key": "ff9411eb-8e3d-4554-a37c-f3e522c0e840", "assembly": 0}
    ],
    "inputs": [
      19,
*** SECTION OMITTED ***
}
```

Chapter 8

APPLICATIONS OF AUTOMATION TO GLOBAL HEALTH

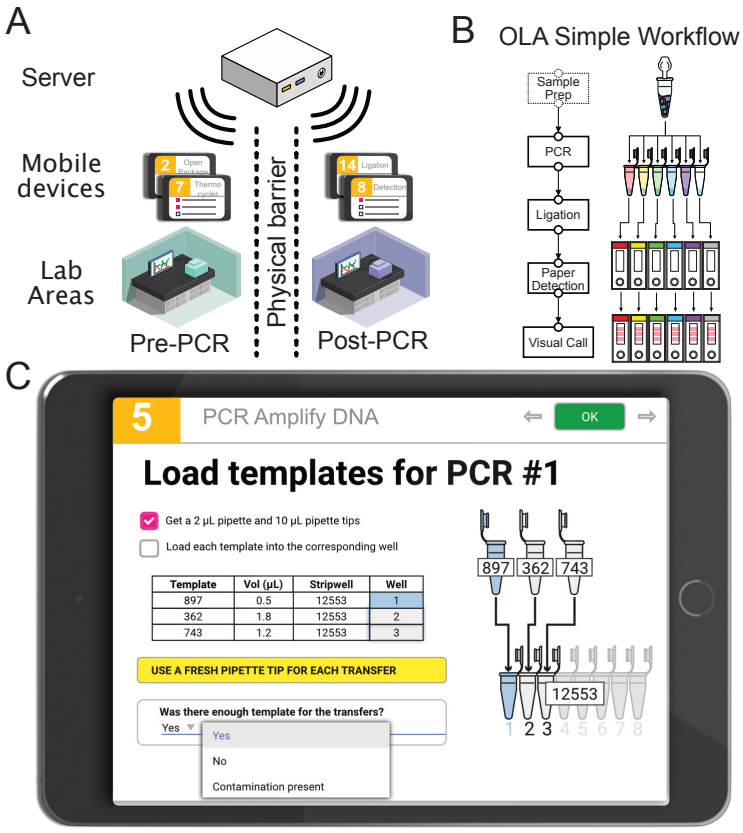


Figure 8.1: OLA-Simple lateral flow assay for diagnosing HIV drug-resistance. Protocol execution is managed by an Aquarium-software powered tablet.

In this chapter, I describe applications of Aquarium to the field of global health. The software-assisted guidance that Aquarium features can provide a massive benefit in the clinical setting, allowing workers, nurses, and lab personnel to perform complex lab procedures with little-to-no experience. The work I performed in this chapter is disseminated in several

publications [119, 207, 208]. The Kenya study is currently in pre-print [208], as we negotiate a journal with our co-authors. N.P. performed UW-Seattle experiments. N.P. invented OLA-Simple and wrote the paper description of the protocol. N.P. and I designed the UW study to characterize OLA-Simple. I wrote the protocol code and workflow for the UW-Seattle study. N.P. oversaw the UW-Seattle, with remote guidance from myself. I wrote the protocol code to automate clinical procedures, the mobile application, and the mobile lab setup. I.B. and I traveled to Nairobi, Kenya to perform pilot study.

Author	**Contribution**
Justin Vrana	Protocol code, application development, Kenya pilot study, data analysis, code and co-management of UW-Seattle pilot study
Nuttada Panpradist	Inventor of OLA-Simple platform, UW-Seattle pilot study, data analysis, experiment design
Ingrid Beck (Seattle Children's)	Kenya pilot study, data analysis, experiment design

8.1 Pilot study of automating HIV diagnostic tests in in Nairobi, Kenya

8.1.1 Abstract

Usability is an overlooked aspect of implementing lab-based assays, particularly novel assays in low-resource settings. Unclear instructions can lead to irreproducible test results and patient harm. To address these issues, we developed a software application based on "Aquarium", a laboratory-operating system that provides step-by-step digital interactive instructions, protocol management, and sample tracking. Aquarium runs on a computer tablet and was paired with a near point-of-care HIV drug resistance (HIVDR) test, "OLA-Simple", that detects mutations associated with virologic failure. We hypothesized that Aquarium would successfully guide untrained users through the multi-step laboratory protocol with little supervision. A feasibility study was conducted in a laboratory at Coptic Hope Center in Nairobi, Kenya. To evaluate the training by Aquarium software, twelve volunteers who were unfamiliar with the kit performed the test on blinded samples (2 blood specimens; 5 codons/sample). Steps guided by Aquarium included: CD4+ T-Cell separation, PCR, ligation, detection, and interpretation of test results. Participants filled out a short survey regarding their demographics and experience with the software and kit. 12/12 technicians had no prior experience performing CD4+ separation and 7/12 had no experience performing laboratory-based molecular assays. 12/12 isolated CD4+ T cells from whole blood with yields comparable to isolations performed by trained personnel. OLA-Simple workflow was completed by all, with correct visual and software interpretation of 90% (108/120) and 98% (111/120) codons, respectively. In the surveys, participants favorably assessed the use of software guidance. Aquarium could increase the accessibility of laboratory assays in low-resource settings and potentially standardize the implementation of clinical laboratory tests.

8.1.2 Clinic Automation in Low Resource Settings

In resource-rich communities, automation drastically improves the daily operation of clinical laboratories [179, 209, 210]. Patient samples can be quickly shipped to centralized labora-

tories for batch-processing using highly efficient workflows that generate high-quality results while reducing costs and turnaround time [211, 212]. However, total automation is ill-suited for low-resource settings for many reasons. First, shipping of samples to centralized laboratories can take $\geq$ 10 days [213], which undermines the benefits of fast turnaround test results from automated workflows. Second, in small communities, the demand for a clinical assay may be low, requiring a longer waiting period to receive enough samples to complete a full batch. Finally, automation is often used in conjunction with high-throughput robotic equipment that is cost-prohibitive for small laboratories. In low-resource settings, high-quality and fast laboratory results for complex assays will likely require unorthodox approaches to automation to build on low-cost equipment and be applicable for small batches of samples.

HIV infects nearly 40M people globally [214] and successful management of HIV relies on multiple laboratory tests. Recent advances include point-of-care HIV diagnosis and viral load quantification [215, 216]. Due to the complexity of HIV drug resistance (HIVDR) tests used to guide treatment regimens, they are performed in centralized, highly-equipped laboratories [217–220]. In low-resource countries with high HIV prevalence like Kenya, few laboratories have the capacity to test for HIVDR [221]. For laboratories without access to sequencers, an oligonucleotide ligation assay (OLA) has been implemented [222] but onboarding OLA required extensive training due to its complexity.

We envision the use of software to automate a simplified version of OLA that uses low-cost equipment. Recently, we developed "OLA-Simple" that simplifies the workflow using lyophilized reagents and lateral flow tests to provide visual results [119, 223, 224]. Additionally, we developed a software application based on "Aquarium" [100, 128] that employs human-in-the-loop automation to tightly integrate all the steps in OLA-Simple. Aquarium provides step-by-step interactive digital instructions, protocol management, data collection and sample tracking. In a pilot study at the University of Washington, Aquarium enabled minimally-trained students to accurately perform the OLA-Simple workflow [225]. Here, we demonstrate use of Aquarium-enabled HIVDR test in a small laboratory in Nairobi, Kenya.

8.1.3 Methods

Preparation of OLA-Simple kit OLA-Simple kit was prepared as previously described [225]. The EasySep isolation kit (STEMCELL Technologies, Washington, USA) to negatively select CD4+ T-Cells was adapted to small blood volume processing, aliquoted and packaged in foil pouches. Reagents for PCR and ligation for detection of five HIV codons (K65R, K103N, Y181C, M184, and G190A) and lateral flow strips to detect reaction products were packaged in foil pouches with desiccant. Each kit component was labeled with a unique identifier, matching the images illustrated in Aquarium instructions.

Laboratory setup at Coptic Hope Center in Nairobi, Kenya A Seattle team travelled to Nairobi in April 2018 to set-up a testing site at the Coptic Hope Center for Infectious Diseases, which is a large-scale, antiretroviral treatment site [120]. The laboratory's existing standard thermal cycler, biological safety hood, bench space, and refrigerator were utilized. To onboard the test, the team brought the OLA-Simple kits, minicentrifuge, micropipettes, scanner (CanoScan LiDE 300), and tablets (Fire HD), foot pedal, UPS battery backup and surge protector (APC 1500VA Compact), and server (Intel NuC NUC7i3BNH Mini PC/HTPC) to set up and run Aquarium. Aquarium code is publicly available [119].

Study design and participant enrollment Twelve laboratory technologists were recruited from the Coptic clinical laboratories to perform OLA-Simple following instructions provided by the Aquarium-based application. Testing was spread over six days (two techs/-day) and consisted of completing a demographic questionnaire and a 30-minute introduction of the kit principle and procedure, followed by processing and testing of two blinded blood samples. CD4+ cells were separated from 0.5 mL uninfected blood and lysed. The cell lysates were then spiked with mixtures of plasmids containing known HIV drug resistance mutations and amplified by PCR, followed by ligation of mutation-specific probes and detection of the ligated products using lateral flow strips. The lateral flow strips were scanned, and the images displayed on the tablets were used by the participants to make visual calls

and generate a report using Aquarium. Finally, the participants completed a questionnaire to give feedback on their experience with the kits and software. This study was approved by the Institutional Ethics Review Committee (IERC) of the Aga Khan University in Kenya, and Seattle Children's Research Institute's IRB.

Post-analysis of samples in Seattle The DNA yield in lysed cells obtained by Kenyan technologists was assessed in Seattle by qPCR of human beta-globin [226]. Scanned images of lateral flow strips were re-analyzed using an in-house Python script [225] to determine if automated analyses improved test accuracy.

8.1.4 Results

Participant characteristics and Laboratory setup for Aquarium-assisted OLA-Simple training The 12 participants recruited were 83% male, median age 30 years old (range 26-42), had a median of 6 (range 3-10) years of experience as a lab technologist, and most spoke English. Their education level ranged from a secondary school diploma to a Master's degree, and they had varied levels of experience working with HIV or molecular techniques (Table 8.1).

Table 8.1: Demographics of Kenya pilot study participants (N=12)

Variable	**Value**
Age, median (range), years	30 (26-42)
Gender, N (%)	
Female	2 (16.7)
Male	10 (83.3)
Primary language, N (%)	

continued

Table 8.1: Demographics of Kenya pilot study participants (N=12)

Variable	**Value**
English	4 (33.3)
Swahili	1 (8.3)
English and Swahili	7 (58.3)
Highest education level, N (%)	
Masters	1 (8.3)
Bachelors	3 (25)
Technical/Vocational training	2 (16.7)
Secondary school diploma	6 (50)
Years of experience as a lab technologist/technician, median (range)	6 (3-10)
Hours/week worked for pay in current position, median (range)	45 (40-54)
Current position in laboratory, N	
Manager	2
Technologist[†]	4
Technician[†]	6
Experience performing lab procedures related to HIV, N (%)	
Viral load tests	2
CD4 testing	3
ELISA test	2
Phlebotomy and blood separation	5
Experience performing DNA or RNA extraction, N (%)	
Kit (Qiagen, COBAS Ampliprep and Taqman, Abbott, M2000rt)	3
Sputum lysis for Genexpert	1

continued

Table 8.1: Demographics of Kenya pilot study participants (N=12)

Variable	**Value**
Cavidi technology	1
Experience with thermocycler, N (%)	3 (25)
Experience with PCR, N (%)	6 (50)

† role performs microbiology, hematology, biochemistry, parasitology immunology sections, CD4/8, Genexpert (TB), ELISA

The laboratory was set up with a WiFi network to run Aquarium and coordinate assay steps across a pre-PCR and post-PCR room (Fig. 8.2A). Laboratory technologists worked in pairs following the implementation workflow (Fig. 8.2B): sample preparation and PCR set-up were carried out in the pre-PCR room, while PCR, ligation and detection were conducted in the post-PCR room to minimize the potential for amplicon-carryover contamination. Each technologist processed two uninfected blood specimens and performed mutation testing on two contrived specimens with known HIV mutations following the interactive digital instructions provided by Aquarium (Fig. 8.2C-E). On average, it took seven hours for a pair of participants to complete the tasks in the workflow from introductory session to completion of surveys.

