



AFRL-RI-RS-TR-2023-068

XPLAN: DOMAIN AGNOSTIC EXPERIMENTAL PLANNING

SMART INFORMATION FLOW TECHNOLOGIES (SIFT), LLC

APRIL 2023

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2023-068 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

WILMAR W. SIFRE
Work Unit Manager

/ S /

GREGORY J. HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
APRIL 2023		FINAL TECHNICAL REPORT		START DATE SEPTEMBER 2017	END DATE OCTOBER 2022
4. TITLE AND SUBTITLE XPlan: Domain Agnostic Experimental Planning					
5a. CONTRACT NUMBER FA8750-17-C-0184		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61101E	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER R2DP	
6. AUTHOR(S) Daniel Bryce					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Smart Information Flow Technologies (SIFT), LLC 319 1st Ave North, Suite 400 Minneapolis MN 55401				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RITA 525 Brooks Road Rome NY 13441-4505			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RI-RS-TR-2023-068
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT SIFT and its subcontractor Raytheon/BBN addressed experimental planning and data representation under TA2 of the Defense Advanced Research Projects Agency (DARPA) Synergistic Discovery and Design (SD2) program. Our team played a role in designing SD2 systems architectures, developing experimental planning and knowledge representation formats and tools, integrating multiple software components within our team and across the program, and providing leadership for challenge problems and working groups. We created several metadata curation, experimental planning, and data analysis tools that facilitate increased automation and traceability of experimental metadata. In collaboration with other SD2 performers, we integrated these tools into a highly automated pipeline called the Round Trip (RT), which lowers the barrier for an experimentalist to quickly submit large experiments to a robotic cloud laboratory and automatically retrieve processed data. The Round Trip became the Test-Learn sub-loop for several Design-Build-Test-Learn loops inside SD2 challenge problems (Yeast States and Novel Chassis) and outside transition partners (AFRL and ARL).					
15. SUBJECT TERMS SD2, SynBioHub, RDF, Round Trip (RT)					
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR		18. NUMBER OF PAGES 34
19a. NAME OF RESPONSIBLE PERSON WILMAR W. SIFRE				19b. PHONE NUMBER (Include area code) N/A	

Table of Contents

List of Figures	ii
List of Tables.....	iii
1 SUMMARY	1
1.1 Experiment Tempo	1
1.2 Analysis Time	2
1.3 Knowledge Creation.....	2
2 INTRODUCTION.....	3
3 METHODS, ASSUMPTIONS, AND PROCEDURES.....	4
3.1 Starting State of Practice	4
3.2 SD2 Program Structure & Project Goals.....	5
3.3 Approach: Software Tools & Knowledge Representation	6
3.3.1 Round Trip Data Model	7
3.3.2 Round Trip Contributions	11
3.3.3 Additional Contributions.....	18
4 RESULTS AND DISCUSSION.....	21
4.1 Summary of Results	21
4.2 Detailed Round Trip Results	22
4.3 LabOP Results.....	24
5 CONCLUSIONS	25
REFERENCES.....	26
6 LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS	28

List of Figures

Figure 1: Cumulative number of Experiments over the course of the SD2 project, separated by program phase with linear fit (dashed) lines.....	1
Figure 2: Mean time to analyze data after experiment data upload by program phase of SD2	2
Figure 3: Increased knowledge sharing is correlated with overall rates of experimentation in SD2 (left), and specific tools are correlated with expansions in key forms of shared knowledge (right)	3
Figure 4: (a) Typical practice in biological collaboration involves many iterations of communication between the collaborators dispersed across multiple channels of communications, leading to confusion, mistakes, and poor tracking of information about experiments; (b) The RT automates and provides structure to human-human interactions required to plan and analyze experiments.....	5
Figure 5: The initial SD2 program structure.....	6
Figure 6: Round-Trip Software Architecture (numbered steps indicate the flow of execution)	7
Figure 7: The RT facilitates several steps in the experimental process from experiment inception through to results presentation	8
Figure 8: The RT transforms an ER into a data set by creating and merging multiple experimental runs, and producing multiple analyses.....	9
Figure 9: The Intent Parser Google Docs Plugin adds a menu to process an Experimental Request	12
Figure 10: The Intent Parser resolves common terms in the ER with the Data Dictionary and SynBioHub.....	13
Figure 11: The Project Dictionary relates common terms with their SynBioHub URIs and lab terms	14
Figure 12: Condition Space and experiment format	15
Figure 13: TASBE output converting flow cytometry measurements in arbitrary units to normalized units.....	16
Figure 14: Red intensity (RL1 A) vs Green Intensity (BL1 A), with predicted live and dead data points	17
Figure 15: Predicted correctness (i.e. intended circuit output signal) for genetic circuits in yeast.....	17
Figure 16: LabOP is a standard for describing laboratory protocols that is built upon multiple standards and approaches to laboratory science, including the UML Activity Model, SBOL, Prov-O, Autoprotocol, and Aquarium	19
Figure 17: The fluorescence prediction model generalizes over designs (columns) and conditions (row) by using a prior distribution (left) and using data to compute a posterior distribution (right).....	20
Figure 18: RT performance in SD2. (Left axis (blue, orange, and green lines): Cumulative number of Experiments over the course of the SD2 project, separated by program phase with linear fit (dashed) lines. The RT was introduced near the start of Phase 2, as indicated by the vertical black line. Phase 1 data did not closely track experiment dates and we illustrate the linear fit only. First right axis (red dotted line): Cumulative data product volume over SD2. Second right axis (gray bars and black lines): Mean time to analyze data after experiment data upload by program phase of SD2.)	21
Figure 19: Increased knowledge sharing is correlated with overall rates of experimentation in SD2 (left), and specific tools are correlated with expansions in key forms of shared knowledge (right).....	22
Figure 20: : LabOP Editor, authoring the Opentrons demo protocol.....	24

List of Tables

Table 1: Round 4.0 Controls Table (each controls table in an ER identifies measurements that the RT will annotate with control labels).....	8
Table 2: Round 4.0 Measurements Table (each row in the measurements table describes a set of expected measurements to be produced by the ER).....	10
Table 3: Expansion of the measurements table in Table 2	10
Table 4: Round 4.0 Parameters Table (the parameters table identifies the protocol id and several parameter-value pairs used to configure the protocol).....	11
Table 5: Example Plate Reader Data Table (Data tables combine measurement metadata with raw data to support analysis tools).....	11
Table 6: Summary of the On-The-Loop Experimental Campaign (Duration and Turnaround are measured in days. The round 2.0 turnaround is measured with respect to round 1.0.).....	22

1 SUMMARY

SIFT and its subcontractor Raytheon/BBN addressed experimental planning and data representation under TA2 of the Defense Advanced Research Projects Agency (DARPA) Synergistic Discovery and Design (SD2) program. Our team played a role in designing SD2 systems architectures, developing experimental planning and knowledge representation formats and tools, integrating multiple software components within our team and across the program, and providing leadership for challenge problems and working groups.

We created several metadata curation, experimental planning, and data analysis tools that facilitate increased automation and traceability of experimental metadata. In collaboration with other SD2 performers, we integrated these tools into a highly automated pipeline called the Round Trip (RT), which lowers the barrier for an experimentalist to quickly submit large experiments to a robotic cloud laboratory and automatically retrieve processed data. The Round Trip became the Test-Learn sub-loop for several Design-Build-Test-Learn loops inside SD2 challenge problems (Yeast States and Novel Chassis) and outside transition partners (AFRL and ARL).

1.1 Experiment Tempo

Our innovations offer radical improvements to the conduct of extended experimental campaigns. In the past, such campaigns involved either substantial delays before follow-on experiments or suffered from imperfect planning of follow-ons. Our developments offer quickly-updated, data-driven plans to guide campaigns. The SD2 program was divided into three phases, where Phase 1 developed automation for pre-existing data sets (developed as part of SD2) and to support proof of concept demonstrations, Phase 2 expanded automation to become agnostic to application domains, and Phase 3 adapted automation to support transition partner use cases.

Figure 1 summarizes improvements to experiment tempo through our project, plotting experiment count against time. Phase 1 experiments are shown as a gray box to indicate that experimental data were uploaded sporadically to the SD2 platform, with several large batches uploaded months after the data were collected. This happened because the automation was being developed concurrently with experiment execution. **In Phase 1, the experimental tempo was 0.19 experiments per day**, using a single experimental protocol.

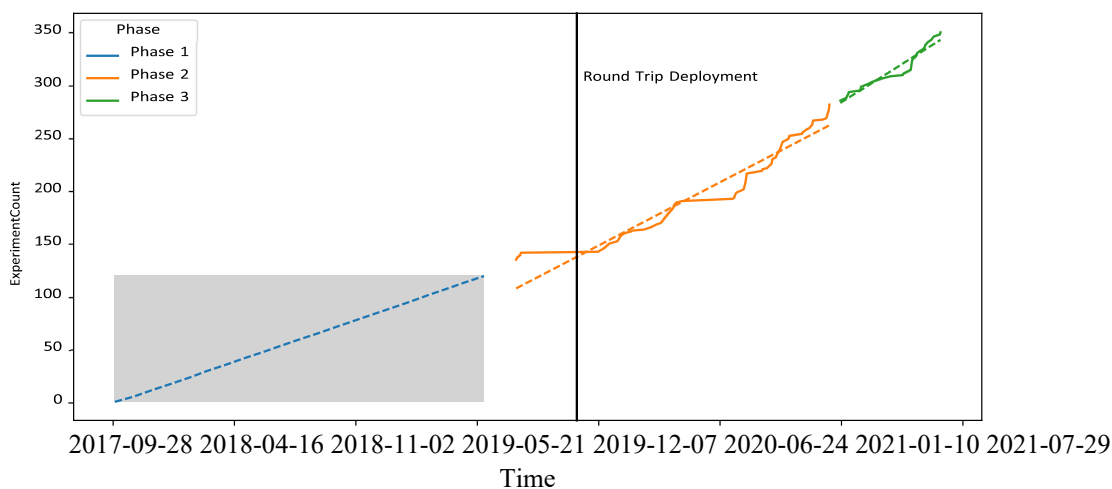


Figure 1: Cumulative number of Experiments over the course of the SD2 project, separated by program phase with linear fit (dashed) lines

The RT was introduced at the start of Phase 2 (indicated by the vertical black line in Figure 1), and deployed in November of 2019. Note that the plot in Phase 2 shows a plateau in mid-2020. This was due to an interruption in sample creation caused by COVID-19 health mandates. The ingest date depicted by Figure 1 over this period is much more closely aligned with the experiment date: data were promptly uploaded. The experiment frequency

afforded by RT was **0.29 experiments per day over Phase 2, and 0.39 experiments per day over Phase 3; a 2.05x increase in experiment tempo over Phase 1.**

In addition to performing a higher raw number of experiments, the RT also supported a more varied *mix* of experiments. The experiments in Phase 1 were all done with a single experimental protocol. In Phase 2, experiments came from four experimental protocols and in Phase 3, **five protocols**. So, at the same time the experiment tempo was increasing, so was the complexity of the set of experiments.

1.2 Analysis Time

Figure 2 shows the increase in speed of data analysis: plotting the mean time to analyze the data produced by an experiment after those data were uploaded to the SD2 platform. RT deployment in Phase 2 with manually-triggered data standardization and analysis led to a **4.20x speed-up in Phase 2** over the non-standardized approach in Phase 1. Introducing automatically-triggered data analysis led to a **30.10x additional speed-up** (126.10x over the nonstandardized analysis). The 4.20x speed-up from Phase 1 to Phase 2 comes from the RT's automated metadata creation and manually triggered, yet standardized analysis. The final 30.10x speedup from Phase 2 to Phase 3 was achieved by automating all metadata creation, data standardization, and analysis. **Overall, from Phase 1 to Phase 3 the SD2 Round Trip elements contributed to a 126.10x speed-up data analysis.**

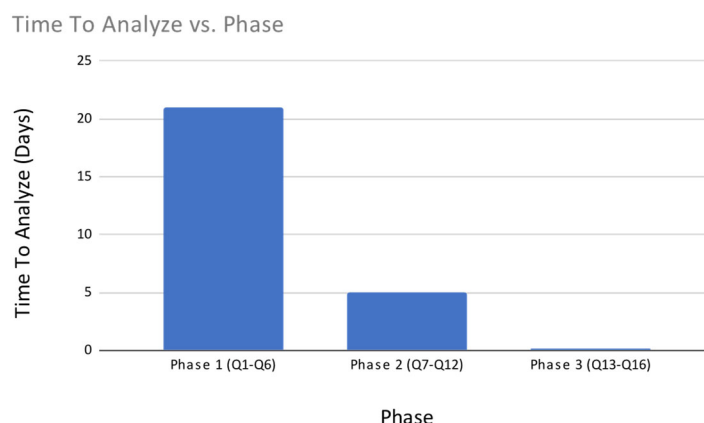


Figure 2: Mean time to analyze data after experiment data upload by program phase of SD2

1.3 Knowledge Creation

Our tools improved information sharing across the SD2 community and the usefulness of that information to community members. In turn this supported the progress of experimental workflows, most notably in the Yeast States, Novel Chassis, and Cell-Free Riboswitches challenge problems.

