Round Trip: An Automated Pipeline for Experimental Design, Execution, and Analysis

Daniel Bryce,* Robert P. Goldman, Matthew DeHaven, Jacob Beal, Bryan Bartley, Tramy T. Nguyen, Nicholas Walczak, Mark Weston, George Zheng, Josh Nowak, Peter Lee, Joe Stubbs, Niall Gaffney, Matthew W. Vaughn, Chris John Myers, Robert C. Moseley, Steven Haase, Anastasia Deckard, Bree Cummins, and Nick Leiby



ABSTRACT: Synthetic biology is a complex discipline that involves creating detailed, purpose-built designs from genetic parts. This process is often phrased as a Design-Build-Test-Learn loop, where iterative design improvements can be made, implemented, measured, and analyzed. Automation can potentially improve both the end-to-end duration of the process and the utility of data produced by the process. One of the most important considerations for the development of effective automation and quality data is a rigorous description of implicit knowledge encoded as a formal knowledge representation. The development



of knowledge representation for the process 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 experimental types. We address these challenges with the DARPA SD2 Round Trip software architecture. The Round Trip is an open architecture that 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 Round Trip is to assist with and otherwise automate metadata creation, curation, standardization, and linkage with experimental data. The Round Trip's focus on metadata supports fast, automated, and replicable analysis of experiments as well as experimental situational awareness and experimental interpretability. We highlight the major software components and data representations that enable the Round Trip to speed up the design and analysis of experiments by 2 orders of magnitude over prior *ad hoc* methods. These contributions support a number of experimental protocols and experimental 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 experimental analysis time and increasing data product volume in the SD2 program.

KEYWORDS: Design-Build-Test-Learn, automation, knowledge curation, high-throughput screening, data analysis, metadata

1. INTRODUCTION

The lack of automated tools for experimental design, execution, and analysis is a key barrier to quickly evaluating biological designs. It is challenging to unambiguously describe experimental plans in a format that laboratories can act upon. Furthermore, it can be difficult to link experimental data to metadata and experimental plans. The connection of experimental intent to raw data to analyses is an expensive (oftentimes manual) undertaking.¹ Metadata describing the contents, conditions, and context of experimental samples is key to gaining insights from experimental data, but acquiring, tracking, and maintaining metadata is tedious, expensive, and error-prone.

The eradication of manual steps and human-mediated gaps between experimental software components is formidable but has the potential to improve both metadata reliability and the speed of analysis through standardization and automation. We have addressed this challenge by developing the Round Trip (RT) architecture. The RT adopts "make metadata easy" design principles, providing tools and representations to automate tedious metadata design and encoding and react and repair as needed.

We developed the RT as part of the Defense Advanced Research Projects Agency (DARPA) Synergistic Discovery and Design (SD2) project. The primary aim of SD2 is to help

Received: July 2, 2021 Published: January 31, 2022







Figure 1. Typical practice in biological collaboration (top) involves many iterations of communication between the collaborators dispersed across multiple channels of communications, often leading to confusion, mistakes, and poor tracking of information about experiments (taking many weeks). The RT (bottom) automates and provides structure to several human–human interactions required to plan and analyze experiments (taking hours). The numbered stages constitute the major steps facilitated by the RT.

scientists and data scientists quickly develop a common understanding of experiments. The SD2 project is 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 include machine learning and data science for discovery, experimental planning and design, experimental execution, technology platform, and challenge problem formulation (in the service of analyzing *ad hoc* socio-technical organizations). The authors represent a cross-section of the organizations addressing the technical areas as well as the challenge problems. Within each challenge problem it addresses, the RT helps connect scientists and data scientists by advancing the metadata and automation needed to easily express and quickly analyze experiments in so-called "AI-ready" data sets. In doing so, the RT also enhances replicability through automated experimental planning, robotic execution, versioned data sets, and containerized analysis tools.

In the following, we describe the scope of experiments addressed by the RT, related work, and an overview of the RT elements.

1.1. Scope. We focus on high-throughput screening experiments for synthetic biology that involve microbes, such as bacteria or yeast, or cell-free systems. Each experiment requires several measurements of the engineered strains or biological materials, calibration samples, and controls under a variety of conditions. The conditions include temperature, media, and reagents (dyes, antibiotics, or inducers). The measurements include plate readers (fluorescence and optical density) and flow cytometry with an option for time series data collection. The number of strains, conditions, and replicates

determines the number of samples. The number of samples and measurements determines the number of data points. Each experiment uses a protocol from a set of protocols described in Section 4.2 to generate the data points. We address experiments that include more samples than can be cultured with a single microplate and, thus, require multiple runs of an experimental protocol to generate the requisite data. These data are aggregated together with metadata downstream of the experimental process.