Performance of participants on OLA-Simple All 12 participants successfully isolated CD4+ cells from whole blood collected from four donors with yields within the expected range (mean±SD: 686,450±216,500 CD4+ cells/mL, as determined by qPCR) and completed genotyping of two samples using the OLA-Simple kit reagents and protocols as instructed by the Aquarium application. The two blinded DNA samples tested by all participants included wild-type-only or mixtures of mutant and wild-type at each of five HIV reverse transcrip-

tase codons tested by OLA-Simple: K65R, K103N, Y181C, M184V and G190A (total 10 codons/participant) (Fig. 8.3A). Participants correctly genotyped 70/72 (97.2%, 95% CI: 90.3-99.7%) mutant codons and 38/48 (79.2%, 95% CI: 65.0-89.5%) wild-type codons. The only two false negatives were due to Participant #6 erroneously testing Sample 1 twice and mistakenly omitting testing of Sample 2. Of 10 false-positive results, 2 were due to testing Sample 1 in place of Sample 2 by Participant #6, 1 appeared to be contamination (strong mutant signal likely from adding the wrong ligation product to that lateral flow strip) and 7 were due to light mutant background at codons K103N (n=6) and K65R (n=1). Analysis of the scanned images using our in-house image analysis software improved test accuracy to 98% (118/120, 95% CI: 94-100%) with 97% (70/72, 95% CI: 90-99%) mutant codons and 98% (47/48 95% CI: 89-100%) wild-type codons (Fig. 8.3B). The performance of assay chemistry combined with software analysis, corrected for one sample added twice yielded 100% accuracy (120/120, 95% CI: 97-100%) with 100% (72/72, 95% CI: 95-100%) mutant codons and 97% (48/48 95% CI: 93-100%) wild-type codons.

Feedback on software and kit features Qualitatively, participants enjoyed the clear instructions to perform the assay and interpretability of the results using the tablets. They strongly agreed that they understood the meaning of the bands on the strips, and that the Aquarium instructions were easy to follow. However, several participants felt the procedure was lengthy and involved too many steps (Table 8.3 summarizes the survey responses).

8.1.5 Discussion

This work presents the first use of human-in-the-loop automation to enable a resource-limited laboratory to successfully operate an HIVDR test with minimal training. We developed a custom mobile application based on Aquarium operating system that describes procedures and workflows for our OLA-Simple kits. Participating laboratory technologists operated the OLA-Simple for the first time with good recovery in sample preparation and high accuracy of HIVDR detection.

The reported turnaround time for HIVDR in resource-limited settings is 18 days [207], while our HIVDR test performed by first-time users had a turnaround time of 7 hours with 98% accuracy when detection strips were analyzed by in-house software. With repetition, it is likely that the performance of the assay would improve for both speed and accuracy. The turnaround time included introductory training and staggering work of two technologists due to space and instrument constraints and would likely be reduced to 4.5-5 hours once users become familiar with the software and the kits. A sample mix-up reduced accuracy, which could be improved with changes in the kit labeling system.

Overall technicians scored the use of software as helpful in learning to perform the assay. The time to complete the assay was judged to be lengthy, and in response we have subsequently developed new chemistries to reduce the assay wait time to 3.5 hours, including 1.5-hour DNA whole blood preparation replaced by 30-minute RNA extraction; and 2-hour PCR replaced by 1-hour reverse transcription PCR (RT-PCR). Each kit tests two samples, as this number is appropriate for the volume of weekly patient samples submitted for HIVDR testing in small laboratories in Kenya.

The installation cost of OLA-Simple is low compared to automated NGS (>$100,000 USD). The workflow uses equipment that exists in most laboratories or is relatively inexpensive to acquire. Setting up the assay and software at the Coptic Hope Center laboratory required $1,000 USD of additional equipment (scanner, microcentrifuge, vortexer, computer tablets, a server and an uninterruptible power supply).

Aquarium software has useful features for HIVDR testing such as the automatic collection of technician interactions in Aquarium's virtual laboratory notebook that can be useful for troubleshooting. Each kit item is labeled with a unique identifier that Aquarium instructions use in conjunction with corresponding pictures to avoid ambiguity and reduce the extent of in-person training needed. In addition, uniquely labeled items allow tracking of stock consumption in real-time potentially useful and timesaving for laboratory management. Importantly, Aquarium could link test results to treatment algorithms to advise clinicians, and algorithms could be changed as clinical recommendations or policies change.

Our study shows it is feasible to use Aquarium to train local personnel and onboard an HIVDR test, but it was limited to the use of contrived specimens to establish analytical performance. Future demonstration and validation of the OLA-Simple in Kenya will include processing clinical HIV-infected specimens.

Aquarium-based software enables deployment of the OLA-Simple with minimal training by lowering technical skills required to perform such test. Human-in-the-loop automation could facilitate daily operations of laboratory-based assays and increase the accuracy and assay performance in small laboratories.

8.1.6 Supplementary

Participants were asked to fill out a short survey immediately following the procedure, which included evaluating four statements on a 5-point Likert scale (strongly agree to disagree) (Table 8.2) and three open-ended qualitative questions (Table 8.3). For ordinal categorical values, modes are displayed for each statement. For open-ended questions, we present a summary of the responses describing the different points raised by all the participants.

Table 8.2: Kenya study participant responses to statements about software-guided OLA-Simple assay

Questionaire statement	**Mode[2]**
Specimen preparation: "I was able to perform these steps in less time than my usual DNA extraction	4
PCR and ligation[1]: "Using dried reagents is easier than setting up a traditional PCR reaction"	4
Detection: "I understood the meaning of the bands in the strip"	5
Kit instructions: "Instructions were easy to follow"	5

[1] Participants 11 and 12 answered NA to the PCR question, likely due to not having performed PCR prior to this training.

[2] Likert rating scale: 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, and 5=strongly agree

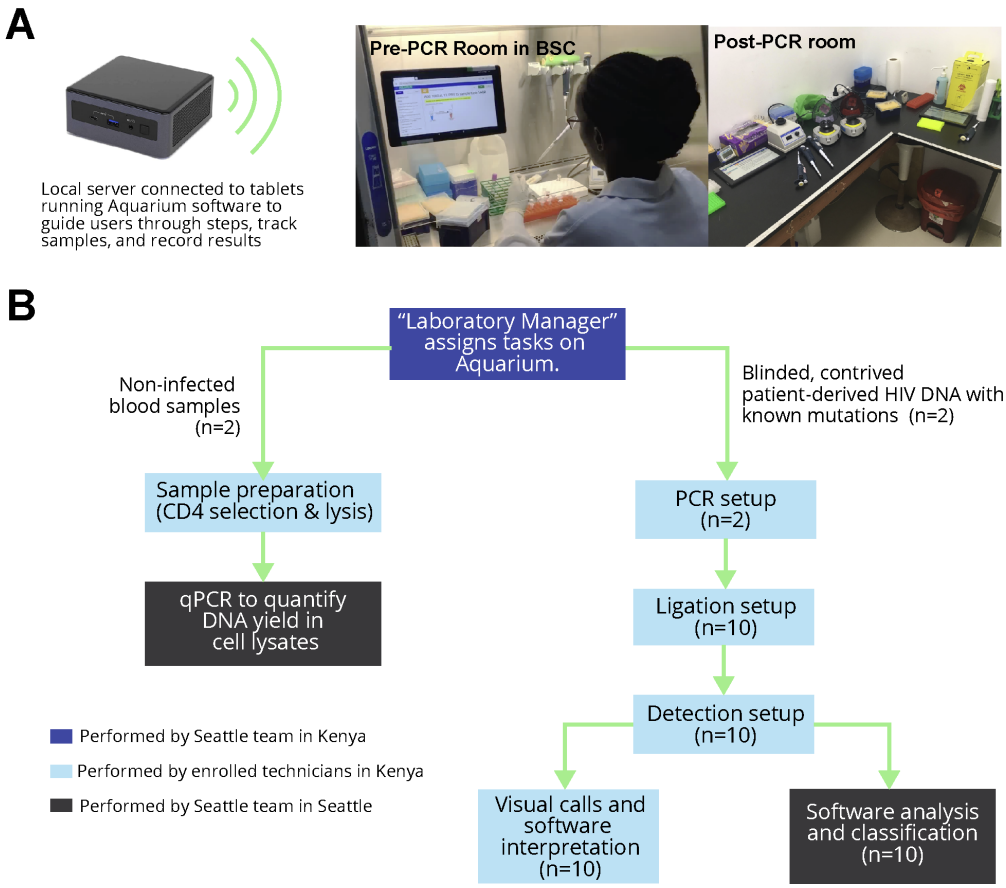


Figure 8.2: **OLA-Simple laboratory setup and workflow (A)** Laboratory setup: Tablets were connected to a local server to run the Aquarium software. The pre-PCR room contains a refrigerator, a freezer, and a class II biosafety cabinet (BSC). cell separation and PCR reactions were setup in the BSC. Technicians controlled the Aquarium software using a wireless foot pedal (not shown in the picture) while performing preparation. The post-PCR room had two designated bench areas for ligation and detection separately. This room also contained a thermal cycler and photoscanner. **(B)** A laboratory manager assigns tasks for the technicians to perform through the Aquarium software. Sample preparation operations were separated from the amplification, ligcation, detection, and interpretation modules. Documentation, associated protocol code, and examples of complete runs are publicly available at https://github.com/OLA-Simple/Papers-Vrana-Panpradist-et-al-2021

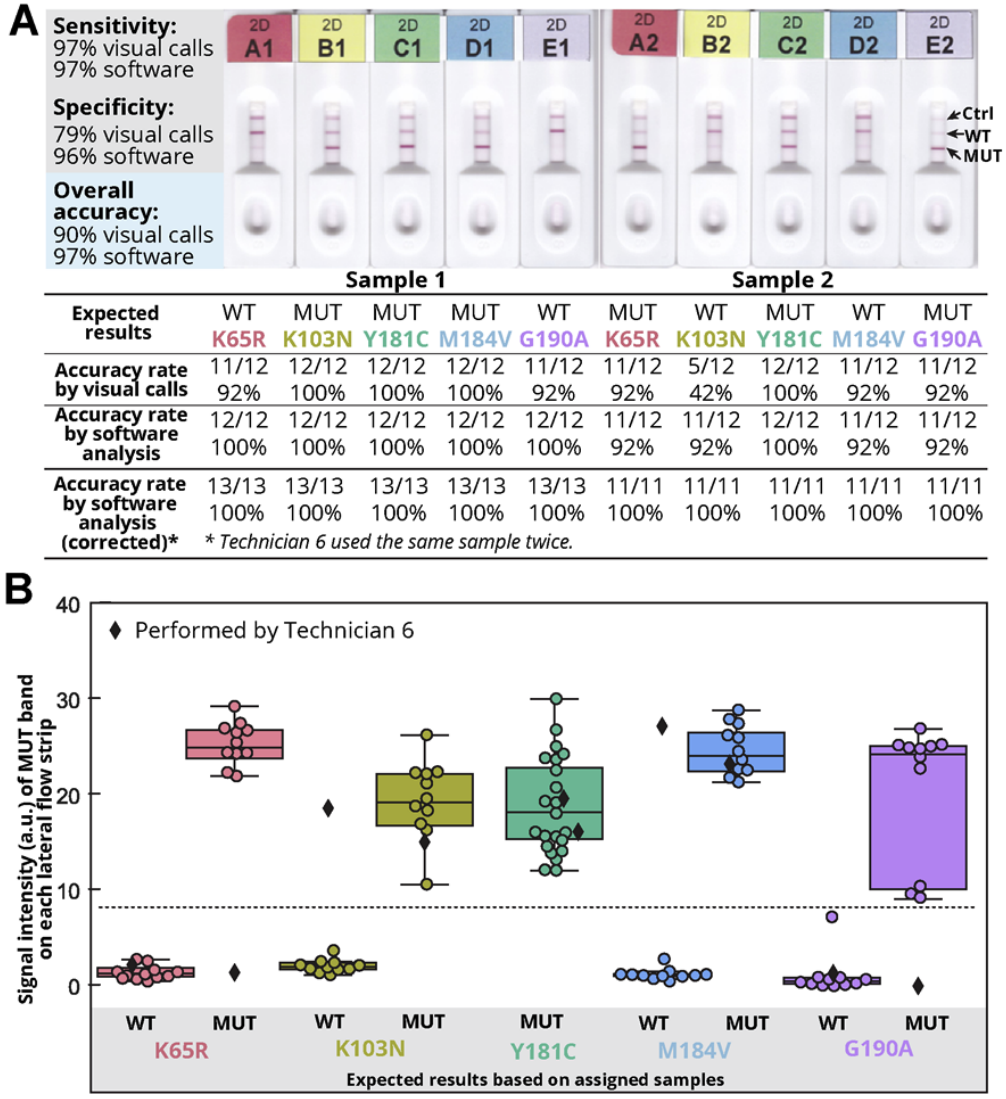


Figure 8.3: **OLA-Simple HIVDR results and interpretations** (A) Examples of scanned images of test stripes for Samples 1 and 2. Table displays the test accuracy across each codon based on (i) visual calls made by each participant and (ii) post-processing image analysis by software. Sample 1 and Sample 2 have different mutation profiles. **(B)** Mutant (MUT) signal intensity of each lateral flow strip is displayed as boxplots. Middle lines indicates mediate; top and bottom lines indicate interquartile ranges. Horizontal dashed line indicates the detection threshold for MUT signal. Both Sample 1 and Sample 2 have mutant genotype at codon Y181C. Diamons correspond to signal from the strips Sample 1 that was erroneously tested in place of Sample 2 during live pilot study.