Program knowledge sharing expanded greatly. Knowledge of protocols and the entities in them was stored as RDF triples in SD2's SynBioHub web site. The SD2 SynBioHub currently has data on 2,874 experiments, 3,549,237 components, and 15,884 modules. This data set is made up of 22,872,306 triples that are organized into 524,377 collections. Moreover, measurement of key knowledge collections shows that progress on knowledge management (KM) tools correlates with increases in knowledge sharing and productivity. Figure 3(left) shows the correlation between knowledge sharing and data production, with both moving much more quickly in the second half of SD2 after our KM tools began to be used. Figure 3(right) shows that expansion of SynBioHub content are correlated with the dates at which each of these key tools were released.

SIFT and Raytheon/BBN developed core automation and data representation methods and software that helped to achieve the SD2 program goals of increasing the rate of data production and the quality of data in laboratory science.

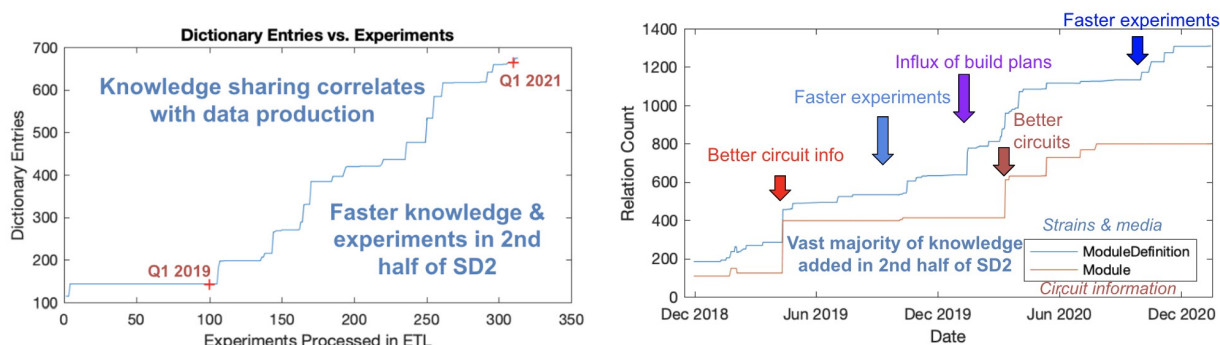


Figure 3: Increased knowledge sharing is correlated with overall rates of experimentation in SD2 (left), and specific tools are correlated with expansions in key forms of shared knowledge (right)

2 INTRODUCTION

This final report describes SIFT and BBN/Raytheon’s application of knowledge representation and automated planning tools to the challenge of increasing the throughput of experimentation in data-intensive laboratory science. The report is intended for an audience with a STEM background, and discusses topics in computer science and synthetic biology.

Subject: Research in laboratory-based science, such as synthetic biology, relies upon evaluating many designs (e.g., purpose-built genetic circuits, proteins designed to bond to pathogens) in many conditions (e.g., various growth media and different incubation temperatures) with several protocols. Through potentially many design iterations, a successful design can provide a high economic or strategic advantage (e.g., as a sensing or chemical synthesis platform, a therapy, etc.). Improving the throughput of this process, from requirements specification to viable designs, requires automation. Automation can replace or reduce inefficient or error-prone human involvement, provide access to designers without the resources to set up their own laboratories, help address the replication crisis, etc. However, effective automation requires data and metadata that describe the experimental context, and procedures (experimental protocols).

Purpose: In an age of automation, the throughput of laboratory science is limited by the ability of stakeholders to collaboratively and effectively develop a shared experimental intent and to harness automation for experimentation. Developing a representation of experiments that is both human and machine interpretable can support both collaboration and automation. However, designing experiment representations poses a number of challenges, including developing effective human-machine interfaces, protecting against and repairing user error, providing flexibility for terminological mismatches, and supporting extensibility to new experiment types.

Scope: In this report, we discuss how we addressed these challenges within the SD2 program. The primary focus of our work is data representation and experimental planning within the SD2 Round Trip (RT) architecture [8]. The RT automates many of the key steps in the Test and Learn phases of a Design-Build-Test-Learn loop for high throughput laboratory science. The primary contribution of the RT is to assist with and otherwise automate metadata creation, curation, standardization, and linkage with experimental data. The RT’s focus on metadata supports fast, automated, and replicable analysis of experiments, as well as experimental situational awareness and experiment interpretability.

Outline: In the following sections, we discuss the methods, assumptions, and procedures used in our work, present results, and conclude with a discussion of our main findings.

In the methods, assumptions, and procedures section, we highlight the major software components and data representations that we contributed to the Round Trip to speed up the design and analysis of experiments by two orders of magnitude over prior *ad hoc* methods. The contributions support a number of experimental protocols and experiment types, demonstrating the Round Trip’s breadth and extensibility. We describe both an illustrative use case using the Round Trip for an on-the-loop experimental campaign, and overall contributions to reducing experiment analysis time and increasing data product volume in the SD2 program.

We present results that measure the knowledge represented by the discussed tools, the experimental throughput improvements due to our tools, and down-stream automation efficiency supported by the developed data and tools. The results focus on several hundred experiments run over the course of four years in the SD2 program, and pertain to improvements in both human and automated processes relating to these experiments.

We conclude by interpreting how data representation and automated planning tools improve not only the process of experimentation, but also how they enable domain agnosticism through their adaptability to new designs, protocols, and analytical methods. The results demonstrate that the products of this research project are capable of transition to a variety of industry or government use-cases.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

In this section, we describe the methods, assumptions, and procedures used to conduct the project. To provide context for our contributions, Section 3.1 describes the state of practice for experimental planning, execution, and analysis at the time of the start of the SD2 program. Section 3.2 describes the SD2 program structure and outlines the project goals in terms of advances over the starting state of practice. Section 3.3 describes our approach in detail, which is based upon multiple new methods for data representation and reasoning as it pertains to experimental planning, execution, and analysis. In particular, Section 3.3.1 provides background on the RT data model, Section 3.3.2 describes the RT and our primary contributions, and Section 3.3.3 describes additional contributions that supported the overall SD2 program goals but were not tightly integrated with the RT.

3.1 Starting State of Practice

When we began SD2, the lack of automated tools for experimental design, execution, and analysis was a key barrier to quickly evaluating biological designs. We faced challenges describing experimental plans in a format that laboratories could execute. It was also difficult for labs to link experimental data to metadata and plans, making this an expensive manual undertaking — evidenced by the SD2 Q0 Hackathon. We found that metadata describing contents, conditions, and the context of experimental samples is key to gaining insights from experimental data, but acquiring, tracking, and maintaining metadata is tedious, expensive, and error-prone. Data sharing was a difficult, bilateral process requiring negotiation between every pair of organizations or experimental phases, incorporating new tools into pipelines was *ad hoc* and time-consuming, and data was hard to trace back to genetic designs and experimental plans.

Figure 4 illustrates the state of practice at the start of SD2. An Experimentalist must communicate (over several weeks) with a Laboratory Technician over email, phone calls, site visits, and shared lab notebooks, to describe their experiment. The Laboratory Technician then takes several days to produce a fully detailed plan for executing the experiment, given the resources available in their laboratory. For highly automated laboratories, this may be an underestimate of preparation time, which may involve special-purpose programming of laboratory equipment.

When prepared and rendered executable, this experiment plan is then executed, producing a data set over the course of several more days. Normally, the Experimentalist and Laboratory Technician then describe the experiment, its intent, and the data produced to a Data Scientist over several days to weeks (again through email, phone, site visits, and lab notebooks). In SD2’s Q0 Hackathon, the data was provided to Data Scientists in a raw format with minimal metadata. The process ends with a Data Scientist producing their analysis. The Q0 Hackathon vividly demonstrated the limitations of this process, as participants struggled to understand the context and formatting of the data provided. In practice, the total process can take weeks to months, and is prone to confusion between participants, mistakes in communication, and poor tracking of information about experiments.

Synthetic biologists are increasingly attempting to improve on this practice via various engineering methodologies, often structured around a Design-Build-Test-Learn (DBTL) loop. While there have been many improvements, they have mainly focused on the technical aspects of automation rather than addressing the challenges of collaborative science. For example, much work on the Test (i.e., experimentation) phase includes automation in the form of Laboratory Information Management Systems (LIMS) and laboratory robots (e.g., Strateos [5], Aquarium [12], Emerald Cloud [4], and the Edinburgh Foundry [3]). While important for organizing repeatable experiments within a laboratory, these improvements to the Test phase do not address collaboration between stakeholders. Furthermore, using such laboratory automation tools can trap their users in “walled gardens,” as each has its own means of configuration that is incompatible with other systems.

The Build phase includes automation at gene foundries such as JGI [2] and MIT-Broad [1], but as in the Test phase, these techniques do not directly address collaboration. Large companies often develop internal, proprietary solutions,

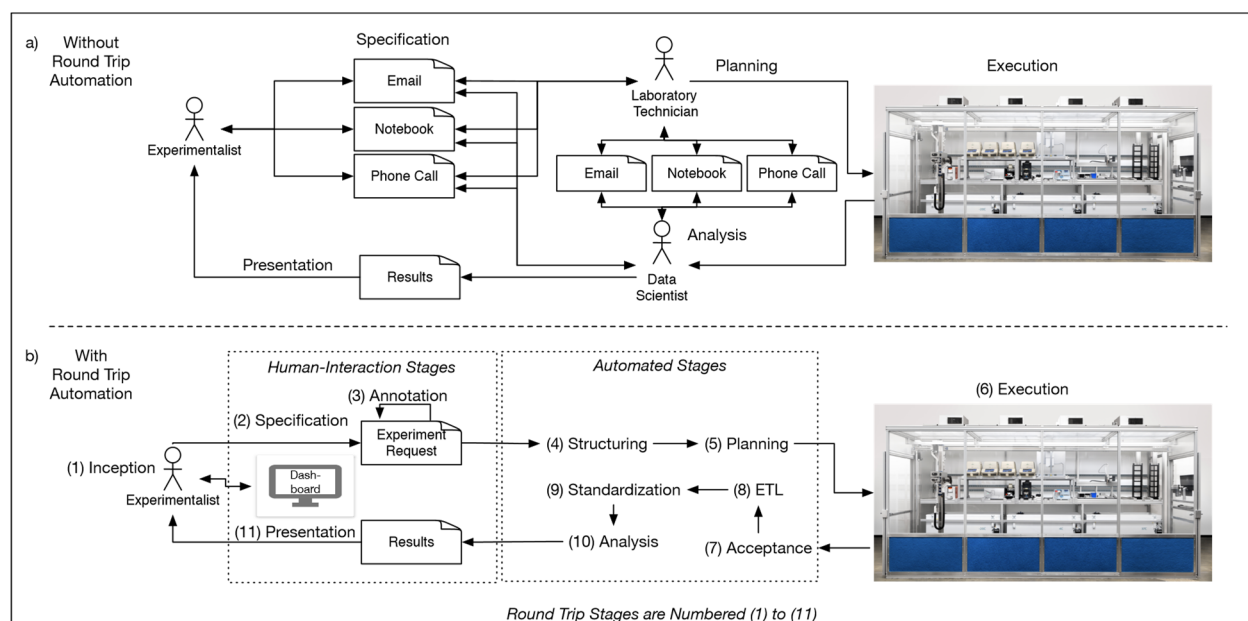


Figure 4: (a) Typical practice in biological collaboration involves many iterations of communication between the collaborators dispersed across multiple channels of communications, leading to confusion, mistakes, and poor tracking of information about experiments; (b) The RT automates and provides structure to human-human interactions required to plan and analyze experiments

but these are often highly specialized, not typically available to the larger community, and do not typically facilitate collaboration outside of the specific organization.