While we describe our work in the context of the scope above, there are a number of ways that it can be adapted. Highthroughput screening usually involves microplate-based protocols executed by robots, but our techniques have also been applied to bench-scale protocols that replace the machinemachine lab interface by a machine-human interface. The supported reagents, containers, and measurements (and resulting data types) have been extended to additional types over the course of the SD2 project but do not require architectural changes to the RT software. Likewise, the addition of protocols may require extensions to experimental descriptions but do not require architecture changes. We make extensive use of the existing data formats, including SBOL² to describe strains and reagents, and formats such as FCS or CSV for measurement data. We developed additional metadata formats as part of the RT that are based on open formats and that are readily extensible.

1.2. State of Practice. Figure 1 (top) illustrates the state of practice encountered by the SD2 project at its inception. This state of practice is illustrative of how scientists collaborate outside of large industrial foundries (*e.g.*, small to medium sized academic laboratories) or across multiple institutions. In this setting, several roles (which may be fulfilled by the same person) must translate information between their respective domains of expertise. In the SD2 program, each challenge problem initially included 1–3 experimental designers, 2–3

laboratory technicians, and 4-5 data scientists (with 1-2 persons acting as both experimental designers and data scientists). The process involves the following steps.

- The experimental designer must communicate their experimental intent to a laboratory technician that conducts the experiment (either at the bench or with robotic automation). A new experiment typically takes around 4 to 6 weeks to configure and send samples to the laboratory, though the time may greatly depend on the required materials and methods.
- The laboratory technician will gather raw data and provide it to a data scientist. The execution of an experiment will take as little as 1 day and up to many weeks, depending on the details of the protocol and the number of experimental runs required. Moderately sized experiments typically require approximately 1 week, though they may be much longer if the protocol requires more than a few days to run.
- The data scientist develops a data set that describes the raw data through a number of metadata attributes for each measurement. He/she performs analysis and develops a presentation of the data set. The process often takes on the order of 4 to 8 weeks for analysis and up to another week to develop the presentation materials to ensure that the analysis accurately captures the experiment.

Ambiguity and/or lack of structure to the information exchanged by the roles complicates the development of the final data set and analysis. Numerous interactions between the roles (at the speed of human communication) will eventually result in a suitable data set. 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 the experiments. Moreover, attempts to conduct similar experiments may need to repeat this process and can result in data sets that are not directly comparable to previous data sets.

In the following, we discuss how competing approaches have developed experimental automation and data management approaches for producing data sets from experimental descriptions.

1.3. Comparison to State of the Art. Synthetic biologists are increasingly attempting to improve the description of experiments 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 specifically addressing the challenges of collaborative science. For example, much work on the test (*i.e.*, experimentation) phase includes automation in the form of Lab Inventory Management System (LIMS) systems and laboratory robots (e.g., Strateos,³ Aquarium,⁴ Emerald Cloud,⁵ and the Edinburgh Foundry⁶). While being important for the organization of repeatable experiments within a laboratory, these improvements to the test phase do not specifically address collaboration between stakeholders. The build phase likewise includes automation at gene foundries such IGI⁷ and MIT-Broad⁸ but similarly does not directly address collaboration. Finally, some highly automated DBTL systems, such as ESCALATE⁹ and the robot scientist,¹⁰ address specific workflows (e.g., mapping a space of chemical reaction parameters) and are able to achieve a 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 of 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.

In addition to these focused approaches, several companies offer general data management systems for organizing experimental data. The common theme among these offerings is a consultation-driven adaptation of a propriety solution. For example, these systems include Riffyn Nexus,¹¹ Synthace,¹² Benchling,¹³ and RadixBio.¹⁴ Riffyn Nexus is a process data system that records and provides user and program access to process data. Riffyn Nexus provides end-user development of scientific workflow, analytical and manufacturing processes, sample and material provenance, standardized data sets for analysis, and data access control. The Synthace platform is a cloud service that helps scientists describe and execute experimental protocols as well as analyze and share their data. Benchling is a cloud software service centered around electronic lab notebooks that include not only experimental protocols but also representations of biological designs, inventory, scientific workflows, and analytics. Benchling also provides tools for laboratory automation and data access control. Radix Bio is a software platform for lab integration that enables end-user development of automated protocols, dynamic scheduling of protocols, equipment integration, and experimental and data provenance.

Unlike the RT, these commercial solutions are closedsource. The extension or migration of data and processes between these services can be challenging or costly due to their proprietary nature. The RT software components are free and open source. The RT can, in principle, be extended to interface with these commercial services and support the interchange of data formats. For example, protocols described by such services can be configured in the same manner that the RT configures protocols. Forthcoming protocol interchange standards,¹⁵ developed as part of the SD2 project, are a possible avenue for migration between the RT and other providers.