Table 8.3: Summary of Kenya study participant responses to open-ended questions about software-guided OLA-Simple assay

What did you like best about using this kit?	**Which instruction(s) were not easy to follow? Please explain why.**	**What advice do you have for us to make kit and instructions easier to use?**
• The instructions were easy to follow • The protocol was easy to understand. • The examples in Aquarium were well labeled and made it easy to know where to add each reagent or sample • The kit is user-friendly and straightforward • The kit is easy to use compared to the plate-based OLA method • There was only one PCR step instead of the usual two for nested PCR • Detection was simplified by Aquarium • Results are generated fast and are also portable • It is easy to read the bands and interpret the results: it clearly shows the control, the wild-type and the mutant • A single sample can be easily processed without the need for batching	• The instructions and procedure were clear and easy to follow. • The instructions were Ok when following a step, but it was difficult to memorize the steps without a written protocol to follow • Some instructions did not include the specific vial or reagent in the heading, but is was in the pictorial • Sample preparation: – the multiple tubes and SOP instructions were not alphabetically arranged – includes many tubes, color coding would help to identify which to use first and avoid mix-ups – too many steps make it confusing – the process is long with many reagents and waiting time • During the ligation/detection step had challenges remembering to add reagent to the next strip • At times the tablet was not working, and we did not have a backup plan such as a written protocol with instructions.	• Enlarge the font under every heading. • Include the reagent vial ID on the headings for better clarity • The tube labels should match the SOP instruction labels. • Harmonize the sequence of adding reagents and closing of tubes to reduce multitasking and possible confusion. • Create system for how to keep track of addition of reagents to tubes • State what to expect at the end of each step, e.g. maybe a clean supernatant /deposit so one can confidently move to the next step • Look for alternatives to replace the current sample preparation method • Make the incubation time shorter • Improve turn-around time for clients. • Have print outs of the protocols as a back-up plan. • Make the kit commercially available with all the steps for laboratory technologists to follow.

Chapter 9

CONCLUSION

At its core, synthetic biology is an engineering discipline about creating new useful biological products. Synthetic biology differs from engineering disciplines in that uses biology as a *medium* for engineering. The very material that synthetic biology uses is diverse, self-replicating, ever-changing, and messy. This makes the field extremely unique, interesting and challenging. Despite all of this, the products of synthetic biology promise to be world-changing. The engineered biomolecules, cells, organisms, and viruses that synthetic biology produces can interface directly with organic systems. Since organism systems themselves are self-replicating and scalable, the products of synthetic biology represent our best solution to solve massive global problems like climate change, food scarcity, and pandemic prevention.

The uniqueness of synthetic biology necessitates the development of new engineering rules and principles. A genetic *part* in synthetic biology is not really a *part* in a traditional sense, it is a self-replicating piece of information. It can be digitally represented as a string of characters and synthesized and inserted into a cell, where it is then represented by the cell as a sequence of nucleotides that become replicated over and over again. Really, a genetic part has no physical persistence, the information is constantly recreated and destroyed repeatedly. The materials in other fields do not have this property. A transistor in a computer, for example, does not self-replicated; our computers, unfortunately, cannot grow more transistors.

In this light synthetic biology is really the discipline of engineering with information; biological information gets converted to and from atoms in such a way that these two concepts become almost indistinguishable. In no other engineering field is the connection between information and the physical world so blurred. Bits become atoms. Atoms become bits.

This blurring between the information and the physical world manifests itself in the challenges of realizing digital designs into biological samples. Constructing non-biological materials is a manner of re-arranging bulk atoms. Constructing biological materials involves rearranging atoms on a much smaller scale so that relevant biological information is retained. While these two distinctions are probably more blurred within fields like nanoengineering, it is probably safe to say that engineering with biology currently requires a much different approach than manufacturing other inorganic products. In other automated manufacturing systems, electrical signals are converted motors and movement to create physical products and there exists. In synthetic biology, we currently cannot do this simply. Tedious lab procedures are often involved only because the most efficient way to manipulate biological material is to use other biological materials.

The work I outlined in this document is my attempt to bridge the critical gap between digital biological designs and the physical realization of those designs. I presented a new set of genetic parts based on the CRISPR-Cas systems that can be used to realize circuit designs in both yeast and mammalian cells. Further, I developed software, Terrarium, that can convert digital representations of cell designs into a series of laboratory procedures that create the designs. The software borrows the spirit of other manufacturing tools and applies them to the biological realm. The software, when combined with laboratory automation tools like Aquarium, can automate the conversion of biological designs into biological samples. These types of tools can be used in fully automated design-build-test-learn pipelines to rapidly accelerate the creation of new engineered cells, biomolecules, or vaccines. Solutions to the biggest global problems like climate change, food/water scarcity, and pandemic prevent may rely on the biological products of synthetic biology. Rapid development and manufacturing of biological products will likely become key to solving these massive global problems.

BIBLIOGRAPHY

1. Church, G. M. *Regenesis : how synthetic biology will reinvent nature and ourselves /* ISBN: 0465021751 (Basic Books, 2012).

2. Ryan Georgianna, D. & Mayfield, S. P. *Exploiting diversity and synthetic biology for the production of algal biofuels* Aug. 2012. https://www.nature.com/articles/nature11479.

3. Lienert, F., Lohmueller, J. J., Garg, A. & Silver, P. A. *Synthetic biology in mammalian cells: Next generation research tools and therapeutics* Feb. 2014. www.nature.com/reviews/molcellbio.

4. El Karoui, M., Hoyos-Flight, M. & Fletcher, L. Future Trends in Synthetic Biology—A Report. *Frontiers in Bioengineering and Biotechnology* **7,** 175. ISSN: 2296-4185. https://www.frontiersin.org/article/10.3389/fbioe.2019.00175/full (Aug. 2019).

5. Cardinale, S. & Arkin, A. P. *Contextualizing context for synthetic biology - identifying causes of failure of synthetic biological systems* July 2012. https://pubmed.ncbi.nlm.nih.gov/22649052/.

6. Kwok, R. *Five hard truths for synthetic biology* Jan. 2010.

7. Liao, C. & Cai, Y. in *A Systems Theoretic Approach to Systems and Synthetic Biology II: Analysis and Design of Cellular Systems* 201–217 (Springer Netherlands, Mar. 2014). ISBN: 9789401790475. https://github.com/.

8. Appleton, E., Madsen, C., Roehner, N. & Densmore, D. Design automation in synthetic biology. *Cold Spring Harbor Perspectives in Biology* **9,** a023978. ISSN: 19430264. http://cshperspectives.cshlp.org/ (Apr. 2017).

9. Kennedy, J. F. *Bio-Design automation: Nobody said it would be easy* Aug. 2012. https://pubs.acs.org/sharingguidelines.

10. Jessop-Fabre, M. M. & Sonnenschein, N. Improving Reproducibility in Synthetic Biology. *Frontiers in Bioengineering and Biotechnology* **7,** 18. ISSN: 2296-4185. https://www.frontiersin.org/article/10.3389/fbioe.2019.00018/full (Feb. 2019).

11. Naugler, C. & Church, D. L. *Automation and artificial intelligence in the clinical laboratory* Feb. 2019. https://pubmed.ncbi.nlm.nih.gov/30922144/.

12. King, R. D. *et al.* The automation of science. *Science* **324,** 85–89. ISSN: 00368075. www.sciencemag.org/cgi/content/full/324/5923/81/DC1 (Apr. 2009).

13. Azab, A., ElMaraghy, H., Nyhuis, P., Pachow-Frauenhofer, J. & Schmidt, M. Mechanics of change: A framework to reconfigure manufacturing systems. *CIRP Journal of Manufacturing Science and Technology* **6,** 110–119. ISSN: 17555817 (Jan. 2013).

14. Cao, Y., Subramaniam, V. & Chen, R. Performance evaluation and enhancement of multistage manufacturing systems with rework loops. *Computers and Industrial Engineering* **62,** 161–176. ISSN: 03608352 (Feb. 2012).

15. Liu, N., Kim, Y. & Hwang, H. An optimal operating policy for the production system with rework. *Computers and Industrial Engineering* **56,** 874–887. ISSN: 03608352 (Apr. 2009).

16. Sarker, B. R., Jamal, A. M. & Mondal, S. Optimal batch sizing in a multi-stage production system with rework consideration. *European Journal of Operational Research* **184,** 915–929. ISSN: 03772217 (Feb. 2008).

17. Leitão, P., Barbosa, J. & Trentesaux, D. *Bio-inspired multi-agent systems for reconfigurable manufacturing systems* in *Engineering Applications of Artificial Intelligence* **25** (Pergamon, Aug. 2012), 934–944.

18. Demeester, L., Eichler, K. & Loch, C. H. Organic production systems: What the biological cell can teach us about manufacturing. *Manufacturing and Service Operations Management* **6,** 115–132. ISSN: 15234614. http://pubsonline.informs.org132.https://doi.org/10.1287/msom.1030.0033http://www.informs.org (Mar. 2004).