Finally, some highly automated DBTL systems, such as ESCALATE [15] and the robot scientist [11], address specific workflows (e.g., mapping a space of chemical reaction parameters), and are able to achieve tight integration of each step in the loop. With a goal of moving toward this level of integration and automation while also providing broad support for many different forms experiments and analyses, it is difficult to fully remove humans from the loop. As we will demonstrate, however, it is possible to accelerate experimentation by reducing the requirements on human participants.

3.2 SD2 Program Structure & Project Goals

The primary aim of SD2 was to help scientists, lab personnel, and data scientists quickly develop a common understanding of experiments in order to speed up experimentation, and render it more replicable. The SD2 project was structured around several challenge problems and technology transition demonstrations, many of which used the RT to conduct experiments. SD2 involves approximately one hundred researchers across multiple universities, companies, and government laboratories. The organizations addressed the five technical areas of the program as part of their contributions to the challenge problems. The technical areas included machine learning and data science for discovery (TA1), experimental planning and design (TA2), experiment execution (TA3), technology platform (TA4), and challenge problem formulation (TA5, in the service of analyzing *ad hoc* socio-technical organizations). Figure 5 illustrates the initial SD2 program structure in terms of the relationships between technical areas TA1-TA4. The program structure emphasized the need for high levels of automation, to be enabled by developing novel software and data infrastructure.

Our team contributed most heavily to the technical area on experimental planning and design (TA2), but also made contributions to the overall program by addressing cross-cutting software infrastructure and data requirements. As such, we contributed to multiple challenge problems that exercised the capabilities of the software infrastructure, including the Yeast States, Novel Chassis, and Cell-Free Riboswitches challenge problems.

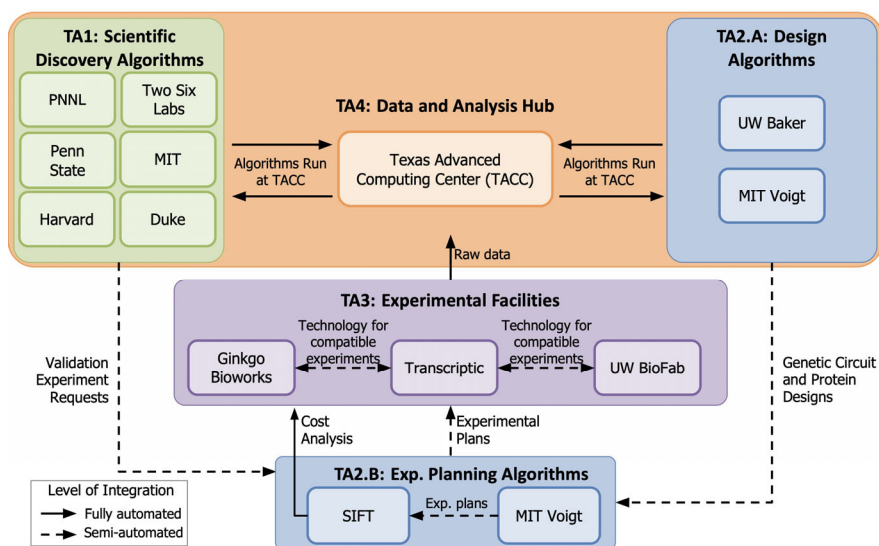


Figure 5: The initial SD2 program structure

The SD2 program goals included: increased data production for laboratory science and reduced time and cost for producing data sets that have scientific significance. The goal to increase data production was met with challenges to increasing the volume of experiments run to produce data, and improving automated data handling. We addressed both challenges by developing the Experimental Planner (XPlan) system to automate experiment planning and by data representation techniques and tools to support experiment automation, as well as data analysis. The goal to reduce the time and cost associated with high-quality data set production presented challenges in terms of data interpretability and automating previously manual processes. We improved data interpretability by developing methods that plan what data is expected to be produced, and generate metadata for each raw data point. We automated the experimental planning and submission process with XPlan, which previously required manually planning an experiment, given available laboratory resources.

3.3 Approach: Software Tools & Knowledge Representation

Our approach to developing experimental planning and data representation techniques, involved creating several software packages. These packages were mainly developed in support of the RT architecture, illustrated in Figure 6. The architecture identifies the major, highly-integrated software components developed by the SD2 program, and characterizes the end of program software architecture that was initially planned by DARPA in Figure 5. It automates the process of experiment design through to final analysis, as indicated by the numbered steps (1)-(11) (described in Figure 7).

The RT focuses upon the Test-Learn aspect of DBTL for two main reasons. First, the Design-Build phases of DBTL are frequently bespoke for a particular application, such as those addressed by SD2 (genetic circuits in yeast, novel host organisms, and cell-free riboswitches). However, these applications share similar Test-Learn screening needs that can be addressed by RT. Second, the RT developed out of an effort to design robust genetic circuits. As part of assessing robustness, it is necessary to selectively characterize the operational envelope of a circuit by screening in a number of experimental conditions. Covering the condition space requires many Test-Learn “inner” loops within a single DBTL “outer” loop. For these reasons, we developed the RT to be highly integrated for Test-Learn and have made it extensible for easy incorporation within application-specific DBTL loops.

Figure 4b illustrates the state of practice provided by the RT and exercised by the SD2 project. The stages numbered from (1) experiment inception to (11) results presentation remove the bottle-necks of the prior state of practice. The RT helps to situate human interaction where intuition and expertise is needed, and automation where tedium and propensity for human error can arise. Figure 6 illustrates the RT software architecture, also numbered by stage. The

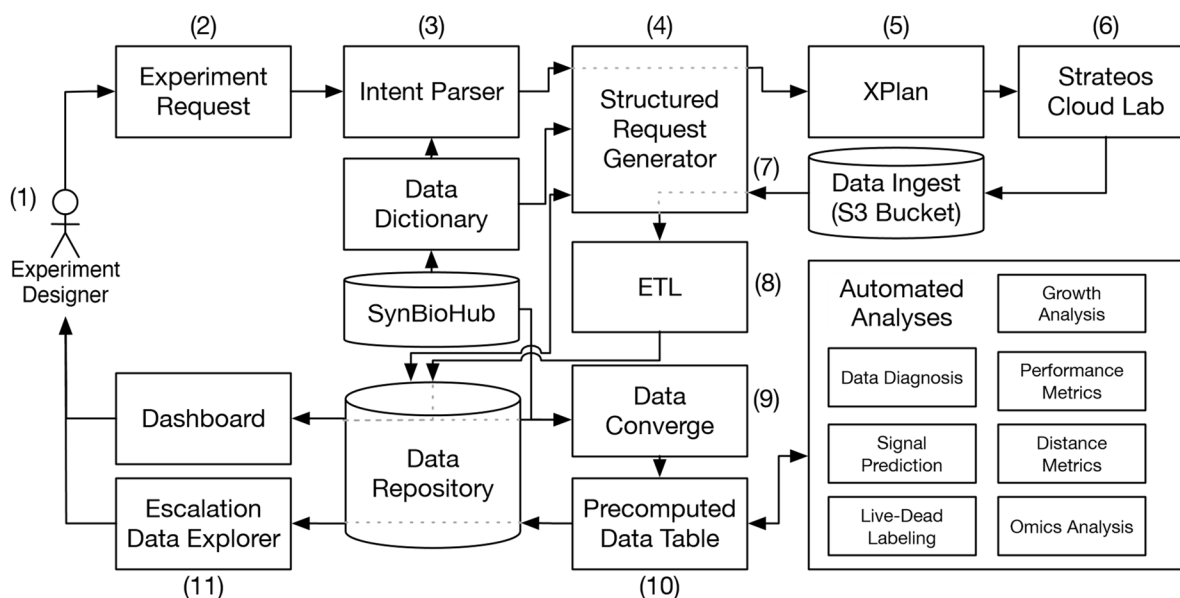


Figure 6: Round-Trip Software Architecture (numbered steps indicate the flow of execution)

stages proceed left-to-right on top, steps (1)-(5), to create an automation-ready experiment, executed in step (6). The RT attaches metadata to raw data as it returns from right to left on the bottom, steps (7)-(11).

We directly contributed to steps (2), (3), (5), (8), and (10), as follows. Step (2) involves formulating an Experiment Request (ER), a semi-structured document that describes the experiment, both in prose (e.g., hypothesis, rationale, and intended outcomes) and more-formal descriptions of the data points that are intended to be produced. We set requirements on the data point descriptions so that our downstream software tools could derive their inputs. We developed the Intent Parser and Data Dictionary tools, step (3), to convert common names used in the ER into SynBioHub references. This allowed the experiment designer to use familiar terms (e.g., "W303" or "SC Media") defined in the Data Dictionary and for the downstream automation to link metadata present in SynBioHub (e.g., DNA sequences or media recipe ingredients) to each data point. Step (5) applied our XPlan component to plan how to generate the requested data points and submit laboratory-specific requests to generate some or all of the data points. Step (8), Extract-Transform-Load (ETL), preprocesses raw data produced by the laboratory. Among the many software tools involved in ETL, we adapted the TASBE software component for preprocessing flow cytometry data. We contributed to step (10), the Analysis, by developing several data analysis tools that were run automatically on all datasets. Specifically, we developed the Growth Analysis, Signal Prediction, and Live-Dead Labeling analysis components.

In the following, we describe the RT data model, components that were essential to the RT, and descriptions of additional software that was developed as part of SD2.

3.3.1 Round Trip Data Model

To better contextualize our contributions, we review the RT data model. The data model (a subject of much of our SD2 work) influenced the design of our software components. The primary input is an ER, and the primary output is a data set containing both raw data and analyses annotated with strain and experimental condition metadata. Figure 8 illustrates the RT data flow from ER to data set. The flow involves determining how the ER maps onto multiple protocol launch runs, collecting and consolidating the run data, and then producing a full data set that includes analysis results.

The ER is a structured, natural language document accessible to manual construction by an experiment designer. The structured elements in the ER (developed in RT steps 1-4) define a tuple (C, M, P) where C is a set of controls, M

is a set of measurements, and P is a protocol. Each control $c \in C$ identifies a measurement $m(c) \in M$ that will be used in analysis, such as the measurement characterizing the minimum or maximum Green Fluorescent Protein (GFP).

1. **Experiment Inception:** The experiment designer determines the hypothesis and experimental method in conjunction with a data scientist.
2. **Specifying an Experiment Request:** The experiment designer, starting with a “notional” experiment in mind, authors a semi-structured ER document. The ER includes the strains, reagents, and protocol needed, along with any hypotheses it will validate.
3. **Annotating the Experiment Request:** The RT annotates the ER with hyperlinks from experiment terms to new or existing SBOL[16] definitions stored in SynBioHub [14] through a Data Dictionary.
4. **Structuring the Experiment Request:** Linked constructs and experiment tables are converted into a structured request, formally defining a set of minimal requirements for the experiment.
5. **Experimental Planning:** The RT uses the XPlan experimental planner to expand the structured request into a set of experimental run launch requests that configure a machine-executable laboratory protocol.
6. **Experiment Execution:** The RT dispatches the experiment to the lab (Strateos), the lab runs the experiment and uploads experiment measurement data and traces.
7. **Acceptance:** The RT compares the expected measurements with the actual measurement data and traces to identify possible discrepancies (e.g., missing, duplicate, or unanticipated measurements).
8. **ETL:** The RT conducts extract, transform, and load steps to ingest data files from the lab into a data repository.
9. **Standardization:** The RT converts metadata in the data repository to a consistent cross-experiment format and aligns it with data to produce AI-ready data tables.
10. **Analysis:** The RT runs several automated analyses on the standardized data tables to produce commonly required information, such as growth rates, reporter fold change, and performance statistics.
11. **Presentation:** The RT provides interactive visual plots and summaries.