1.4. SD2 Round Trip. The RT is an open system that connects experimental data and subsequent analyses with deeply represented experimental constructs by resolving user-friendly construct names. It focuses on helping experimental designers specify the requested high-level data products and then automating the selection of low-level details when possible. The RT provides situational awareness of the experimental process and monitors for errors, such as mismatches between expected and actual data. It creates AI-ready data sets that it automatically analyzes and presents. Finally, the resulting data sets support reproducible analysis through data-level aggregation (beyond just conclusion-level aggregation).

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. In order to cover the condition space,



Figure 2. Round Trip architecture. The primary components include the Intent Parser,¹⁶ Data Dictionary,¹⁷ SynBioHub,¹⁸ the Structured Request Generator, XPlan,¹⁹ Strateos,³ ETL and Data Repository (www.sd2e.org), Data Converge, Precomputed Data Table, Escalation,²⁰ and a Dashboard.

many Test-Learn "inner" loops within a single DBTL "outer" loop are required. For these reasons, we have developed the RT to be highly integrated for Test-Learn and have made it extensible for easy incorporation within application-specific DBTL loops. Furthermore, while our discussion is limited to the types of experiments noted above, the RT is readily adaptable to related experiments that include different types of equipment, protocols, and organisms.

Figure 1 (bottom) illustrates the state of practice provided by the RT and exercised by the SD2 project. This workflow is similar in spirit to the proprietary solutions mentioned above in that it seeks to simplify and automate as much of the metadata collection and execution as possible. The stages numbered from (1) the experimental inception to (11) the presentation of the results remove the bottlenecks of the prior state of practice. The RT helps to situate human interaction where intuition and expertise are needed and automation where tedium and the propensity for human error can arise. Figure 2 illustrates the RT software architecture, also numbered by stage (with relevant components illustrated in a clockwise manner as the stages proceed). The preexperimental stages are implemented by components as they proceed left-to-right on top, steps (1) to (5), to create an automation-ready experiment, executed in step (6). The RT attaches metadata to raw data as it returns through postexperimental data processing components, steps (7) to (11). The key steps are cross-referenced in Figures 1 and 2 and are detailed in Section 2 section.

The RT has been developed over the course of three years and has been applied to nearly one hundred experiments. Each experiment produces hundreds to thousands of measurements, often comprising many gigabytes of raw data. The resulting inception-to-analysis latency of the RT is a few hours, not including laboratory time. Figure 1 accounts for the approximate time required by each RT processing stage, which is approximately 8–10 h, not including a (typically) 1 week execution time at the laboratory. In contrast, prior to developing the RT, the inception-to-analysis latency within SD2 typically required several weeks or months. We attribute this speed-up to automating some of the previously humanintensive tasks. At the end of the SD2 program, we reduced the number of experimental designers from one to three down to one, the lab technicians from two to three down to one, and the number of data scientists from four to five down to two to three (with the experimental designer also acting as the data scientist). This reduction from approximately ten persons to three represents a significant labor savings that is also multiplied across the reduced experimental cycle duration. We attribute the staffing and workload differences to both reduced complexity and automation.

Automation contributed to removing human-human communication bottlenecks that arise from coordinating and correcting ad hoc analogs of the steps automated by RT. For example, a frequently observed issue in SD2 arose when the experimental designer, laboratory technician, and data scientist used different identifiers for strains. Without the RT, the reconciliation of the discrepancies both before and after the experiments were run involved: manually identifying the issue, tracing through the experimental protocol, initiating several correspondences between the stakeholders, and then instituting a brittle identifier mapping in the analysis software. Mistakes or other changes to the mappings necessitated another manual repetition of the process. An important lesson taught by this process is that more opportunities for manual data entry equates to more data entry errors. With the RT, data entry is the responsibility of one individual (the experimental designer) and automation propagates updates throughout the metadata. While human error can still play a role, automation avoids inconsistent corrections, high communication latency, and misunderstandings.

New RT users may access the RT by establishing an account with the SD2 Enterprise organization, hosted by the Texas Advanced Computing Center (TACC) (https://sd2e.org). The components are also currently deployed at TACC as containers and can be adapted to other container-based deployments. The execution of experiments at Strateos also requires one to establish an account with Strateos (https:// www.strateos.com). The SD2 project has open-sourced several software components implementing the RT and will complete the process for all components in late 2021 (https://github. com/SD2E). In the remainder of the manuscript, we first describe the RT data model and software architecture. We then demonstrate its efficacy through the presentation of the empirical results of applying the RT in the SD2 program over two years, demonstrating 2 orders of magnitude improvement in time from experimental design to completion of primary data analysis. Finally, we conclude with a discussion of the results and future directions.