19. Koren, Y. in *The Global Manufacturing Revolution* 227–252 (John Wiley & Sons, Inc., Hoboken, NJ, USA, June 2010). https://onlinelibrary.wiley.com/doi/10.1002/9780470618813.ch9.

20. William E. Engelke. How to Integrate CAD/ CAM Systems. *International Journal of Production Research* **26,** 702–703. ISSN: 0020-7543. https://www.tandfonline.com/doi/abs/10.1080/00207548808947896 (Apr. 1988).

21. Elmaraghy, H. *Changeable and Reconfigurable Manufacturing Systems* (Springer London, 2009).

22. Abdi, M. R., Labib, A. W., Delavari Edalat, F. & Abdi, A. in *Integrated Reconfigurable Manufacturing Systems and Smart Value Chain* 1–13 (Springer International Publishing, Cham, 2018). http://link.springer.com/10.1007/978-3-319-76846-5_1.

23. Wang, C. *et al.* Microbial Platform for Terpenoid Production: Escherichia coli and Yeast. *Frontiers in Microbiology* **9,** 2460. ISSN: 1664-302X. https://www.frontiersin.org/article/10.3389/fmicb.2018.02460/full (Oct. 2018).

24. Madhavan, A. *et al. Synthetic biology and metabolic engineering approaches and its impact on non-conventional yeast and biofuel production* Apr. 2017.

25. Tsai, C. S., Kwak, S., Turner, T. L. & Jin, Y. S. *Yeast synthetic biology toolbox and applications for biofuel production* 2015. https://pubmed.ncbi.nlm.nih.gov/25195615/.

26. Hong, K. K. & Nielsen, J. *Metabolic engineering of Saccharomyces cerevisiae: A key cell factory platform for future biorefineries* Aug. 2012. https://pubmed.ncbi.nlm.nih.gov/22388689/.

27. Li, M. & Borodina, I. *Application of synthetic biology for production of chemicals in yeast Saccharomyces cerevisiae* 2015. https://pubmed.ncbi.nlm.nih.gov/25238571/.

28. Baltes, N. J. & Voytas, D. F. *Enabling plant synthetic biology through genome engineering* Feb. 2015. https://pubmed.ncbi.nlm.nih.gov/25496918/.

29. Wu, M. R., Jusiak, B. & Lu, T. K. *Engineering advanced cancer therapies with synthetic biology* Apr. 2019. www.nature.com/nrc.

30. Kiani, S. *et al.* CRISPR transcriptional repression devices and layered circuits in mammalian cells. *Nature Methods* **11,** 723–726. ISSN: 15487105. https://www.nature.com/articles/nmeth.2969 (May 2014).

31. Gander, M., Vrana, J., Voje, W., Carothers, J. & Klavins, E. Digital logic circuits in yeast with CRISPR-dCas9 NOR gates. *Nature Communications* **8.** ISSN: 20411723 (2017).

32. Heidenreich, M. & Zhang, F. *Applications of CRISPR-Cas systems in neuroscience* Jan. 2016. www.nature.com/nrn.

33. Huang, C. H., Lee, K. C. & Doudna, J. A. *Applications of CRISPR-Cas Enzymes in Cancer Therapeutics and Detection* July 2018. https://pubmed.ncbi.nlm.nih.gov/29937048/.

34. Knott, G. J. & Doudna, J. A. *CRISPR-Cas guides the future of genetic engineering* Aug. 2018. https://pubmed.ncbi.nlm.nih.gov/30166482/.

35. *A Handbook of Transcription Factors* (ed Hughes, T. R.) ISBN: 978-90-481-9068-3. http://link.springer.com/10.1007/978-90-481-9069-0 (Springer Netherlands, Dordrecht, 2011).

36. Westermarck, J. *Regulation of Transcription Factor Function by Targeted Protein Degradation: An Overview Focusing on p53, c-Myc, and c-Jun* 2010. https://pubmed.ncbi.nlm.nih.gov/20694659/.

37. Song, T. *et al.* Programming DNA-Based Biomolecular Reaction Networks on Cancer Cell Membranes. *Journal of the American Chemical Society* **141,** 16539–16543. ISSN: 15205126. https://pubs.acs.org/doi/abs/10.1021/jacs.9b05598 (Oct. 2019).

38. Khalil, A. S. *et al.* A synthetic biology framework for programming eukaryotic transcription functions. *Cell* **150,** 647–658. ISSN: 10974172. http://dx.doi.org/10.1016/j.cell.2012.05.045 (Aug. 2012).

39. Jensen, M. K. & Keasling, J. D. *Recent applications of synthetic biology tools for yeast metabolic engineering* 2015. https://pubmed.ncbi.nlm.nih.gov/25041737/.

40. Moore, R., Chandrahas, A. & Bleris, L. *Transcription activator-like effectors: A toolkit for synthetic biology* Oct. 2014. https://pubmed.ncbi.nlm.nih.gov/24933470/.

41. Schreiber, T., Prange, A., Hoppe, T. & Tissier, A. Split-TALE: A TALE-Based Two-Component System for Synthetic Biology Applications in Planta. *Plant Physiology* **179,** 1001–1012. ISSN: 0032-0889. https://academic.oup.com/plphys/article/179/3/1001-1012/6116442 (Mar. 2019).

42. Kittleson, J. T., Wu, G. C. & Anderson, J. C. *Successes and failures in modular genetic engineering* Aug. 2012. https://pubmed.ncbi.nlm.nih.gov/22818777/.

43. Nielsen, A. A. *et al.* Genetic circuit design automation. *Science* **352.** ISSN: 10959203. www.cellocad.org (Apr. 2016).

44. Jinek, M. *et al.* A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science* **337,** 816–821. ISSN: 10959203. http://science.sciencemag.org/ (Aug. 2012).

45. Shvets, A. A. & Kolomeisky, A. B. Mechanism of Genome Interrogation: How CRISPR RNA-Guided Cas9 Proteins Locate Specific Targets on DNA. *Biophysical Journal* **113,** 1416–1424. ISSN: 15420086. /pmc/articles/PMC5627312/%20/pmc/articles/PMC5627312/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5627312/ (Oct. 2017).

46. Karvelis, T., Young, J. K. & Siksnys, V. in *Methods in Enzymology* 219–240 (Academic Press Inc., Jan. 2019). ISBN: 9780128167601. https://pubmed.ncbi.nlm.nih.gov/30691644/.

47. Gasiunas, G. *et al.* A catalogue of biochemically diverse CRISPR-Cas9 orthologs. *Nature Communications* **11,** 1–10. ISSN: 20411723. https://doi.org/10.1038/s41467-020-19344-1 (Dec. 2020).

48. Najm, F. J. *et al.* Orthologous CRISPR-Cas9 enzymes for combinatorial genetic screens. *Nature Biotechnology* **36,** 179–189. ISSN: 15461696. https://www.nature.com/articles/nbt.4048 (Feb. 2018).

49. Mali, P. *et al.* RNA-guided human genome engineering via Cas9. *Science* **339,** 823–826. ISSN: 10959203. www.sciencemag.org/cgi/content/full/science.1231143/DC1 (Feb. 2013).

50. Cong, L. *et al.* Multiplex genome engineering using CRISPR/Cas systems. *Science* **339,** 819–823. ISSN: 10959203. www.sciencemag.org/cgi/content/full/339/6121/816/DC1 (Feb. 2013).

51. Jinek, M. *et al.* RNA-programmed genome editing in human cells. *eLife* **2013.** ISSN: 2050084X (Jan. 2013).

52. Nielsen, A. A. & Voigt, C. A. Multi-input CRISPR / C as genetic circuits that interface host regulatory networks. *Molecular Systems Biology* **10,** 763. ISSN: 1744-4292. https://pubmed.ncbi.nlm.nih.gov/25422271/ (Nov. 2014).

53. Zalatan, J. G. *et al.* Engineering complex synthetic transcriptional programs with CRISPR RNA scaffolds. *Cell* **160,** 339–350. ISSN: 10974172. http://dx.doi.org/10.1016/j.cell.2014.11.052 (Jan. 2015).

54. Gilbert, L. A. *et al.* CRISPR-mediated modular RNA-guided regulation of transcription in eukaryotes. *Cell* **154,** 442. ISSN: 10974172. https://pubmed.ncbi.nlm.nih.gov/23849981/ (July 2013).

55. Smith, J. D. *et al.* Quantitative CRISPR interference screens in yeast identify chemical-genetic interactions and new rules for guide RNA design. *Genome Biology* **17,** 45. ISSN:

1474760X. https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0900-9 (Mar. 2016).

56. Doench, J. G. *et al.* Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature Biotechnology* **34,** 184–191. ISSN: 15461696. https://www.nature.com/articles/nbt.3437 (Feb. 2016).

57. Qing, A. in *Differential Evolution* 41–60 (John Wiley & Sons, Ltd, Chichester, UK, Sept. 2009). https://onlinelibrary.wiley.com/doi/10.1002/9780470823941.ch2.

58. Hobert, O. Common logic of transcription factor and microRNA action. *Trends in Biochemical Sciences* **29,** 462–468. ISSN: 09680004. https://pubmed.ncbi.nlm.nih.gov/15337119/ (2004).

59. *Logical Modeling of Biological Systems — Wiley* https://www.wiley.com/en-gb/Logical+Modeling+of+Biological+Systems-p-9781848216808.

60. Gibson, D. G. *et al.* Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Methods* **6,** 343–345. ISSN: 15487091. http://npg.nature.com/reprintsandpermissions/ (Apr. 2009).

61. Gietz, R. D. & Schiestl, R. H. Applications of high efficiency lithium acetate transformation of intact yeast cells using single-stranded nucleic acids as carrier. *Yeast* **7,** 253–263. ISSN: 10970061. https://pubmed.ncbi.nlm.nih.gov/1882550/ (1991).

62. Qian, Y., Huang, H. H., Jiménez, J. I. & Del Vecchio, D. Resource Competition Shapes the Response of Genetic Circuits. *ACS Synthetic Biology* **6,** 1263–1272. ISSN: 21615063. https://pubs.acs.org/doi/abs/10.1021/acssynbio.6b00361 (July 2017).

63. Daher, M., Mustoe, A. M., Morriss-Andrews, A., Brooks, C. L. & Walter, N. G. Tuning RNA folding and function through rational design of junction topology. *Nucleic Acids Research* **45,** 9706–9715. ISSN: 13624962. http://medicine.yale.edu/keck/ (Sept. 2017).

64. Marino, S., Hogue, I. B., Ray, C. J. & Kirschner, D. E. *A methodology for performing global uncertainty and sensitivity analysis in systems biology* Sept. 2008. /pmc/articles/PMC2570191/%20/pmc/articles/PMC2570191/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2570191/.

65. Jorgensen, P. *et al.* The size of the nucleus increases as yeast cells grow. *Molecular Biology of the Cell* **18,** 3523–3532. ISSN: 10591524. https://pubmed.ncbi.nlm.nih.gov/17596521/ (Sept. 2007).

66. Richardson, C. D., Ray, G. J., DeWitt, M. A., Curie, G. L. & Corn, J. E. Enhancing homology-directed genome editing by catalytically active and inactive CRISPR-Cas9 using asymmetric donor DNA. *Nature Biotechnology* **34,** 339–344. ISSN: 15461696. https://www.nature.com/articles/nbt.3481 (Mar. 2016).

67. Pelechano, V., Chávez, S. & Pérez-Ortín, J. E. A Complete Set of Nascent Transcription Rates for Yeast Genes. *PLoS ONE* **5** (ed Santos, J.) e15442. ISSN: 1932-6203. https://dx.plos.org/10.1371/journal.pone.0015442 (Nov. 2010).