Figure 7: The RT facilitates several steps in the experimental process from experiment inception through to results presentation

The measurement set M , made of both the controls and experimental samples, represents the metadata describing the data points expected by the ER. The protocol describes the method by which the ER will generate the data points.

Table 1: Round 4.0 Controls Table (each controls table in an ER identifies measurements that the RT will annotate with control labels)

Table 1: Control				
Control Type	Strains	Channel	Contents	Timepoint
EMPTY VECTOR	W303		SC Media	8 hours
HIGH FITC	UWBF 6390	BL1-A	SC Media	18 hours

Tables 1, 2, and 4 illustrate excerpts from three structured sections of an ER, respectively corresponding to C , M , and P . The contents of these tables include values from Round 4.0 (the final round) of the multi-round experiment campaign to screen yeast circuits.

The experiment designer will specify the controls tables (Table 1) to label specific measurements with control type labels. Down-stream analysis tools may then use these labels to configure their analyses. In the Round 4.0 experiment, the controls are the reference high (logical true) and low (logical false) output for the twenty-four strains implementing one of four truth table rows of six logic circuits (e.g., that a strain implements the NOR(true, true) = false rule of the NOR circuit). The controls are denoted by two strain identifiers: W303, the low control that is the wildtype background strain, and UWBF 6390, the high control that is the same background strain with a constitutively expressed GFP. The controls are measured while grown in multiple different media and at different timepoints. The controls table identifies which timepoints are used for the reference GFP values.

Table 1 is “Table 1” of several controls tables (not shown), that are cross-referenced in the first column of the measurements table illustrated in Table 2. The controls table maps control type labels to measurements (by strain and conditions). That is, any measurement that is consistent with the description of a control type label will receive the label. The controls table also identifies, in the case of flow cytometry data, the channels that correspond to the control label (e.g., the BL1-A channel corresponds to the HIGH FITC label).

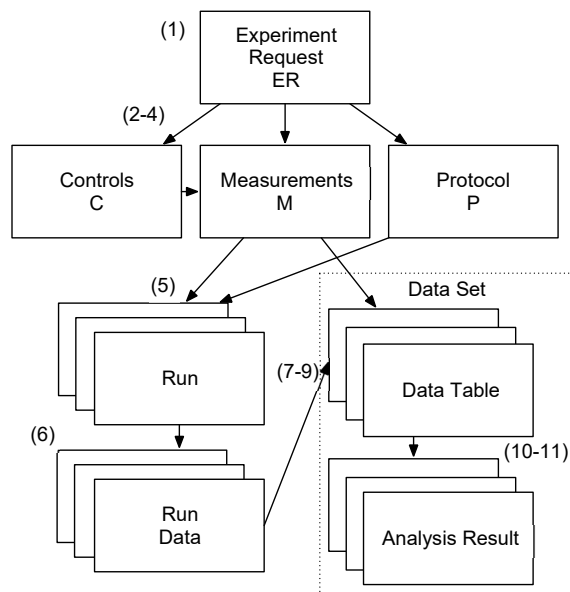


Figure 8: The RT transforms an ER into a data set by creating and merging multiple experimental runs, and producing multiple analyses

The control labels support down-stream automated analysis. For example, one RT-integrated analysis, called signal prediction, uses control type labels for flow cytometry measurements to identify training data and class labels. It then learns a model to predict whether a flow cytometry event corresponds to a low control (EMPTY VECTOR) or a high control (HIGH FITC). With these predictions, the analysis result can be used to calculate the quality of each logic circuit in the Round 4.0 experiment (e.g., the proportion of events reporting the correct circuit output).

Table 2: Round 4.0 Measurements Table (each row in the measurements table describes a set of expected measurements to be produced by the ER)

Control	Media	Measurement-Type	File-Type	Replicate	Strain	Timepoint	Temperature
Table 1	SC Media	PLATE READER	CSV	6	UWBF .6390, W303	0, 8, 12, 16, 18 hour	30 celsius
Table 4	SC Media	FLOW	FCS	6	UWBF .6390, W303	8, 12, 16, 18 hour	30 celsius
...	...	...	...	...	...	...	...
	SC Media, rich media, high osm .media	PLATE READER	CSV	3	UWBF6389, UWBF7375, UWBF8225, ...	0, 8, 12, 16, 18 hour	30 celsius
	SC Media, rich media, high osm .media	FLOW	FCS	3	UWBF6389, UWBF7375, UWBF8225, ...	8, 12, 16, 18 hour	30 celsius
...	...	...	...	...	...	...	...

The experiment designer will also specify a measurements table (Table 2) to identify every measurement that the experiment shall produce. In the Round 4.0 experiment, the experiment designer requests measurements of three types of media (possible sources of variation), with plate reader (at five timepoints) and flow cytometry (at four timepoints). The measurements include both controls (six replicates) and the strains (three replicates) implementing the logic circuits (e.g., the strains UWBF 6389, UWBF 7375, and UWBF 8825, among a total of 24 strains).

Table 2 lists a “Control” column, followed by several metadata columns. The “Control” column identifies which controls table will be used to label measurements described by a row in the measurements table because each row in the measurements table corresponds to multiple measurements. The measurements described by a row correspond to the cross-product across columns within the row. For example, the first row in Table 2 defines:

$$\{\text{SC Media}\} \times \{\text{PLATE READER}\} \times \{\text{CSV}\} \times \{1, 2, 3, 4, 5, 6\} \times \{\text{UWBF 6390, W303}\} \times \{0, 8, 12, 16, 18\} \times \{30\}$$

where the low and high controls are grown in SC Media and measured by plate reader at hours 0, 8, 12, 16, and 18 at 30C. These sixty data points correspond to the rows of an expanded table with one measurement per row, as in Table 3.

Table 3: Expansion of the measurements table in Table 2

Media	Measurement-Type	File-Type	Replicate	Strain	Timepoint	Temperature
SC Media	PLATE READER	CSV	1	UWBF 6390	0 hour	30 celsius
SC Media	PLATE READER	CSV	1	UWBF 6390	8 hour	30 celsius
SC Media	PLATE READER	CSV	1	UWBF 6390	12 hour	30 celsius
...	...	...	...	...	...	...

Table 4: Round 4.0 Parameters Table (the parameters table identifies the protocol id and several parameter-value pairs used to configure the protocol)

Parameter	Value
Protocol	TimeSeries
Inoculation volume	10 microliter
Inoculation media volume	700 microliter
...	...

The protocol P defines a lab-specific identifier for the protocol and a set of parameter-value pairs for the protocol. The protocol identifier maps to an opaque protocol supported by the laboratory. For example, the Round 4.0 experiment uses the Strateos TimeSeries protocol to generate the plate reader and flow cytometry measurements. Strateos provides a JSON-based protocol schema that the RT planning component must follow to launch an experimental run. The RT describes the interface to each protocol schema with a number of common terms for the parameters (resolved by its Data Dictionary). Table 4 lists a subset of the common term parameter-value pairs needed to configure the TimeSeries protocol. The RT determines the remaining parameter-value pairs by planning the experiment over a number of protocol runs. For example, the measurements encoded by Table 2 must be generated by at least three distinct protocol runs (at least one per media type). Each run will extend the parameters table values with run-specific parameter-value pairs that are determined by the RT planner in step 5 of the RT process. For example, runs may use one of the values {SC Media, rich media, high osm media} for the Media parameter.

The lab will execute each protocol run and generate a set of measurements in step 6 of the RT process. The lab will upload the data and lab-specific metadata for a set of measurements to the RT. In steps 7 to 9 of the RT process, the RT maps the measurements produced by the lab to the measurements in M . The RT will then process the measurements and produce a data set. The data set includes many data products, which result from several processing stages. Most notably, the RT produces a data table for each sample and measurement type. For example, Table 5 lists raw optical density and fluorescence measurements for samples in M . In steps 10 to 11 of the RT process, the analysis tools consume the raw measurement tables to produce additional analysis results. For example, the growth curve analysis tool aggregates the optical density values over the timepoints to calculate a growth rate. Its output is another table that eliminates the Timepoint column and adds a Growth Rate column. The RT analyses either aggregate over columns such as timepoints or reagent concentrations to compute new columns containing their outputs, or compute per data point output values that extend the table columns.

Table 5: Example Plate Reader Data Table (Data tables combine measurement metadata with raw data to support analysis tools)

Media	Replicate	Strain	Timepoint	Temperature	Optical Density	Fluorescence
SC Media	1	UWBF 6390	0 hour	30 celsius	0.258	20.491
SC Media	1	UWBF 6390	8 hour	30 celsius	0.260	26.066
SC Media	1	UWBF 6390	12 hour	30 celsius	0.373	38.534
...	...	...	...	...	...	...

3.3.2 Round Trip Contributions

In this section, we describe the most prominent elements that we contributed to the RT, including the Intent Parser, Project Dictionary, XPlan, TASBE, Live-Dead Labeling, Signal Prediction, and Growth Analysis. In addition to these software components, we contributed cross-cutting data representation tools and techniques, including pySBOL, SynBioHub plugins, and the OPIL lab interface standard. We describe each component, as follows.

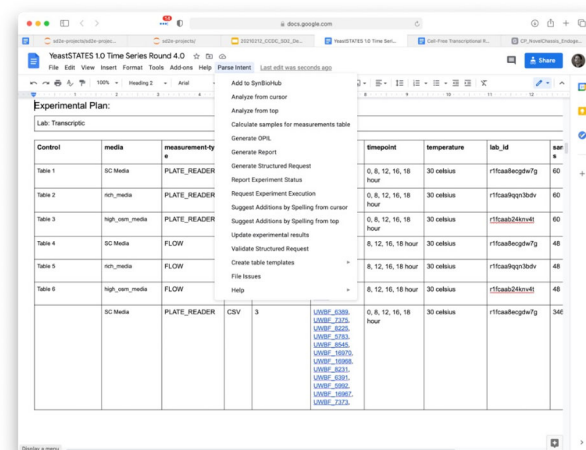


Figure 9: The Intent Parser Google Docs Plugin adds a menu to process an Experimental Request

Intent Parser: The Intent Parser enables a user to submit an experiment through the Intent Parser add-on to the experiment request Google doc (Figure 9). The Intent Parser identifies terms in the ER, flags them for annotation, and provides possible quick-fix actions. The first step to submission involves replacing strings with common terms with terms hyperlinked to their SynBioHub definitions in SBOL (Figure 10). Upon submission, the semi-structured ER is converted into a Structured Request (SR). The SR is a stand-alone JSON representation of the ER that relates the Control Tables, Measurement Table, and Parameters Table, along with hyperlinks for all terms.

The Intent Parser, Structured Request Generator (Structured Request Generator), and Data Dictionary coordinate to populate the Structured Request. The Intent Parser invokes the Structured Request Generator, sending the annotated ER as input. The Structured Request Generator populates the SR with an expansion of the Measurement Table into a list of measurements, and further annotates the SR with laboratory specific terms. While the ER contains common terms that are hyperlinked to their SBOL representations in SynBioHub, the lab chosen for experiment execution will have their own terms, often opaque LIMS identifiers. The Data Dictionary labels synonyms of common terms with those of the laboratory technician stakeholder. (An added benefit is the ability to re-purpose ERs for execution at multiple laboratories because the Data Dictionary resolves the terms.) With a complete SR, the Structured Request Generator submits the experiment for planning.