68. Christiano, R., Nagaraj, N., Fröhlich, F. & Walther, T. C. Global Proteome Turnover Analyses of the Yeasts S.cerevisiae and S.pombe. *Cell Reports* **9,** 1959–1965. ISSN: 22111247. https://pubmed.ncbi.nlm.nih.gov/25466257/ (Dec. 2014).

69. Hochschild, A. & Ptashne, M. Cooperative binding of λ repressors to sites separated by integral turns of the DNA helix. *Cell* **44,** 681–687. ISSN: 00928674. http://www.cell.com/article/0092867486908330/fulltext%20http://www.cell.com/article/0092867486908330/abstract%20https://www.cell.com/cell/abstract/0092-8674(86)90833-0 (Mar. 1986).

70. Ramos, J. L. *et al.* The TetR Family of Transcriptional Repressors. *Microbiology and Molecular Biology Reviews* **69,** 326–356. ISSN: 1092-2172. https://pubmed.ncbi.nlm.nih.gov/15944459/ (June 2005).

71. Brewster, R. C. *et al.* The transcription factor titration effect dictates level of gene expression. *Cell* **156,** 1312–1323. ISSN: 10974172. http://dx.doi.org/10.1016/j.cell.2014.02.022 (Mar. 2014).

72. Hirose, T. *et al.* NEAT1 long noncoding RNA regulates transcription via protein sequestration within subnuclear bodies. *Molecular Biology of the Cell* **25,** 169–183. ISSN: 10591524. https://pubmed.ncbi.nlm.nih.gov/24173718/ (Jan. 2014).

73. Liu, X., Wu, B., Szary, J., Kofoed, E. M. & Schaufele, F. Functional sequestration of transcription factor activity by repetitive DNA. *Journal of Biological Chemistry* **282,** 20868–20876. ISSN: 00219258. https://pubmed.ncbi.nlm.nih.gov/17526489/ (July 2007).

74. Yamanaka, T. & Nukina, N. in *Methods in Molecular Biology* 215–229 (Humana Press Inc., 2010). https://pubmed.ncbi.nlm.nih.gov/20700715/.

75. Ricci, F., Vallée-Bélisle, A. & Plaxco, K. W. High-Precision, In Vitro Validation of the Sequestration Mechanism for Generating Ultrasensitive Dose-Response Curves in Regulatory Networks. *PLoS Computational Biology* **7** (ed Fan, C.) e1002171. ISSN: 1553-7358. https://dx.plos.org/10.1371/journal.pcbi.1002171 (Oct. 2011).

76. Rydenfelt, M., Cox, R. S., Garcia, H. & Phillips, R. Statistical mechanical model of coupled transcription from multiple promoters due to transcription factor titration. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* **89.** ISSN: 15393755. https://pubmed.ncbi.nlm.nih.gov/24580252/ (Jan. 2014).

77. Ryder, S. P., Recht, M. I. & Williamson, J. R. Quantitative analysis of protein-RNA interactions by gel mobility shift. *Methods in Molecular Biology* **488,** 99–115. ISSN: 10643745. https://pubmed.ncbi.nlm.nih.gov/18982286/ (2008).

78. Clarke, L. & Carbon, J. Isolation of a yeast centromere and construction of functional small circular chromosomes. *Nature* **287,** 504–509. ISSN: 00280836. https://www.nature.com/articles/287504a0 (1980).

79. Huong Le, T. T. *et al.* Efficient and precise CRISPR/Cas9-mediated MECP2 modifications in human-induced pluripotent stem cells. *Frontiers in Genetics* **10,** 625. ISSN: 16648021. /pmc/articles/PMC6614930/%20/pmc/articles/PMC6614930/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6614930/ (2019).

80. Yeo, N. C. *et al.* An enhanced CRISPR repressor for targeted mammalian gene regulation. *Nature Methods* **15,** 611–616. ISSN: 15487105. https://doi.org/10.1038/s41592-018-0048-5 (Aug. 2018).

81. Qi, L. S. *et al.* Repurposing CRISPR as an RNA-γuided platform for sequence-specific control of gene expression. *Cell* **152,** 1173–1183. ISSN: 10974172. /pmc/articles/PMC3664290/%20/pmc/articles/PMC3664290/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3664290/ (Feb. 2013).

82. Radzisheuskaya, A., Shlyueva, D., Müller, I. & Helin, K. Optimizing sgRNA position markedly improves the efficiency of CRISPR/dCas9-mediated transcriptional repression. *Nucleic Acids Research* **44,** e141. ISSN: 13624962. /pmc/articles/PMC5062975/%20/pmc/articles/PMC5062975/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5062975/ (Oct. 2016).

83. Counsell, J. R. *et al.* Lentiviral vectors can be used for full-length dystrophin gene therapy. *Scientific Reports* **7,** 1–11. ISSN: 20452322. www.nature.com/scientificreports/ (Mar. 2017).

84. Sweeney, N. P. & Vink, C. A. The impact of lentiviral vector genome size and producer cell genomic to gag-pol mRNA ratios on packaging efficiency and titre. *Molecular Therapy - Methods and Clinical Development* **21,** 574–584. ISSN: 23290501. https://doi.org/10.1016/j.omtm.2021.04.007. (June 2021).

85. Zhou, H., Huang, C. & Xia, X. G. A tightly regulated Pol III promoter for synthesis of miRNA genes in tandem. *Biochimica et Biophysica Acta - Gene Regulatory Mechanisms* **1779,** 773–779. ISSN: 18749399. /pmc/articles/PMC2607239/%20/pmc/

articles/PMC2607239/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2607239/ (Nov. 2008).

86. Meissner, W., Rothfels, H., Schäfer, B. & Seifart, K. Development of an inducible pol III transcription system essentially requiring a mutated form of the TATA-binding protein. *Nucleic Acids Research* **29,** 1672–1682. ISSN: 03051048. /pmc/articles/PMC31323/%20/pmc/articles/PMC31323/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC31323/ (Apr. 2001).

87. Haurwitz, R. E., Jinek, M., Wiedenheft, B., Zhou, K. & Doudna, J. A. Sequence- and structure-specific RNA processing by a CRISPR endonuclease. *Science* **329,** 1355–1358. ISSN: 00368075. https://pubmed-ncbi-nlm-nih-gov.offcampus.lib.washington.edu/20829488/ (Sept. 2010).

88. Haurwitz, R. E., Sternberg, S. H. & Doudna, J. A. Csy4 relies on an unusual catalytic dyad to position and cleave CRISPR RNA. *EMBO Journal* **31,** 2824–2832. ISSN: 02614189. /pmc/articles/PMC3380207/%20/pmc/articles/PMC3380207/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3380207/ (June 2012).

89. Tsai, S. Q. *et al.* Dimeric CRISPR RNA-guided FokI nucleases for highly specific genome editing. *Nature Biotechnology* **32,** 569–576. ISSN: 15461696. https://pubmed.ncbi.nlm.nih.gov/24770325/ (2014).

90. Sternberg, S. H., Haurwitz, R. E. & Doudna, J. A. Mechanism of substrate selection by a highly specific CRISPR endoribonuclease. *RNA* **18,** 661–672. ISSN: 13558382. /pmc/articles/PMC3312554/%20/pmc/articles/PMC3312554/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3312554/ (Apr. 2012).

91. Ferreira, R., Skrekas, C., Nielsen, J. & David, F. Multiplexed CRISPR/Cas9 Genome Editing and Gene Regulation Using Csy4 in Saccharomyces cerevisiae. *ACS Synthetic Biology* **7,** 10–15. ISSN: 21615063. https://pubs.acs.org/doi/abs/10.1021/acssynbio.7b00259 (Jan. 2018).

92. Chen, B. *et al.* Dynamic imaging of genomic loci in living human cells by an optimized CRISPR/Cas system. *Cell* **155,** 1479–1491. ISSN: 10974172 (Dec. 2013).

93. Orioli, A. *et al.* Widespread occurrence of non-canonical transcription termination by human RNA polymerase III. *Nucleic Acids Research* **39,** 5499–5512. ISSN: 03051048 (July 2011).

94. Gilbert, L. A. *et al.* Genome-Scale CRISPR-Mediated Control of Gene Repression and Activation. *Cell* **159,** 647–661. ISSN: 10974172. http://dx.doi.org/10.1016/j.cell.2014.09.029 (Oct. 2014).

95. Kearns, N. A. *et al.* Functional annotation of native enhancers with a Cas9-histone demethylase fusion. *Nature Methods* **12,** 401–403. ISSN: 15487105. /pmc/articles/PMC4414811/%20/pmc/articles/PMC4414811/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4414811/ (Apr. 2015).

96. Gao, X. *et al.* Comparison of TALE designer transcription factors and the CRISPR/d-Cas9 in regulation of gene expression by targeting enhancers. *Nucleic Acids Research* **42,** e155. ISSN: 13624962. /pmc/articles/PMC4227760/%20/pmc/articles/PMC4227760/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4227760/ (Nov. 2014).

97. Thakore, P. I. *et al.* Highly specific epigenome editing by CRISPR-Cas9 repressors for silencing of distal regulatory elements. *Nature Methods* **12,** 1143–1149. ISSN: 15487105. /pmc/articles/PMC4666778/%20/pmc/articles/PMC4666778/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4666778/ (Dec. 2015).

98. Donahue, P. S. *et al.* The COMET toolkit for composing customizable genetic programs in mammalian cells. *Nature Communications* **11,** 1–19. ISSN: 20411723. https://doi.org/10.1038/s41467-019-14147-5 (Dec. 2020).

99. Der, B. S. *et al.* DNAplotlib: Programmable Visualization of Genetic Designs and Associated Data. *ACS Synthetic Biology* **6,** 1115–1119. ISSN: 21615063. https://pubs.acs.org/doi/abs/10.1021/acssynbio.6b00252 (July 2017).

100. Keller, B., Vrana, J., Miller, A., Newman, G. & Klavins, E. Aquarium: The Laboratory Operating System (Version v2.5.0. en. *Zenodo.* http://doiorg/105281/zenodo2535715..

101. Fang, F. C. & Casadevall, A. *Reforming science: Structural reforms* Mar. 2012. /pmc/articles/PMC3294664/%20/pmc/articles/PMC3294664/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3294664/.

102. Begley, C. G. & Ellis, L. M. Drug development: Raise standards for preclinical cancer research. *Nature* **483,** 531–533. ISSN: 00280836. https://pubmed.ncbi.nlm.nih.gov/22460880/ (Mar. 2012).

103. Prinz, F., Schlange, T. & Asadullah, K. *Believe it or not: How much can we rely on published data on potential drug targets?* Sept. 2011. http://www.newyorker.com/.

104. National Academies of Sciences Engineering & Medicine. *Reproducibility and Replicability in Science* ISBN: 978-0-309-48616-3. https://www.nap.edu/catalog/25303/reproducibility-and-replicability-in-science (The National Academies Press, Washington, DC, 2019).

105. Ioannidis, J. P. A. Why Most Published Research Findings Are False. *PLoS Medicine* **2,** e124. ISSN: 1549-1676. https://dx.plos.org/10.1371/journal.pmed.0020124 (Aug. 2005).

106. Goodman, S. N., Fanelli, D. & Ioannidis, J. P. in *Getting to Good: Research Integrity in the Biomedical Sciences* 341, 96–102 (Springer International Publishing, July 2018). ISBN: 9783319513584. www.ScienceTranslationalMedicine.org.