Project Dictionary: The Project Dictionary (Figure 11) harmonized data across organizations, as noted above. The Data Dictionary lists *common terms* and synonyms, unique SynBioHub URIs defining the terms, and laboratory-

	A	B	C	D
1	Write chemicals and local name below: the dictionary will automatically generate "stub" SynBioHub objects, link the URIs in, and mark them with local names			
2	Common Name	Alias Names	Type	SynBioHub URI
3	UWBF_XOR_11		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_16967/1
4	UWBF_OR_01		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_5783/1
5	UWBF_XNOR_00		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_7300/1
6	UWBF_AND_11		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_7374/1
7	UWBF_NAND_11		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_8542/1
8	UWBF_XOR_00		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_16970/1
9	UWBF_NAND_00		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_8544/1
10	UWBF_OR_11		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_5992/1
11	UWBF_NOR_10		Strain	https://hub.sd2e.org/user/sd2e/design/UWBF_6389/1

Figure 11: The Project Dictionary relates common terms with their SynBioHub URIs and lab terms

condition space is a list of factors. Each factor has a domain of values, a data type (dtype), a name, an internal lab name, an object type (used for deserialization), and factor type. Figure 12 lists a media factor with four possible values in its domain. The values are string dtypes. The value for this factor maps to the Strateos protocol name “growth media 2”. The media is a Condition (other otypes include DesignElement, such as strain). The factor type (ftype) can be “experiment”, “batch”, or “sample”. The factor type is used while constructing sample batches to determine which factors must be held constant among sample groups. For example, a “batch” factor must be the same for all samples in a batch. Strateos requires each sample in a batch (96 well plate) has the same media.

The batches are a list of batch ids with sample sizes. XPlan also supports an auto generated set of batches. Batch creation interacts with the default conditions specified. For example, a finite batch specification, such as that showing in Figure 12, will be filled with samples consistent with the default conditions (adding replicates as needed to fill the batches). In an auto generated set of batches, XPlan will not add replicates beyond those consistent with the default conditions to minimize the number of batches.

The defaults include three types of configuration data: constants, parameters, and conditions. Constants are values used by XPlan to create experiments, and parameters are values passed along to the experiment specification used by the lab. The conditions are a specification of an experiment. In contrast with the condition space (which is the envelope of all possible samples), the conditions are a subset of the condition space for the experiment. XPlan will generate an experiment consistent with both the condition space and conditions. Different experiments can hold the condition space constant, but vary the conditions. The conditions specify the experiment using a disjunctive list of factor constraints. Each factor constraint is a conjunctive list over factors of disjunctive values. In logic, each factor constraint is in conjunctive normal form. I.e., each sample must satisfy at least one factor constraint. A sample satisfies a factor constraint if it assigns each factor in the factor constraint a value in the list of allowed values. In Figure 12, a sample will satisfy the shown factor constraint if it assigns “media = high osm media” and “od = 0.002”. When the conditions are empty or partial, XPlan plans samples consistent with the constraints. In such cases, XPlan uses its model of certainty of the condition space to prioritize samples.

XPlan encodes this decision problem as a Satisfiability Modulo Theories (SMT) problem [6], and applies the z3 solver [9]. The SMT problem seeks to assign a set of variables that satisfy several constraints, including: each requested measurement is produced by an aliquot of a container of a batch, each container in a batch obeys batch constraints (e.g., use the same temperature because a batch occupies the same incubator), each aliquot in a column has the same reagent concentrations (e.g., if the protocol uses multi-tip pipettes that address columns), and each aliquot of a container produces the same measurements (e.g., a plate read at four hours, and flow cytometry reading at eight hours).

```
{ "challenge_problem": "YEAST_STATES",
  "experiment_id": "2019_10_13_20_29_20",
```

```

"condition_space": {
  "factors": {
    [...]
    "media": {
      "domain": [ "standard_media", "rich_media", "slow_media", "high_osm_media"],
      "dtype": "str",
      "name": "media",
      "lab_name": "growth_media_2",
      "otype": "Condition",
      "ftype": "batch"},
    [...]
  },
  "batches": [ { "id": "0", "samples": 88 }, { "id": "1", "samples": 88 }, { "id": "2", "samples": 88 }],
  "defaults": {
    "constants": {
      "container_search_string": "YeastGatesStockPlate-02",
      "lab": "Strateos"
    },
    "parameters": {
      "inducer": "none",
      "inducer_conc": 0.0,
      "inducer_unit": "millimolar",
      [...]
    },
    "conditions": [
      {
        "factors": [
          { "factor": "media",
            "values": [ "high_osm_media"]},
          { "factor": "od",
            "values": [ "0.002", "0.001", "0.005" ] }, [...]
        ]
      }
    ]
  }
}

```

Figure 12: Condition Space and experiment format

If satisfiable, XPlan extracts the reagents, aliquots, and containers used for each measurement from the assignment found by the SMT solver. XPlan extends the measurement metadata in the SR with these decisions and produces an ED. From the ED, XPlan creates a set of *Lab Parameters*, and submits each set of Lab Parameters for execution at the lab. The lab responds with a run identifier for each set of Lab Parameters, and XPlan attaches it to the respective measurements in the ED. At this point, the ED has all the information needed to link requested measurements with the data to be produced by the lab. XPlan sends this ED to the Structured Request Generator for inclusion in the SR.

TASBE: TASBE flow cytometry tools [7] were used to compare and compose measurements (and derived models) across instruments and laboratories. These measurements must be calibrated to reproducible units (e.g., fluorescence into molecules of equivalent fluorescein (MEFL), forward scatter into equivalent μm (Eum) diameter). Methods for precise quantification and calibration are well established, but generally require careful application of complex multistage workflows. Prior tools do not provide these workflows, making it difficult for any but expert users to obtain precise, calibrated data.

TASBE Flow Analytics provides a solution for this problem: modular calibration and analysis workflows usable via Excel, Matlab, Octave, or Python, distributed under a free and open license.¹ This package was initially developed

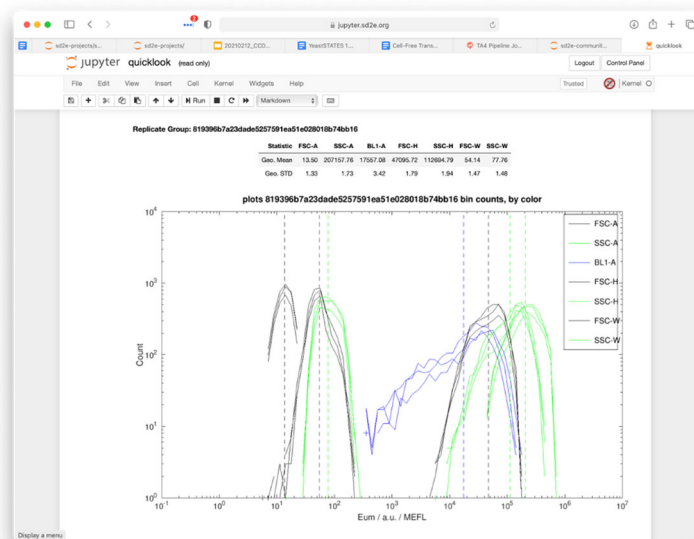


Figure 13: TASBE output converting flow cytometry measurements in arbitrary units to normalized units

internally at BBN as part of the Toolchain to Accelerate Synthetic Biology Engineering (TASBE) project, for purposes of characterizing transcriptional repressor devices with equivalent units on inputs and outputs, then refined across a number of projects, including the prediction of repressor circuits and replicon systems, the development of novel CRISPR-based repressors and promoter insulators, and interlaboratory studies on flow cytometry calibration. Now publicly released, it has also inspired two related packages, FlowCal and CytoFlow, each providing a subset of its functionality.

Flow cytometry is a powerful method for high-throughput precision measurement of cell fluorescence and size. Effective use of this tool for quantification of synthetic biology devices and circuits, however, generally requires careful application of complex multistage workflows for calibration, filtering, and analysis with appropriate statistics. The TASBE Flow Analytics package provides a free, open, and accessible implementation of such workflows in a form designed for high-throughput analysis of large synthetic biology data sets. Given a set of experimental samples and controls, this package can process them to output calibrated data, quantitative analyses and comparisons, automatically generated figures, and detailed debugging and diagnostic reports in both human-readable and machine-readable forms. TASBE Flow Analytics can be used through a simple user-friendly interactive Excel interface, as a library supporting Matlab, Octave, or Python interactive sessions, or as a component integrated into automated workflows.

Live-Dead Labeling: We developed a novel method for labeling flow cytometry data points (cells) as either live or dead. Subsequent analysis techniques benefit from omitting data points corresponding to dead cells, where accurate determinations of engineered cell behavior are not possible (e.g., by removing false negative data points). Prior approaches incorporate a red dye (sytox) into experimental protocols that is absorbed by dead cells with a ruptured cell membrane. This red dye can be read with a flow cytometer, and used to tag data points as dead (red) or live (not red). In an attempt to filter data from prior experiments to plan subsequent experiments, we sought to measure the red intensity from each data point. However, there was no clear threshold to separate varying red intensities into live and dead. For example, Figure 14 illustrates how a single threshold on the red intensity cannot separate the live (orange) and dead (blue) cells, and that another, even linear, function could separate the data better. To automate selection of a red intensity threshold, we trained a logistic regression model with data corresponding to live and dead control samples. We applied the model to predict live and dead data points in non-control samples that included engineered

¹ <https://tasbe.github.io/>, <https://github.com/TASBE/>

cells, and were able to get high (0.99) testing accuracy and predictions on control samples that correlated with intended engineered behavior (i.e., predicted live cells were more likely to function properly).

The logistic regression model, while linear, is more general than prior approaches that use a constant threshold, and

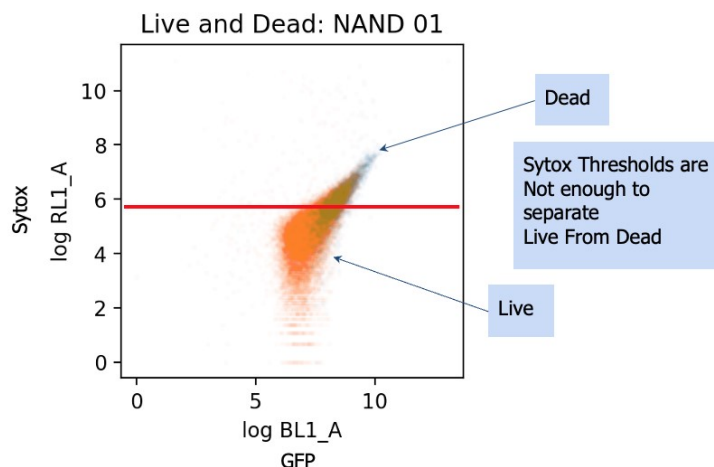


Figure 14: Red intensity (RL1 A) vs Green Intensity (BL1 A), with predicted live and dead data points

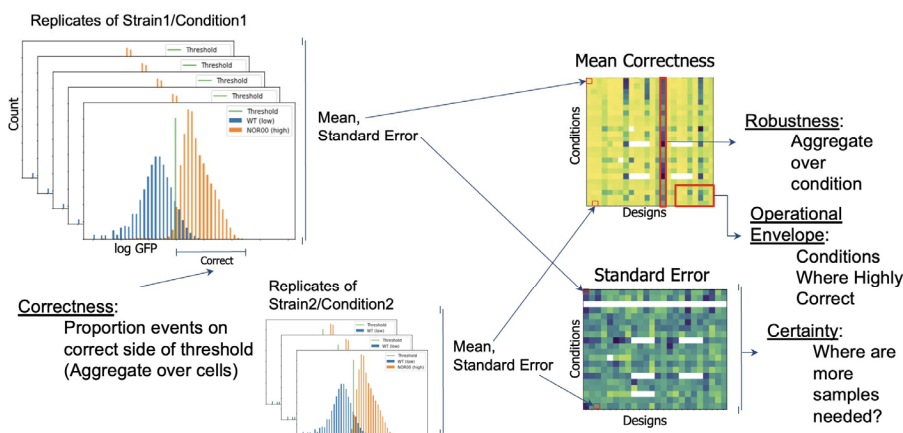


Figure 15: Predicted correctness (i.e. intended circuit output signal) for genetic circuits in yeast

we sought to understand why it performed better (0.99 accuracy) than constant threshold models (0.74 accuracy) for red intensity (subsequently developed using a gradient descent algorithm). Upon investigating the data point features used most strongly to make predictions, we found that the side scatter, and not the red intensity, was most used to predict the difference between live and dead cells. Consulting with other SD2 participants we found that side scatter is important (despite not typically used in practice) to determining dead cells. Side scatter measures the side diffraction of a laser, and increases with the number of granules present in a cell. Granules are often indicative of dead cells because they form from the breakdown of cellular components.