107. Baker, M. & Penny, D. *Is there a reproducibility crisis?* May 2016.

108. Fanelli, D. *Is science really facing a reproducibility crisis, and do we need it to?* Mar. 2018. www.pnas.org/lookup/suppl/doi:10.1073/pnas.1708272114/-/DCSupplemental..

109. Fonio, E., Golani, I. & Benjamini, Y. *Measuring behavior of animal models: Faults and remedies* Dec. 2012. https://pubmed.ncbi.nlm.nih.gov/23223171/.

110. Arroyo-Araujo, M. *et al.* Reproducibility via coordinated standardization: a multi-center study in a Shank2 genetic rat model for Autism Spectrum Disorders. *Scientific Reports* **9,** 1–10. ISSN: 20452322. https://doi.org/10.1038/s41598-019-47981-0 (Dec. 2019).

111. Nussbeck, S. Y. *et al.* The laboratory notebook in the 21 st century. *EMBO reports* **15,** 631–634. ISSN: 1469-221X. https://pubmed.ncbi.nlm.nih.gov/24833749/ (June 2014).

112. Miles, B. & Lee, P. L. Achieving Reproducibility and Closed-Loop Automation in Biological Experimentation with an IoT-Enabled Lab of the Future. *SLAS Technology* **23,** 432–439. ISSN: 24726311. http://journals.sagepub.com/doi/10.1177/2472630318784506 (Oct. 2018).

113. Dennis, J. B. *First version of a data flow procedure language* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **19 LNCS** (Springer Verlag, 1974), 362–376. ISBN: 9783540068594. https://link.springer.com/chapter/10.1007/3-540-06859-7_145.

114. *Ruby Programming Language* https://www.ruby-lang.org/en/.

115. Younger, D., Berger, S., Baker, D. & Klavins, E. High-throughput characterization of protein–protein interactions by reprogramming yeast mating. *Proceedings of the National Academy of Sciences of the United States of America* **114,** 12166–12171. ISSN: 10916490. www.pnas.org/cgi/doi/10.1073/pnas.1705867114 (Nov. 2017).

116. Khakhar, A., Bolten, N. J., Nemhauser, J. & Klavins, E. Cell-Cell Communication in Yeast Using Auxin Biosynthesis and Auxin Responsive CRISPR Transcription Factors. *ACS Synthetic Biology* **5,** 279–286. ISSN: 21615063. https://pubs.acs.org/doi/abs/10.1021/acssynbio.5b00064 (Apr. 2016).

117. Khakhar, A., Leydon, A. R., Lemmex, A. C., Klavins, E. & Nemhauser, J. L. Synthetic hormone-responsive transcription factors can monitor and reprogram plant development. *eLife* **7.** ISSN: 2050084X (May 2018).

118. Groves, B., Khakhar, A., Nadel, C. M., Gardner, R. G. & Seelig, G. Rewiring MAP kinases in Saccharomyces cerevisiae to regulate novel targets through ubiquitination. *eLife* **5.** ISSN: 2050084X (Aug. 2016).

119. Panpradist, N. *et al.* OLA-Simple: A software-guided HIV-1 drug resistance test for low-resource laboratories. *EBioMedicine* **50.** ISSN: 23523964 (2019).

120. Vrana, J. *OLA Simple HIV Drug Resistance Test: An Aquarium Workflow Github.com2020 Available from: klavinslab.github.io/ola-simple/* en.

121. Marrs, K. A. & Novak, G. *Just-in-Time Teaching in biology: Creating an active learner classroom using the internet* Mar. 2004.

122. Rohl, C. A., Strauss, C. E., Misura, K. M. & Baker, D. Protein Structure Prediction Using Rosetta. *Methods in Enzymology* **383,** 66–93. ISSN: 00766879 (Jan. 2004).

123. Hillson, N. J., Rosengarten, R. D. & Keasling, J. D. J5 DNA assembly design automation software. *ACS Synthetic Biology* **1,** 14–21. ISSN: 21615063. https://pubs.acs.org/doi/abs/10.1021/sb2000116 (Jan. 2012).

124. Untergasser, A. *et al.* Primer3-new capabilities and interfaces. *Nucleic Acids Research* **40,** e115–e115. ISSN: 03051048. http://www.ncbi.nlm.nih.gov/tools/primer-blast/ (Aug. 2012).

125. Koressaar, T. & Remm, M. Enhancements and modifications of primer design program Primer3. *Bioinformatics* **23,** 1289–1291. ISSN: 13674803. https://pubmed.ncbi.nlm.nih.gov/17379693/ (May 2007).

126. Kõressaar, T. *et al.* Primer3-masker: Integrating masking of template sequence with primer design software. *Bioinformatics* **34,** 1937–1938. ISSN: 14602059. https://pubmed.ncbi.nlm.nih.gov/29360956/ (June 2018).

127. Appleton, E., Tao, J., Haddock, T. & Densmore, D. Interactive assembly algorithms for molecular cloning. *Nature Methods* **11,** 657–662. ISSN: 15487105. http://www.sbolstandard.org/ (Apr. 2014).

128. Vrana, J. *et al.* Aquarium: open-source laboratory software for design, execution and data management. *Synthetic Biology.* ISSN: 2397-7000. https://mc.manuscriptcentral.com/synbiohttps://mc.manuscriptcentral.com/synbio (Jan. 2021).

129. Blount, B. A., Weenink, T. & Ellis, T. *Construction of synthetic regulatory networks in yeast* July 2012. https://pubmed.ncbi.nlm.nih.gov/22309848/%20https://pubmed.ncbi.nlm.nih.gov/22309848/?dopt=Abstract.

130. Regot, S. *et al.* Distributed biological computation with multicellular engineered networks. *Nature* **469,** 207–211. ISSN: 00280836. https://pubmed.ncbi.nlm.nih.gov/21150900/%20https://pubmed.ncbi.nlm.nih.gov/21150900/?dopt=Abstract (Jan. 2011).

131. Buchler, N. E. & Cross, F. R. Protein sequestration generates a flexible ultrasensitive response in a genetic network. *Molecular Systems Biology* **5.** ISSN: 17444292. https://pubmed.ncbi.nlm.nih.gov/19455136/%20https://pubmed.ncbi.nlm.nih.gov/19455136/?dopt=Abstract (Jan. 2009).

132. Marucci, L. *et al.* How to turn a genetic circuit into a synthetic tunable oscillator, or a bistable switch. *PLoS ONE* **4.** ISSN: 19326203. https://pubmed.ncbi.nlm.nih.gov/19997611/%20https://pubmed.ncbi.nlm.nih.gov/19997611/?dopt=Abstract (2009).

133. Ajo-Franklin, C. M. *et al.* Rational design of memory in eukaryotic cells. *Genes and Development* **21,** 2271–2276. ISSN: 08909369. https://pubmed.ncbi.nlm.nih.gov/17875664/%20https://pubmed.ncbi.nlm.nih.gov/17875664/?dopt=Abstract (Sept. 2007).

134. Yamanishi, M. & Matsuyama, T. A modified cre-lox genetic switch to dynamically control metabolic flow in saccharomyces cerevisiae. *ACS Synthetic Biology* **1,** 172–

180. ISSN: 21615063. https://pubmed.ncbi.nlm.nih.gov/23651155/%20https://pubmed.ncbi.nlm.nih.gov/23651155/?dopt=Abstract (May 2012).

135. Ryu, J. & Park, S. H. Simple synthetic protein scaffolds can create adjustable artificial MAPK circuits in yeast and mammalian cells. *Science Signaling* **8.** ISSN: 19379145. https://pubmed.ncbi.nlm.nih.gov/26126717/%20https://pubmed.ncbi.nlm.nih.gov/26126717/?dopt=Abstract (June 2015).

136. Youk, H. & Lim, W. A. Secreting and sensing the same molecule allows cells to achieve versatile social behaviors. *Science* **343.** ISSN: 10959203. https://pubmed.ncbi.nlm.nih.gov/24503857/%20https://pubmed.ncbi.nlm.nih.gov/24503857/?dopt=Abstract (2014).

137. Chen, Y. *et al.* Genetic circuit design automation for yeast. *Nature Microbiology* **5,** 1349–1360. ISSN: 20585276. https://doi.org/10.1038/s41564-020-0757-2 (Nov. 2020).

138. Ottoz, D. S., Rudolf, F. & Stelling, J. Inducible, tightly regulated and growth condition-independent transcription factor in Saccharomyces cerevisiae. *Nucleic Acids Research* **42.** ISSN: 13624962. https://pubmed.ncbi.nlm.nih.gov/25034689/%20https://pubmed.ncbi.nlm.nih.gov/25034689/?dopt=Abstract (June 2014).

139. Langan, R. A. *et al.* De novo design of bioactive protein switches. *Nature* **572,** 205–210. ISSN: 14764687. https://pubmed.ncbi.nlm.nih.gov/31341284/%20https://pubmed.ncbi.nlm.nih.gov/31341284/?dopt=Abstract (Aug. 2019).

140. Ng, A. H. *et al.* Modular and tunable biological feedback control using a de novo protein switch. *Nature* **572,** 265–269. ISSN: 14764687. https://pubmed.ncbi.nlm.nih.gov/31341280/%20https://pubmed.ncbi.nlm.nih.gov/31341280/?dopt=Abstract (Aug. 2019).

141. Ryo, S. *et al.* Positive Feedback Genetic Circuit Incorporating a Constitutively Active Mutant Gal3 into Yeast GAL Induction System. *ACS Synthetic Biology* **6,** 928–935.

ISSN: 21615063. https://pubmed.ncbi.nlm.nih.gov/28324652/%20https://pubmed.ncbi.nlm.nih.gov/28324652/?dopt=Abstract (June 2017).

142. Yang, Y., Nemhauser, J. L. & Klavins, E. Synthetic Bistability and Differentiation in Yeast. *ACS Synthetic Biology* **8,** 929–936. ISSN: 21615063. https://pubmed.ncbi.nlm.nih.gov/31021593/%20https://pubmed.ncbi.nlm.nih.gov/31021593/?dopt=Abstract (May 2019).

143. Krivoruchko, A., Siewers, V. & Nielsen, J. Opportunities for yeast metabolic engineering: Lessons from synthetic biology. *Biotechnology Journal* **6,** 262–276. ISSN: 18606768. https://pubmed.ncbi.nlm.nih.gov/21328545/%20https://pubmed.ncbi.nlm.nih.gov/21328545/?dopt=Abstract (Mar. 2011).

144. Ferreira, R. *et al.* Model-Assisted Fine-Tuning of Central Carbon Metabolism in Yeast through dCas9-Based Regulation. *ACS Synthetic Biology* **8,** 2457–2463. ISSN: 21615063. https://pubmed.ncbi.nlm.nih.gov/31577419/%20https://pubmed.ncbi.nlm.nih.gov/31577419/?dopt=Abstract (Nov. 2019).

145. Ellis, T., Wang, X. & Collins, J. J. Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nature Biotechnology* **27,** 465–471. ISSN: 10870156. https://pubmed.ncbi.nlm.nih.gov/19377462/%20https://pubmed.ncbi.nlm.nih.gov/19377462/?dopt=Abstract (May 2009).

146. Casini, A. *et al.* A Pressure Test to Make 10 Molecules in 90 Days: External Evaluation of Methods to Engineer Biology. *Journal of the American Chemical Society* **140,** 4302–4316. ISSN: 15205126. https://pubs.acs.org/sharingguidelines (Mar. 2018).

147. Trosset, J. Y. & Carbonell, P. *Synthetic biology for pharmaceutical drug discovery* Dec. 2015. /pmc/articles/PMC4675648/%20/pmc/articles/PMC4675648/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4675648/.