Signal Prediction: Building upon the method used to predict live and dead cells, we trained a model to predict the output signal of logic circuits engineered within host organisms. The output signal is typically measured as the intensity of GFP, and requires threshold (similar to that of the red intensity, above) to measure if a cell circuit has an on (high) or off (low) output. We developed an analysis tool that predicts for each data point generated via flow cytometry,

whether it was high or low. We used this feature to measure the accuracy of the engineered logic circuits (i.e., how many flow cytometry data points from a sample have the intended output signal).

Figure 15 illustrates how the signal prediction helps to develop a measure of circuit performance and robustness. The histograms with blue (low output cells) and orange (high output cells) bars indicate the frequency of GFP intensity measured by flow cytometry with respect to single sample. Training the signal prediction from data corresponding to low and high control samples, it predicts the low and high cells in non-control samples. The proportion of flow cytometry events whose predicted low or high output matches the intended output determines the correctness of the sample. Aggregating over samples, the mean correctness and standard error for each design under each condition provides a measure of quality and certainty, respectively. Aggregating over conditions provides a measure of robustness (i.e., robust designs perform consistently over conditions).

Growth Analysis: We developed a third analysis tool called Growth Analysis that calculates the growth rate for samples from both plate reader and flow cytometer time series data. The analysis helps to diagnose problems with samples demonstrating unexpected growth rates, such as particularly slow growth rates that indicate a technical failure, or poor reaction to growth media or other reagents used in the experiment.

pySBOL3: The Synthetic Biology Open Language version 3 (SBOL3) provides a data model for the representation of synthetic biology information across multiple scales and throughout the DBTL workflow. To support practical use of this data model, we have developed pySBOL3, a Python library that allows programmers to create and edit SBOL3 documents. The resulting implementation is a compact and maintainable core that provides both a familiar, pythonic interface for manipulating SBOL3 objects as well as mechanisms for building additional extensions and representations on this base.

SynBioHub Plugins: We developed multiple plugins for SynBioHub that added new functionality for data visualization, submission, and exchange. These include the sbol-dictionary-maintainer, a tool that synchronizes reagents, constructs, strains, etc. between Google Sheets and SynBioHub; sbol-dictionary-writer, an API to write reagents, constructs, strains, etc. to both Google Sheets and SynBioHub; synbiohub-adapter, a Python 3 package that provides an API to SynBioHub; SynBioHub metrics, a framework for gathering statistics or metrics from a SynBioHub instance, and SynBioHub Prospector, a tool that allows exploration of a SynBioHub database. Each of these tools supported RT use of SynBioHub at multiple phases of the test-learn cycle, and were used additionally by the Yeast States challenge problem to facilitate highly detailed descriptions of the design-build data elements (e.g., genetic circuit designs and parts).

OPIL Standard: We developed the Open Protocol Interface Language (OPIL) standard to enable labs to share information about their protocols with experiment planning tools. Communicating information about experimental design among a team of collaborators is challenging because different people tend to describe experiments in different ways and with different levels of detail. Sometimes, humans can interpret missing information by making assumptions and drawing inferences from information already provided. Doing so, however, is error-prone and typically requires a high level of interpersonal communication.

3.3.3 Additional Contributions

We developed several additional software tools to support the SD2 program, that did not directly contribute to the RT software infrastructure. Our most recent contribution was the Laboratory Open Protocol (LabOP) language to describe experimental protocols, which was based upon lessons-learned from the RT. The SYNBICT and Excel-to-SBOL tools aided with design-build data representation and analysis. The REDOER tool aided with the analysis of experiment data. The Fluorescence Prediction tool was applied to design experiments for a preliminary RT-like software infrastructure that focused on the interaction between XPlan and Strateos.

LabOP: The LabOP language for experimental protocols is an open standard for describing primitive operations (e.g., liquid transfer, incubation, and measurement) and the information and control flow between the primitives. In addition to replicability, interchange, and reuse, LabOP addresses an issue frequently encountered during our RT work – experimental campaigns have limited ability to modify protocols to explore unforeseen phenomena. For example, adding an additional plate reader measurement to one of our protocols required software updates within the Strateos protocol and throughout several of the RT modules. If the RT were based upon LabOP, the software tools could have implemented the standard and require no changes.

Even outside of SD2, laboratory protocols are critical to biological research and development, but can be difficult to communicate or reproduce due to the differences in context, skills, and resources between different projects, investigators, and organizations. Addressing this problem requires a data model for description of laboratory protocols

that is unambiguous enough for precise interpretation and automation, yet simultaneously abstract enough to support reuse and adaptation.

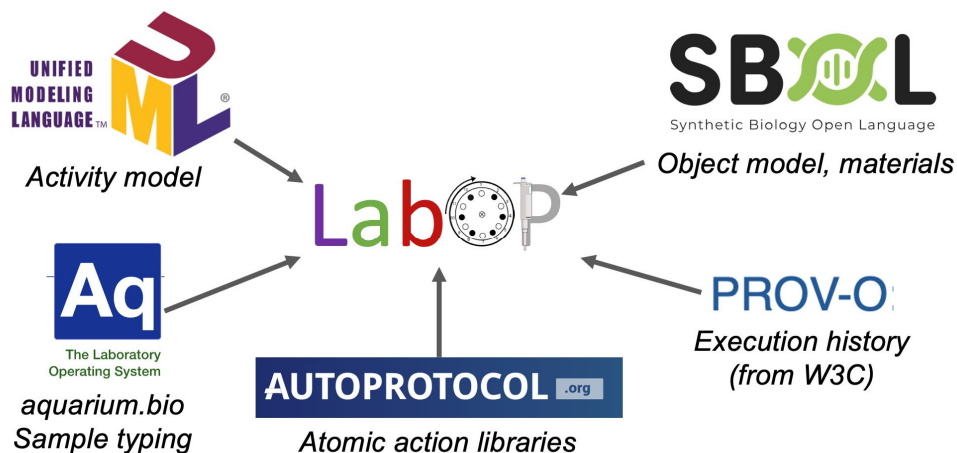


Figure 16: LabOP is a standard for describing laboratory protocols that is built upon multiple standards and approaches to laboratory science, including the UML Activity Model, SBOL, Prov-O, Autoprotocol, and Aquarium

Specifically, LabOP is an effective protocol representation that supports the following goals regarding biological experimentation: (1) Execution of a protocol by either humans or machines, (2) maintaining execution records and associated metadata markup, (3) mapping protocols from one laboratory environment to another, (4) recording modifications of protocols and the relationship between different versions, (5) maintaining and improving protocols with multiple contributors over time (6) verification and validation of protocol completeness and coherence (7) planning, scheduling, and allocation of laboratory resources.

While a number of machine-readable representations for protocols already exist, including protocols.io, Autoprotocol, Aquarium, and various hardware automation libraries, each of these serves only a small fraction of the needs above. To address this gap, we have developed LabOP, building on foundations from RDF, SBOL, UML, and Autoprotocol (Figure 16).

SYNBICT: Synthetic Biology Curation Tools (SYNBICT) is a Python tool suite for automation-assisted annotation, curation, and functional inference for genetic designs. It enables automated generation of linked data representations of designs from their sequences. Much progress has been made in developing tools to generate component-based design representations of biological systems from standard libraries of parts. Most biological designs, however, are still specified at the sequence level. Consequently, there exists a need for a tool that can be used to automatically infer component-based design representations from sequences, particularly in cases when those sequences have minimal levels of annotation. Such a tool would assist computational synthetic biologists in bridging the gap between the outputs of sequence editors and the inputs to more sophisticated design tools, and it would facilitate their development of automated workflows for design curation and quality control. We have validated SYNBICT by applying it to genetic designs in SD2 and the International Genetically Engineered Machines (iGEM) 2018 distribution. Most notably, SYNBICT is more automated and parallelizable than manual design editors, and it can be applied to interpret existing designs instead of only generating new ones.

Excel-to-SBOL: The Excel-to-SBOL tool automates genetic design ingest into SBOL from sequence data in Excel. Standards support synthetic biology research by enabling the exchange of component information. However, using formal representations, such as the Synthetic Biology Open Language (SBOL), typically requires either a thorough understanding of these standards or a suite of tools developed in concurrence with the ontologies. Since these tools may be a barrier for use by many practitioners, the Excel-SBOL Converter was developed to allow easier use of SBOL and integration into existing workflows. The converter consists of two Python libraries: one that converts Excel

templates to SBOL, and another that converts SBOL to an Excel workbook. Both libraries can be used either directly or via a SynBioHub plugin. We illustrate the operation of the Excel-SBOL Converter with two case studies: uploading experimental data with the study's metadata linked to the measurements and downloading the Cello part repository. **REDOER**: The REDOER tool enables quality-control on automated experimentation. Effective data-driven research in synthetic biology indeed, any scientific field first and foremost requires quality datasets that have been validated, curated, and correctly annotated with metadata for machine processing. However, human curation of datasets can quickly become unwieldy in high-throughput, automated laboratory environments. Moreover, human inspection of metadata is slow, expensive, and prone to overlook any flaw that is not highly salient in its presentation. Humans are also prone to mistakes, ranging from typos in data entry to large-scale conceptual errors. Automation-assisted curation can help address these issues of scale and integration.

For this reason, we devised the Reverse Engineering Design-of-Experiments for Review (REDOER) tool to assist with the curation and debugging of large, multi-factorial datasets. Given a collection of metadata from individual experimental samples, REDOER attempts to reconstruct the original design of the experiment. This more terse summary can then be inspected by humans or aligned against machine representations of design to identify errors in design, execution, or reporting of experiments.

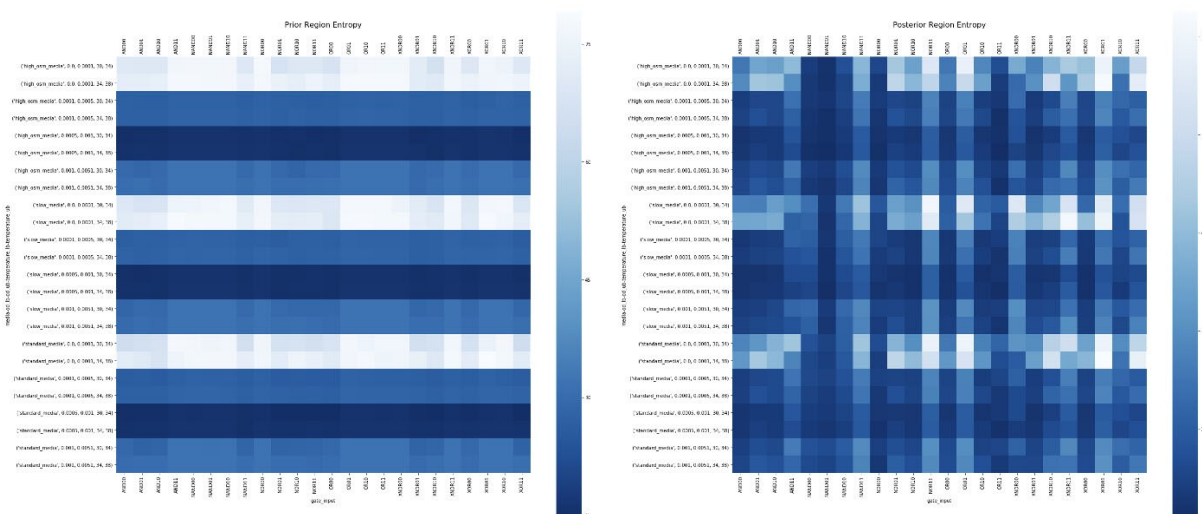


Figure 17: The fluorescence prediction model generalizes over designs (columns) and conditions (row) by using a prior distribution (left) and using data to compute a posterior distribution (right)

Fluorescence Prediction: We developed a probabilistic programming based method to predict the measured fluorescence intensity of samples from the experimental conditions (see Figure 17). The learned model was used by XPlan to optimize experiments with respect to the predicted information gain associated with gaining additional data under different experimental conditions. The models captured the behavior of two-input logic gates implemented in yeast cells. These circuits take promoters which they treat as input signals, either high or low, depending on concentration, compute the Boolean function through a sequence of genetic operations, and then signal their response through fluorescence. This output fluorescence signal is read by flow cytometry. Our work sought to replicate the original logic gate results, identify factors that account for (in)correct functioning of the gates, and characterize the operational envelope of the circuits: specifically, the growth conditions under which the circuit will (not) function correctly. “Correctness” of circuit function is defined in terms of the proportion of cells that exhibit high (low) output by fluorescence above (below) a defined threshold. Growth conditions explored in these experiments included choice of growth medium, incubation temperature, optical density (the density of the cell populations), etc. Our Bayesian probabilistic predictive model supports automatic or semi-automatic experiment planning: here what cell strains and growth conditions should be run to gain the most information. While this specific model aims at predicting the results

of flow cytometry measurements of these yeast gates, our overarching aim is to develop a general methodology for experiment planning.

4 RESULTS AND DISCUSSION

Synthetic biology experiments can be complex, particularly when high-throughput approaches are used. These high-throughput experiments can involve many strains developed from modifying host organisms in a number of ways. Furthermore, the experimental protocols can subject the strains to various conditions and take measurements at several times, possibly with different instruments. Analyzing the data produced by the experiments requires not only the context of each data point, but also the relationships between points. Capturing, representing, and exploring the context (i.e., metadata) behind each data point is a considerable challenge, especially as the stakeholders represent different organizations. Manual or *ad hoc* techniques for representing metadata are prone to data entry errors, human-human communication breakdowns and delays, and costly error correction. Our work addresses these problems by automating metadata construction, capture, and updating, in the context of supporting domain agnosticism.

4.1 Summary of Results

Our innovations impacted methods to conduct extended experimental campaigns. In Phase 1 of SD2, experiments proceeded with a slow tempo, involving delayed or imperfectly planned follow-on experiments. The inability to quickly analyze the experiment outcomes contributed to the delay between initial and follow-up experiments. Near the end of Phase 1 and start of Phase 2, the RT and related software tools led to a higher tempo where data-driven plans could guide each experimental campaign. Figure 18 illustrates how the RT and our contributions led to a 2.05x increase in the rate of experimental runs.

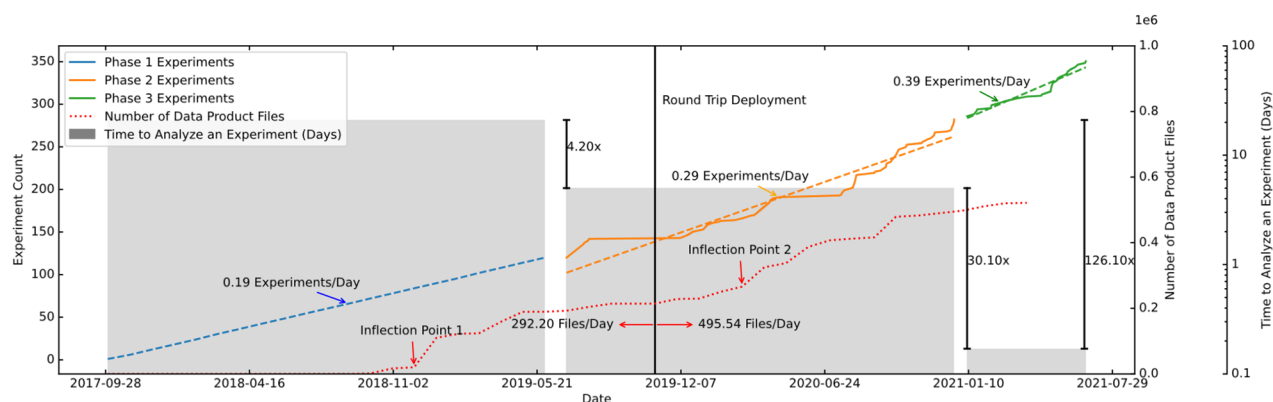


Figure 18: RT performance in SD2. (Left axis (blue, orange, and green lines): Cumulative number of Experiments over the course of the SD2 project, separated by program phase with linear fit (dashed) lines. The RT was introduced near the start of Phase 2, as indicated by the vertical black line. Phase 1 data did not closely track experiment dates and we illustrate the linear fit only. First right axis (red dotted line): Cumulative data product volume over SD2. Second right axis (gray bars and black lines): Mean time to analyze data after experiment data upload by program phase of SD2.)

Our tools helped improve the sharing and utility of information across the SD2 community, supporting the progress of ongoing experimental workflows, most notably in the Yeast States, Novel Chassis, and Cell-Free Riboswitches challenge problems. Program knowledge sharing expanded greatly. SD2's SynBioHub currently has data on 2,874 experiments, 3,549,237 components, and 15,884 modules, composed of 22,872,306 triples that are organized into 524,377 collections. Moreover, measurement of key knowledge collections shows that progress on tools correlates with increases in knowledge sharing and productivity. Figure 19 (left) shows the correlation between knowledge sharing and data production, with both moving much more quickly in the second half of SD2 after these tools began

to be released, and Figure 19 (right) shows that knowledge expansions in SynBioHub are correlated with the dates at which each of these key tools were released. Even now, however, available information about designs and experiments is only partially exploited for analysis.

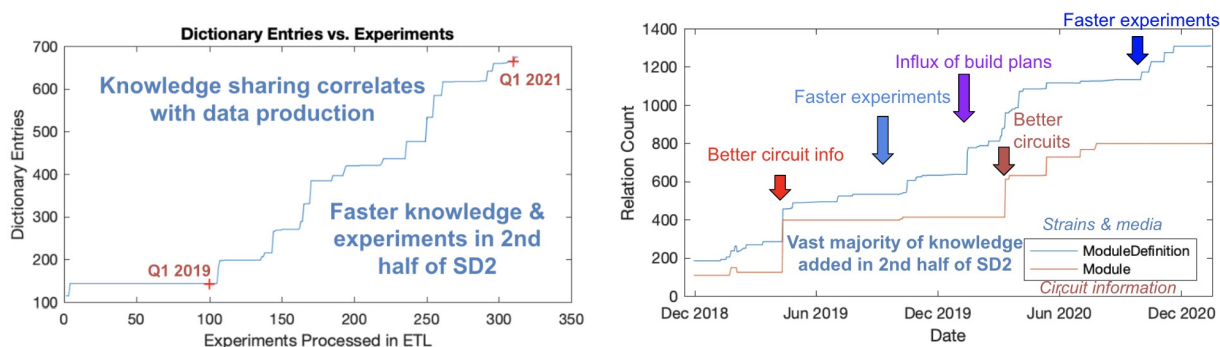


Figure 19: Increased knowledge sharing is correlated with overall rates of experimentation in SD2 (left), and specific tools are correlated with expansions in key forms of shared knowledge (right)

Table 6: Summary of the On-The-Loop Experimental Campaign (Duration and Turnaround are measured in days. The round 2.0 turnaround is measured with respect to round 1.0.)

Round	Runs	Measurements	Submission	Completion	Duration (Overall)	Duration (Active)	Turnaround (Overall)	Turnaround (Active)
1.0	4	2916	7-10-20	7-18-20	8	3	-	-
1.1	2	1458	9-1-20	10-1-20	30	2	44	2
2.0	1	486	9-29-20	10-1-20	3	2	72	0.5
3.0	4	2916	10-27-20	11-7-20	11	4	28	1
4.0	3	2187	1-28-21	2-11-21	14	3	83	1
Mean	2.8	1992.6	-	-	13.2	2.8	56.8	1.1

4.2 Detailed Round Trip Results

We evaluate the impact of the RT in terms of a representative use case and a broad characterization of the benefits gained over the course of the SD2 project.

In the following, we use a running example of an experiment called On-the-Loop Round 4.0 (abbreviated as Round 4.0). The On-the-Loop of a multiple-round, screening-focused experimental campaign conducted with the RT. The experiments sought to optimize the experimental conditions to identify controllable sources of variation and to assess the relative performance of various synthetic biology constructions. We used twenty-four strains implementing six logic circuits in *Saccharomyces cerevisiae* (yeast) using *dCas9-MxiI*-based NOR-gates [10] made from constitutively expressed short guide RNA segments. Over five experimental rounds, we used a configurable time series protocol to collect flow cytometry and plate reader measurements, and restricted our variable conditions to four media types (standard, rich, slow growth, and ethanol), two temperature conditions, and the timing of sampling.

On-the-Loop Discovery and Identification of Sources of Variation: The RT supports a rapid experimental investigation paradigm where the experiment designer can guide and monitor a largely automated experimental campaign while “on-the-loop”, rather than being deeply involved in each step. We conducted several rounds of experiments with the yeast circuits to identify the sources of variation and candidate circuit designs requiring redesign to improve their quality and robustness. We describe the impact of the RT upon this process in terms of the duration and types of analyses supported.

Table 6 lists the RT end-to-end duration and turnaround time for each round of the experimental campaign. Each round involved a number of different runs (indicated in the table), where each used the time series protocol with a different media and distinct starting 96-well microplate. The table lists the submission and completion date of each

round, measured from the start of ER authoring to the final analysis completion. The table also lists the duration in days between submission and completion that includes overall (wall-clock) time and active (user-engagement) time. Finally, the turnaround time denotes the overall and active time for the experiment designer to plan the round, based upon data from the previous round. While the overall duration and turnaround was impacted by unrelated laboratory and experiment designer workload (i.e., other work tasks and vacation/holidays) the active time was relatively consistent.

The primary impact upon a round's active duration was the extent to which runs were run in parallel at the laboratory. The results show that the mean active duration of an experiment round was 2.8 days, and the mean active turnaround time spent planning a round in response to the previous round was 1.1 days. This was for a mean number of 2.8 experimental runs per round, and resulted in a mean of 1992.6 measurements per round.

Cumulative RT Experimental Impact: In addition to the experimental campaign described above, the RT has been used for many other experiments within the SD2 program. Figure 18 illustrates the cumulative number of experiments run over the course of the SD2 project separated by phase, cumulative number of data product files produced, and the time required to analyze new experiment data to produce data sets. Each experiment data point refers to the date where the data for a respective experiment was delivered to the SD2 data repository. Prior to the RT deployment and use (i.e., from late 2017 through to mid 2019), data was uploaded in a preliminary format. Figure 18 illustrates the Phase 1 experiments as a dashed trend line only to indicate that the experiment upload to the SD2 platform was sporadic, with several large batches occurring months after execution. Overall, in Phase 1 there were 0.19 experiments per day, using one experimental protocol.

The RT was deployed in November of 2019. We note that the lack of additional experiments in mid 2020 was due to reduced capacity in candidate experiment designers creating samples to screen with the RT due to COVID-19 mandates. The ingest date depicted by Figure 18 over this period is much more closely aligned with the experiment date. The experiment frequency afforded by RT was 0.29 experiments per day over Phase 2, and 0.39 experiments per day over Phase 3 – representing a 2.05x increase in the rate of experiments. The experiments in Phase 2 comprised four experimental protocols and in Phase 3, five protocols – contributing to the increased complexity of the set of possible experiments.

The first right axis in Figure 18 illustrates the cumulative number of data product files uploaded to or created on the SD2E platform over the course of the SD2 project. The plot of the cumulative number of data products exhibits two inflection points, located in December 2018 and March 2020. Prior to the first inflection point, as noted above, we developed a preliminary data format and built software to support a more rapid, bulk ingest of the data. During this time before December 2018, the rate of data product generation was 44.11 files per day. Following bulk ingest and prior to RT deployment, the rate of data product generation increased to 931.30 files per day – mainly due to the bulk processing experiments run in the prior months. The combined rate of data product generation prior to the RT was 292.20 files per day. Following RT introduction, the rate of data product generation was 495.54 files per day, a 1.70x increase in the rate of data product generation. The second inflection point indicates the time at which RT data set production was made operational for a large set of experiments.

Finally, the second right axis in Figure 18 illustrates the estimated time to produce an analysis of an experiment after data upload. The 4.20x speed-up from Phase 1 to Phase 2 corresponds to introduction of the RT to automate metadata creation with manually triggered, yet standardized analysis. The final 30.10x speedup from Phase 2 to Phase 3 was achieved by automating all metadata creation, data standardization, and analysis. Overall, from Phase 1 to Phase 3 the SD2 RT elements contributed to a 126.1x speed-up in producing the analyses of experiments.