148. Tang, T. C. *et al. Materials design by synthetic biology* Apr. 2021. www.nature.com/natrevmats.

149. Nguyen, P. Q., Courchesne, N.-M. D., Duraj-Thatte, A., Praveschotinunt, P. & Joshi, N. S. Engineered Living Materials: Prospects and Challenges for Using Biological Systems to Direct the Assembly of Smart Materials. *Advanced Materials* **30,** 1704847. ISSN: 09359648. https://onlinelibrary.wiley.com/doi/10.1002/adma.201704847 (May 2018).

150. Moradali, M. F. & Rehm, B. H. *Bacterial biopolymers: from pathogenesis to advanced materials* Apr. 2020. www.nature.com/nrmicro.

151. Oberortner, E., Cheng, J. F., Hillson, N. J. & Deutsch, S. Streamlining the Design-to-Build Transition with Build-Optimization Software Tools. *ACS Synthetic Biology* **6,** 485–496. ISSN: 21615063. https://boost.jgi.doe.gov (Mar. 2017).

152. Madsen, C. *et al.* The SBOL stack: A platform for storing, publishing, and sharing synthetic biology designs. *ACS Synthetic Biology* **5,** 487–497. ISSN: 21615063. http://parts.igem.org/ (June 2016).

153. Ham, T. S. *et al.* Design, implementation and practice of JBEI-ICE: An open source biological part registry platform and tools. *Nucleic Acids Research* **40.** ISSN: 03051048. https://pubmed.ncbi.nlm.nih.gov/22718978/ (Oct. 2012).

154. Geer, L. Y. *et al.* The NCBI BioSystems database. *Nucleic Acids Research* **38.** ISSN: 03051048. https://pubmed.ncbi.nlm.nih.gov/19854944/ (Oct. 2009).

155. Wilson, M. L., Hertzberg, R., Adam, L. & Peccoud, J. A step-by-step introduction to rule-based design of synthetic genetic constructs using GenoCAD. *Methods in enzymology* **498,** 173–88. ISSN: 1557-7988. http://www.ncbi.nlm.nih.gov/pubmed/21601678 (2011).

156. Czar, M. J., Cai, Y. & Peccoud, J. Writing DNA with genoCAD™. *Nucleic Acids Research* **37.** ISSN: 03051048. https://pubmed.ncbi.nlm.nih.gov/19429897/ (2009).

157. Beal, J. & Rogers, M. Levels of autonomy in synthetic biology engineering. *Molecular Systems Biology* **16,** e10019. ISSN: 1744-4292. https://www.embopress.org/doi/full/10.15252/msb.202010019%20https://www.embopress.org/doi/abs/10.15252/msb.202010019 (Dec. 2020).

158. Chapman, T. Lab automation and robotics: Automation on the move. *Nature* **421,** 661–666. ISSN: 00280836. www.nature.com/nature (Feb. 2003).

159. Blow, N. Lab automation: Tales along the road to automation. *Nature Methods* **5,** 109–112. ISSN: 15487091. http://www.nature.com/naturemethods (Jan. 2008).

160. Linshiz, G. *et al.* PaR-PaR laboratory automation platform. *ACS Synthetic Biology* **2,** 216–222. ISSN: 21615063. http://sbolstandard.org (May 2013).

161. Tabor, D. P. *et al. Accelerating the discovery of materials for clean energy in the era of smart automation* May 2018. www.nature.com/natrevmats.

162. Burger, B. *et al.* A mobile robotic chemist. *Nature* **583,** 237–241. ISSN: 14764687. https://doi.org/10.1038/s41586-020-2442-2 (July 2020).

163. Granda, J. M., Donina, L., Dragone, V., Long, D. L. & Cronin, L. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature* **559,** 377–381. ISSN: 14764687. https://doi.org/10.1038/s41586-018-0307-8 (July 2018).

164. Daniszewski, M. *et al. Automated Cell Culture Systems and Their Applications to Human Pluripotent Stem Cell Studies* Aug. 2018. http://journals.sagepub.com/doi/10.1177/2472630317712220.

165. Henson, A. B., Gromski, P. S. & Cronin, L. Designing Algorithms to Aid Discovery by Chemical Robots. *ACS Central Science* **4,** 793–804. ISSN: 23747951. http://pubchem.ncbi.nlm.nih.gov/ (July 2018).

166. Correa-Baena, J. P. *et al. Accelerating Materials Development via Automation, Machine Learning, and High-Performance Computing* Aug. 2018.

167. Baker, M. *Biotech giant publishes failures to confirm high-profile science* Feb. 2016.

168. Freedman, L. P., Cockburn, I. M. & Simcoe, T. S. The Economics of Reproducibility in Preclinical Research. *PLOS Biology* **13,** e1002165. ISSN: 1545-7885. https://dx.plos.org/10.1371/journal.pbio.1002165 (June 2015).

169. Al-wswasi, M., Ivanov, A. & Makatsoris, H. A survey on smart automated computer-aided process planning (ACAPP) techniques. *International Journal of Advanced Manufacturing Technology* **97,** 809–832. ISSN: 14333015. https://doi.org/10.1007/s00170-018-1966-1 (July 2018).

170. Xu, X., Wang, L. & Newman, S. T. Computer-aided process planning - A critical review of recent developments and future trends. *International Journal of Computer Integrated Manufacturing* **24,** 1–31. ISSN: 13623052. https://www.tandfonline.com/doi/abs/10.1080/0951192X.2010.518632 (2011).

171. Kumar, M. & Rajotia, S. *Integration of scheduling with computer aided process planning* in *Journal of Materials Processing Technology* **138** (Elsevier, July 2003), 297–300.

172. Ham, I. & Lu, S. C. Computer-Aided Process Planning: The Present and the Future. *CIRP Annals - Manufacturing Technology* **37,** 591–601. ISSN: 17260604 (1988).

173. Halevi, G. & Weill, R. D. in *Principles of Process Planning* 317–332 (Springer Netherlands, 1995).

174. Behandish, M., Nelaturi, S. & de Kleer, J. Automated process planning for hybrid manufacturing. *CAD Computer Aided Design* **102,** 115–127. ISSN: 00104485 (Sept. 2018).

175. Park, J. Y. & Khoshnevis, B. A real-time computer-aided process planning system as a support tool for economic product design. *Journal of Manufacturing Systems* **12,** 181–193. ISSN: 02786125 (1993).

176. Culler, D. E. & Burd, W. A framework for extending computer aided process planning to include business activities and computer aided design and manufacturing (CAD/-CAM) data retrieval. *Robotics and Computer-Integrated Manufacturing* **23,** 339–350. ISSN: 07365845 (June 2007).

177. Pendleton, I. M. *et al.* Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): A software pipeline for automated chemical experimentation and data management. *MRS Communications* **9,** 846–859. ISSN: 21596867. https://www.cambridge.org/core/journals/mrs-communications/article/abs/experiment-specification-capture-and-laboratory-automation-technology-escalate-a-software-pipeline-for-automated-chemical-experimentation-and-data-management/7C6C5B29BCBD1B48A170696AF187E023 (Sept. 2019).

178. Schreck, J. S., Coley, C. W. & Bishop, K. J. Learning Retrosynthetic Planning through Simulated Experience. *ACS Central Science* **5,** 970–981. ISSN: 23747951. http://pubs.acs.org/journal/acscii (June 2019).

179. Sarkozi, L., Simson, E. & Ramanathan, L. The effects of total laboratory automation on the management of a clinical chemistry laboratory. Retrospective analysis of 36 years. en. *Clin Chim Acta* **329,** 89–94.

180. Finnigan, W., Hepworth, L. J., Flitsch, S. L. & Turner, N. J. RetroBioCat as a computer-aided synthesis planning tool for biocatalytic reactions and cascades. *Nature Catalysis* **4,** 98–104. ISSN: 25201158. https://doi.org/10.1038/s41929-020-00556-z (Feb. 2021).

181. Lin, K., Xu, Y., Pei, J. & Lai, L. Automatic retrosynthetic route planning using template-free models. *Chemical Science* **11,** 3355–3364. ISSN: 20416539. https://pubs.rsc.org/en/content/articlehtml/2020/sc/c9sc03666k%20https://pubs.rsc.org/en/content/articlelanding/2020/sc/c9sc03666k (Mar. 2020).

182. Segler, M. H., Preuss, M. & Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **555,** 604–610. ISSN: 14764687. https://www.nature.com/articles/nature25978 (Mar. 2018).

183. Opgenorth, P. *et al.* Lessons from Two Design-Build-Test-Learn Cycles of Dodecanol Production in Escherichia coli Aided by Machine Learning. *ACS Synthetic Biology* **8,** 1337–1351. ISSN: 21615063. https://pubs.acs.org/sharingguidelines (June 2019).

184. Carbonell, P., Radivojevic, T. & García Martín, H. *Opportunities at the Intersection of Synthetic Biology, Machine Learning, and Automation* July 2019. http://www.jbei.org.

185. Dartigues, C., Ghodous, P., Gruninger, M., Pallez, D. & Sriram, R. *CAD/CAPP Integration using Feature Ontology* tech. rep. ().

186. Hou, M. & Faddis, T. N. Automatic tool path generation of a feature-basedCAD/CAPP/CAM integrated system. *International Journal of Computer Integrated Manufacturing* **19,** 350–358. ISSN: 0951192X (June 2006).

187. Crockford, D. C. *The application/json Media Type for JavaScript Object Notation (JSON)* https://datatracker.ietf.org/doc/html/rfc4627.

188. Chlebík, M. & Chlebíková, J. The Steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science* **406,** 207–214. ISSN: 03043975 (Oct. 2008).

189. Biniaz, A., Maheshwari, A. & Smid, M. On the hardness of full Steiner tree problems. *Journal of Discrete Algorithms* **34,** 118–127. ISSN: 15708667 (Sept. 2015).

190. Gietz, R. D. & Woods, R. A. Transformation of yeast by lithium acetate/single-stranded carrier DNA/polyethylene glycol method. *Methods in Enzymology* **350,** 87–96. ISSN: 00766879 (Jan. 2002).

191. Cummins, B., Gedeon, T., Harker, S. & Mischaikow, K. DSGRN: Examining the Dynamics of Families of Logical Models. *Frontiers in Physiology* **9,** 549. ISSN: 1664-042X. https://www.frontiersin.org/article/10.3389/fphys.2018.00549/full (May 2018).

192. Tong, Y., Zhou, J., Zhang, L. & Xu, P. A Golden-Gate Based Cloning Toolkit to Build Violacein Pathway Libraries in Yarrowia lipolytica. *ACS Synthetic Biology* **10,** 115–124. ISSN: 21615063. https://dx.doi.org/10.1021/acssynbio.0c00469 (Jan. 2021).

193. Lee, M. E., DeLoache, W. C., Cervantes, B. & Dueber, J. E. A Highly Characterized Yeast Toolkit for Modular, Multipart Assembly. *ACS Synthetic Biology* **4,** 975–986. ISSN: 21615063. https://pubs.acs.org/sharingguidelines (Sept. 2015).

194. Patrascu, M. B. *et al.* From desktop to benchtop with automated computational workflows for computer-aided design in asymmetric catalysis. *Nature Catalysis.* https://doi.org/10.1038/s41929-020-0468-3 (1929).

195. Vrabič, R., Erkoyuncu, J. A., Butala, P. & Roy, R. *Digital twins: Understanding the added value of integrated models for through-life engineering services* in *Procedia Manufacturing* **16** (Elsevier B.V., 2018), 139–146.

196. Fuller, A., Fan, Z., Day, C. & Barlow, C. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access* **8,** 108952–108971. ISSN: 21693536 (2020).

197. Kritzinger, W., Karner, M., Traar, G., Henjes, J. & Sihn, W. *Digital Twin in manufacturing: A categorical literature review and classification* in. **51** (Elsevier B.V., Jan. 2018), 1016–1022.

198. Bilberg, A. & Malik, A. A. Digital twin driven human–robot collaborative assembly. *CIRP Annals* **68,** 499–502. ISSN: 17260604 (Jan. 2019).

199. Madni, A., Madni, C. & Lucero, S. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems* **7,** 7. ISSN: 2079-8954 (Jan. 2019).

200. Kusiak, A. The generalized group technology concept. *International Journal of Production Research* **25,** 561–569. ISSN: 1366588X. https://www.tandfonline.com/doi/abs/10.1080/00207548708919861 (1987).

201. Ronda, C. *et al.* CrEdit: CRISPR mediated multi-loci gene integration in Saccharomyces cerevisiae. *Microbial Cell Factories* **14,** 97. ISSN: 14752859. http://microbialcellfactories.biomedcentral.com/articles/10.1186/s12934-015-0288-3 (July 2015).

202. Siddiqui, M. S., Choksi, A. & Smolke, C. D. A system for multilocus chromosomal integration and transformation-free selection marker rescue. *FEMS Yeast Research* **14,** 1171–1185. ISSN: 15671364. /pmc/articles/PMC4270834/%20/pmc/articles/PMC4270834/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4270834/ (Dec. 2014).

203. Norrander, J., Kempe, T. & Messing, J. Construction of improved M13 vectors using oligodeoxynucleotide-directed mutagenesis. *Gene* **26,** 101–106. ISSN: 03781119. https://pubmed.ncbi.nlm.nih.gov/6323249/ (1983).

204. Camacho, C. *et al.* BLAST+: Architecture and applications. *BMC Bioinformatics* **10,** 1–9. ISSN: 14712105. http://www.biomedcentral.com/1471-2105/10/421 (Dec. 2009).

205. Gibson, D. G. *et al.* Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Methods* **6,** 343–345. ISSN: 15487091. https://pubmed.ncbi.nlm.nih.gov/19363495/ (2009).

206. Engler, C. & Marillonnet, S. Golden Gate Cloning. *Methods in Molecular Biology* **1116.**

207. Panpradist, N. *et al.* Simpler and faster Covid-19 testing: Strategies to streamline SARS-CoV-2 molecular assays. *EBioMedicine* **64.** ISSN: 23523964 (2021).

208. Vrana, J. D. *et al.* Implementation of an interactive mobile application to pilot a rapid assay to detect HIV drug resistance mutations in Kenya. https://doi.org/10.1101/2021.05.06.21256654.

209. Holland, L. L., Smith, L. L. & Blick, K. E. Total laboratory automation can help eliminate the laboratory as a factor in emergency department length of stay. en. *Am J Clin Pathol* **125,** 765–770.

210. G, D. R., M, Z. & G, L. Integration of Diagnostic Microbiology in a Model of Total Laboratory Automation. en. *Lab Med* **47,** 73–82.

211. Miler, M., Nikolac Gabaj, N., Dukic, L. & Simundic, A. M. Key Performance Indicators to Measure Improvement After Implementation of Total Laboratory Automation Abbott Accelerator a3600. en. *J Med Syst* **42.**

212. Chung, H. J., Song, Y. K., Hwang, S. H., Lee, D. H. & Sugiura, T. Experimental fusion of different versions of the total laboratory automation system and improvement of laboratory turnaround time. en. *J Clin Lab Anal* **32.**

213. Bolduc, S. *Sputum collection and transport in Africa: perspectives from Mozambique - part 1: Challenges DNA Genotek's Infectious Disease Blog* en.

214. U.N.A.I.D.S. *Global HIV & AIDS statistics — 2020 fact sheet* io.

215. Kulkarni, S. *et al.* GeneXpert HIV-1 quant assay, a new tool for scale up of viral load monitoring in the success of ARTprogramme in India. en. *BMC Infect Dis* **17.**

216. Manoto, S. L., Lugongolo, M., Govender, U. & Mthunzi-Kufa, P. Point of Care Diagnostics foHIV in Resource Limited Settings: An Overview. en. *Medicina (Kaunas* **54.**

217. May, S., Adamska, E. & Tang, J. Evaluation of Vela Diagnostics HIV-1 genotyping assay on an automated next generation sequencing platform. en. *J Clin Virol* **127.**

218. Food, U. S. & Administration, D. Evaluation of A 298 utomatic Class III Designation for Sentosa SQ HIV-1 Genotyping Assay and Associated Sentosa NGS (Next Generation Sequencing. en. *Workflow System.*

219. Raymond, S. *et al.* Performance evaluation of the Vela Dx Sentosa next-generation sequencing system for HIV-1 DNA genotypic resistance. en. *J Clin Virol* **122.**

220. Dessilly, G. *et al.* First evaluation of the Next-Generation Sequencing platform for the detection of HIV-1 drug resistance mutations in Belgium. en. *PLoS One* **13.**

221. Kingwara, L. *et al.* From Sequence Data to Patient Result: A Solution for HIV Drug Resistance Genotyping With Exatype, End to End Software for Pol-HIV-1 Sanger Based Sequence Analysis and Patient HIV Drug Resistance Result Generation. en. *J Int Assoc Provid AIDS Care* **19.**

222. Duarte, H. A. *et al.* Implementation of a point mutation assay for HIV drug resistance testing in Kenya. fr. *AIDS* **32,** 2301–2308.

223. Panpradist, N. *et al.* Near point-of-care, point-mutation test to detect drug resistance in HIV-1: a validation study in a Mexican cohort. fr. *AIDS.*

224. Panpradist, N. *et al.* Simplified Paper Format for Detecting HIV Drug Resistance in Clinical Specimens by Oligonucleotide Ligation. en. *PLoS One* **11.**

225. Chung, M. *et al.* Impact of Prior HAART Use on Clinical Outcomes in a Large Kenyan HIV Treatment Program. *Current HIV Research* **7,** 441–446. ISSN: 1570162X. https://pubmed.ncbi.nlm.nih.gov/19601781/ (Aug. 2009).

226. Zunt, J. R. *et al.* Human T-lymphotropic virus type 1-associated myelopathy/tropical spastic paraparesis: Viral load and muscle tone are correlated. *Journal of NeuroVirology* **12,** 466–471. ISSN: 13550284. https://pubmed.ncbi.nlm.nih.gov/17162662/ (Dec. 2006).

SPECIAL TERMS

API application programming interface. 87

AWL Aquarium workflow language. 86, 87, 98–100

BFGS Broyden–Fletcher–Goldfarb–Shanno. 28

BMF Biological Manufacturing File. 116–118, 120, 122, 125, 128, 130, 136

BSC biosafety cabinet. 167

CAD computer-aided design. 103, 116

CAM computer-aided manufacturing. 4, 116, 134

CAPP computer-aided process planner. 3, 4, 116, 134, 145, 146

CHO Chinese hamster ovary. 4

CRISPR-Cas9 CRISPR-associated protein 9. 3, 4

DSGRN Dynamic Signatures Generated by Regulatory Networks. 45, 59, 62, 63, 132

DSL domain-specific language. 87

GUI graphical user interface. 87

HEK human embryonic kidney. 4

HIVDR HIV drug resistance. 158, 159, 164–166

IDE integrated developer environment. 98

LIMS laboratory information management system. 90, 100, 102, 135

NGS next-generation sequencing. 165

OLA oligonucleotide ligation assay. 159

PRCC Partial rank correlation coefficient. 35

RMS reconfigurable manufacturing system. 2

RT-PCR reverse transcription PCR. 165

SD2 Synergistic Design and Discovery. 135

GLOSSARY

E. coli Escherichia coli, a gram-negative bacteria. Used as a model organism. 90, 92

S. cerevisiae Also known as Baker's yeast. Used as a model organism in research and is among the most well known eukaryotic organisms.. iv, 1, 3, 6, 8, 11, 12, 14, 16, 18, 19, 26, 29, 44, 45, 52, 59, 107, 114, 129

S. pyogenes Streptococcus pyogenes, a gram-positive pathogenic bacterium that causes.. 10

item An Aquarium software related term. A physical manifestation of a sample that exists in the laboratory. A miniprep stock is an item of a given plasmid sample. 90

job An Aquarium software related term. A batch of operations of the same type that are run concurrently by a technician following instructions generated from the operation type protocol written in Krill. 91

location wizard An Aquarium software related term. Manages location of Aquarium inventory based on their object type and current inventory. 90

object type An Aquarium software related term. A category of items that includes a name and a default location, and belongs to a particular sample type. Examples could be 'Plasmid Stock' or '400mL Bottle of Media'. 90

operation An Aquarium software related term. The basic unit of laboratory work planned in Aquarium, in which inputs are converted to outputs according to a protocol defined using the Krill protocol language. 87

plan An Aquarium software related term. A set of operations that are linked by connecting inputs and outputs. Structure forms a directed acyclic (DAG) graph of inputs, outputs, and operations.. 87

sample type An Aquarium software related term. A category of samples, such as 'Plasmid' or 'Mammalian Cell Line'. 90

sample An Aquarium software related term. A biologically unique entity, with properties defined by the needs of the user. A description of a specific plasmid is a sample. 90

Aquarium Open-source laboratory management software.. 85, 158–161, 163–166

BLAST+ Local sequence alignment tool. 152

CD4+ Also known as T helper cells, CD4+ cells play an important role in the adaptive immune system. These cells stimulates the activity of other immune cells by releasing cytokines. CD4 cell count is an indicator of immune function. In individuals living with HIV, CD4 counts are used to monitor immune health and degree of infection and often is used to determine if opportunistic infection prophylaxis is needed to treat HIV-positive individuals.. 158, 160, 163, 167

CEN.PK113-7D An industrial yeast strain based on *S. cerevisiae*. 131

GFP Green fluorescent protein. A fluorescent protein often used as a reporter for protein expression.. 150

Krill An Aquarium software related term. The Domain-Specific Language used to define protocols, a core element of an operation type. Krill extends Ruby by including methods specific for managing Aquarium objects and generating on-screen instructions for technicians. 86, 91

OLA-Simple Diagnostic paper-based assay used to diagnose HIV drug-resistance. Invented by Nuttada Panpradist at the University of Washington.. 158–160, 163–166

pUC19 pUC cloning vector. A standard DNA cloning vector.. 150

replicability Defined as obtaining consistent results by *recollecting the data* and performing the same analyses as the original study. This is a higher standard to achieve than "reproducibility" as it requires re-running the data collection procedure. 85, 86

reproducibility Defined as obtaining consistent results using the same data and code as the original study. There is some confusion of this term in that it is often used to also mean to repeat the data collection process (see term "replicability"). However in this document, this term is strictly used to mean consistence in the analysis and code rather than recollecting the data. 86

Ruby Dynamic programming language. 91

Trident Python interface to the Aquarium software.. 87

UW BIOFAB Facility at the University of Washington that provides 'science for hire' using the Aquarium-software system to manage workflows. Labor is executed by rolling undergraduate hires. 101

VITA

Justin Vrana is from Racine, Wisconsin. He earned in Bachelor's of Science in Philosophy and Chemistry and a Certificate of Computer Science from the University of Wisconsin. He is currently a PhD candidate in Bioengineering at the University of Washington. He has co-authored 10 papers in optogenetics and synthetic biology. As of 2021, he will begin a position as Senior Software Engineer as Just–Evotec Biologics in Seattle, WA.