These results illustrate how data representation and automation, in particular domain agnostic experimental planning, can improve the rate, volume, and depth of data over loosely integrated, largely manual methods.

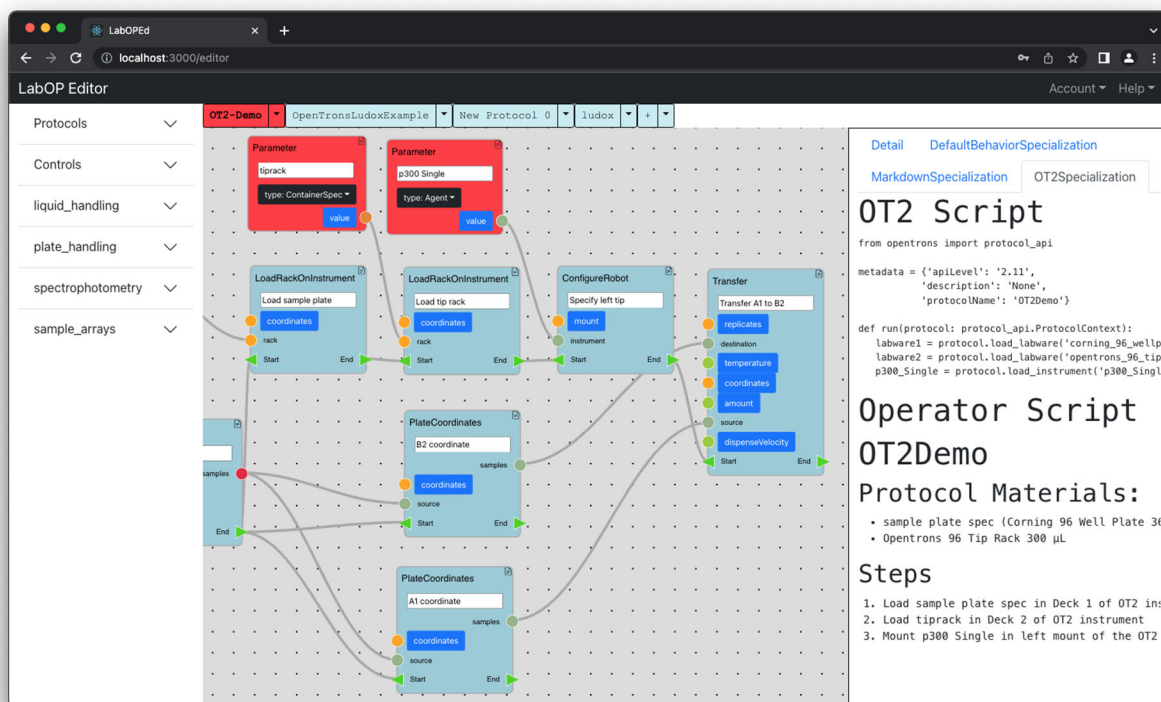


Figure 20: LabOP Editor, authoring the Opentrons demo protocol

4.3 LabOP Results

The LabOP language was a major focus of our work over the final year of the project. We developed several demonstrations of LabOP and undertook transition activities to widen adoption. In the following, we highlight the LabOP technical results and transition-related results.

LabOP Technical Results: We have developed demonstrations of LabOP that illustrate its capabilities, as follows:

- **iGEM Interlab:** We collaborated with the iGEM Engineering Committee to launch the iGEM 2022 interlaboratory experiments, using LabOP protocols (published on the iGEM website²). The LabOP protocols were run by ten laboratories as part of an interlab test run and used by more than 100 iGEM teams for the full-scale interlab experiments.

The protocols include:

- three color calibration (https://old.igem.org/wiki/images/f/f8/InterLab_2022_Exp1.pdf),
- culture and measure (https://old.igem.org/wiki/images/f/ff/InterLab_2022_Exp2.pdf), and – time series (https://old.igem.org/wiki/images/4/4c/InterLab_2022_Exp3_Standard.pdf).

- **Opentrons Integration:** We developed an Opentrons specialization for LabOP, working with a LUDOX calibration protocol and demonstrated its execution on the Opentrons simulator and an OT2 liquid handling robot. In the Opentrons execution environment for LabOP, we supported parallel generation of a human-readable protocol (to manually set up the robot) and an OT2 API script, and implemented a Ludox calibration protocol for Opentrons (validated on the OT2 execution in BBN's lab). We developed and demonstrated a

² <https://technology.igem.org/interlabs/2022>

LabOP representation of an OT2 PCR protocol in simulation, and a forthcoming physical demonstration at BBN. Demonstration versions of the LabOP protocols include:

- Opentrons protocol demo (protocol)
- Opentrons PCR protocol (protocol)
- **Multiformat Specialization:** LabOP supports numerous specializations of a interchangeable protocol into target execution formats, including Autoprotocol, Opentrons, and human-readable Markdown, in addition to the native RDF format.
- **Chemical Process Control:** In addition to laboratory protocols, LabOP supports protocols that represent chemical process control. In coordination with the SiLA standard governing body, we developed a representative chemical process control protocol. Process control, unlike traditional laboratory protocols uses multiple execution control flow decision points, both incorporating human operator feedback and device status.
- **LabOP Editor:** We developed the LabOP Editor as a way to lower the barrier to entry for non-programmers to author protocols. The editor supports authoring protocols, converting to the specialization formats (above), and exporting as RDF. The editor is based on modern web technology and can be run as a public-facing web application or a locally installed application. Figure 20 shows a protocol under authoring developed using the LabOP Editor.

LabOP Transition: We formed a governing body, called the Bioprotocols Working Group, which oversees the LabOP standard as an open source community. The Bioprotocols Working Group (labop.io and bioprotocols.org) is organized under OpenCollective, has elected leaders (chair and finance committee), and ratified articles of governance. The working group holds regular meetings, hosts workshops at affiliated community events, and pursues public and private investment in sustainability. The working group currently includes more than seventy members across more than twenty organizations.

5 CONCLUSIONS

We have described how our work on knowledge representation and automated planning tools as part of the SD2 program. It has led to increasing the throughput of experimentation in data intensive laboratory science. This includes research on methods for designing and applying DBTL loops in a number of application domains. In this setting, we addressed the challenge of supporting a group of stakeholders to collaboratively and effectively develop a shared experimental intent and to harness automation for experimentation.

The primary focus of our work was the SD2 RT architecture. The RT comprises several software components that are aimed to help create and curate metadata in support of generating and automatically analyzing large data sets. The contributions of the RT are flexible knowledge representations and automated processing steps that support experiment designers in specifying their experiments and maintaining the integrity of the data produced. We have demonstrated an increased rate of experiments and complexity, and decreased data analysis times that result in significantly more data being generated per time period. We have also applied the RT to multiple different experimental protocols, organisms, and DBTL loops.

REFERENCES

- [1] Broad Institute. <https://www.broadinstitute.org>. Accessed: 2020-6-17.
- [2] DOE Joint Genome Institute. <https://jgi.doe.gov>. Accessed: 2020-6-17.
- [3] Edinburgh genome foundry. www.ed.ac.uk/biology/research/facilities/edinburgh-genome-foundry. Accessed: 2020-6-17.
- [4] Emerald Cloud Laboratory. www.emeraldcloudlab.com. Accessed: 2020-6-17.
- [5] Strateos, Inc. www.strateos.com. Accessed: 2020-6-17.
- [6] BARRETT, C., AND TINELLI, C. Satisfiability modulo theories. In *Handbook of Model Checking*. Springer, 2018, pp. 305–343.
- [7] BEAL, J., OVERNEY, C., ADLER, A., YAMAN, F., TIBERIO, L., AND SAMINENI, M. TASBE flow analytics: A package for calibrated flow cytometry analysis. *ACS Synthetic Biology* 8, 7 (2019), 1524–1529. PMID: 31053031.
- [8] BRYCE, D., GOLDMAN, R. P., DEHAVEN, M., BEAL, J., BARTLEY, B., NGUYEN, T. T., WALCZAK, N., WESTON, M., ZHENG, G., NOWAK, J., LEE, P., STUBBS, J., GAFFNEY, N., VAUGHN, M. W., MYERS, C. J., MOSELEY, R. C., HAASE, S., DECKARD, A., CUMMINS, B., AND LEIBY, N. Round trip: An automated pipeline for experimental design, execution, and analysis. *ACS Synthetic Biology* 11, 2 (2022), 608–622. PMID: 35099189.
- [9] DE MOURA, L., AND BJØRNER, N. Z3: An efficient SMT solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (Berlin, Heidelberg, 2008), TACAS’08/ETAPS’08, Springer-Verlag, p. 337–340.
- [10] GANDER, M. W., VRANA, J. D., VOJE, W. E., CAROTHERS, J. M., AND KLAVINS, E. Digital logic circuits in yeast with CRISPR-dCas9 NOR gates. *Nature Communications* 8 (2017).
- [11] KING, R. D., WHELAN, K. E., JONES, F. M., REISER, P. G., BRYANT, C. H., MUGGLETON, S. H., KELL, D. B., AND OLIVER, S. G. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 427, 6971 (2004), 247–252.
- [12] KLAVINS, E., KELLER, B., MILLER, A., NEWMAN, G., YANG, Y., LE, P., JOONSEOKII, VRANA, J., BOLTEN, N., TAKAHASHI, C., STRICKLAND, D., AYESHASAL, NILES, S., CASH, A., GNOMICOS, ZAMAN, L., ANDERSON, M., MICHELLEPARKS, AND TAMIMEUR. *aquariumbio/aquarium: Aquarium version 2.8.1*, Sept. 2020.
- [13] KUTER, U., ET AL. XPLAN: Experiment planning for synthetic biology. In *ICAPS Workshop on Hierarchical Planning* (2018).
- [14] MCLAUGHLIN, J. A., ET AL. SynBioHub: A standards-enabled design repository for synthetic biology. *ACS Syn. Bio.* 7, 2 (2018), 682–688.
- [15] PENDLETON, I. M., CATTABRIGA, G., LI, Z., NAJEEB, M. A., FRIEDLER, S. A., NORQUIST, A. J., CHAN, E. M., AND SCHRIER, J. Experiment specification, capture and laboratory automation technology (ESCALATE): a software pipeline for automated chemical experimentation and data management. *MRS Communications* 9, 3 (2019), 846–859.
- [16] ROEHNER, N., ET AL. Sharing structure and function in biological design with SBOL 2.0. *ACS Syn. Bio.* 5, 6 (2016), 498–506.

6 LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AFRL	Air Force Research Labotory
API	Application Programming Interface
DARPA	Defense Advanced Research Projects Agency
DBTL	Design-Build-Test-Learn
DNA	deoxyribonucleic acid
ED	Experimental Design
ER	Experiment Request
ESCALATE	Experiment Specification, Capture And Laboratory Automation TEchnology
ETL	Extract-Transform-Load
GFP	Green Fluorescent Protein
iGEM	Internationally Genetically Engineered Machine
JGI	Joint Genome Institute
JSON	JavaScript Object Notation
KM	Knowledge Management
LabOP	Laboratory Open Protocol
LIMS	Laboratory Information Management System
MEFL	Mean Equivalent Fluorescein
MIT	Massachusetts Institute of Technology
OPIL	Open Protocol Interface Language
PCR	Polymerase Chain Reaction
PNNL	Pacific Northwest National Laboratory
RDF	Resource Description Format
REDOER	Reverse Engineering Design-of-Experiments for Review
RT	Round Trip
SBOL	Synthetic Biology Open Language
SD2	Synergistic Discovery and Design
SIFT	Smart Information Flow Technologies, LLC.

SMT	Satisfiability Modulo Theories
SR	Structured Request
SRG	Structured Request Generator
STEM	Science, Technology, Engineering, and Mathematics
SYNBICT	Synthetic Biology Curation Tools
TA	Technical Area
TASBE	Toolchain to Accelerate Synthetic Biology Engineering
UML	Unified Modeling Language
URI	Uniform Resource Identifier
UW	University of Washington
W3C	World Wide Web Consortium
XPlan	Experimental Planner