2. RESULTS

Synthetic biology experiments can be complex, particularly when high-throughput approaches are used. These highthroughput experiments can involve many strains developed from the modification of 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. The analysis of the data produced by the experiments requires not only the context of each data point but also the relationships between points. The capture, representation, and exploration of the context (i.e., metadata) behind each data point are considerable challenges, 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. The RT addresses these problems by automating metadata construction, capture, and updates.

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 was 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 24 strains implementing six logic circuits in Saccharomyces cerevisiae (yeast) using dCas9-Mxi1-based NOR gates²¹ 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. We use this example to describe the RT data model, software architecture, and results from using the RT in the SD2 Project.

2.1. Round Trip Data Model. We describe the RT data model in terms of how it transforms its input, called an Experiment Request (ER), to the output, which is a data set containing both raw data and analyses annotated with strain and experimental condition metadata. We use the Round 4.0 experiment to show how the RT makes this transformation. Figure 3 illustrates the RT data flow from the ER to the 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 experimental designer (Section S2 includes a full ER). 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 Pis a protocol. Each control $c \in C$ identifies a measurement $m(c) \in M$ that will be used in the analysis, such as the



Figure 3. RT transforms an ER into a data set by creating and merging multiple experimental runs and producing multiple analyses. The numbers 1 to 11 reference the RT stages that process the indicated data model elements.

measurement characterizing the minimum or maximum green fluorescent protein (GFP). 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 Control Table^a

| Table 1: Control | | | | | | | | |
|------------------|-----------|---------|-------------|-------------------|--|--|--|--|
| Control type | Strains | Channel | Contents | Time point (h) | | | | |
| EMPTY_VECTOR | W303 | | SC media | 8 | | | | |
| HIGH_FITC | UWBF_6390 | BL1-A | SC media | 18 | | | | |

"Each control table in an ER identifies measurements that the RT will annotate with control labels.

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 multiround experimental campaign to screen yeast circuits.

The experimental designer will specify the control table (Table 1) to label specific measurements with control type labels. Downstream 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 24 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 wild-type 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 time points. The control table identifies which time points are used for the reference GFP values.

Table 1 is "Table 1" of several control tables (not shown) that are cross-referenced in the first column of the measurements table illustrated in Table 2. The control table maps control type labels to measurements (by strain and

Table 2. Round 4.0 Measurement Table^a

| Control | Media | Measurement Type | Ffile Type | Replicate | Strain | Time point (h) | Temperature (°C) | |
|--|--|------------------|---------------|-----------|-------------------------------------|---------------------|---------------------|--|
| Table 1 | SC media | PLATE_READER | CSV | 6 | UWBF_6390, W303 | 0, 8, 12, 16, 18 | 30 | |
| Table 4 | SC media | FLOW | FCS | 6 | UWBF_6390, W303 | 8, 12, 16, 18 | 30 | |
| | | | | | | | | |
| | SC media, rich_media, high_osm_ media | PLATE_READER | CSV | 3 | UWBF_6389, UWBF_7375, UWBF_8225, | 0, 8, 12, 16, 18 | 30 | |
| | SC media, rich_media, high_osm_ media | FLOW | FCS | 3 | UWBF_6389, UWBF_7375, UWBF_8225, | 8, 12, 16, 18 | 30 | |
| | | | | | | | | |
| ^{a} Each row in the measurement table describes a set of expected measurements to be produced by the ER. | | | | | | | | |

| Table 3. Expansion of the Measurements Table in Ta | able | 2 |
|--|------|---|
|--|------|---|

| Media | Measurement Type | File Type | Replicate | Strain | Time point (h) | Temperature (°C) |
|----------|------------------|-----------|-----------|-----------|----------------|------------------|
| SC media | PLATE_READER | CSV | 1 | UWBF_6390 | 0 | 30 |
| SC media | PLATE_READER | CSV | 1 | UWBF_6390 | 8 | 30 |
| SC media | PLATE_READER | CSV | 1 | UWBF_6390 | 12 | 30 |
| | | | | | | |

conditions). That is, any measurement that is consistent with the description of a control type label will receive the label. The control 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).

The control labels support downstream 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).

The experimental designer will also specify a measurement table (Table 2) to identify every measurement that the experiment shall produce. In the Round 4.0 experiment, the experimental designer requests measurements of three types of media (possible sources of variation) with a plate reader (at five time points) and flow cytometry (at four time points). 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 control table will be used to label the measurements described by a row in the measurement table because each row in the measurement 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:

{SC media} × {PLATE_READER} × {CSV} × {1, 2, 3, 4, 5, 6} × {UWBF_6390, W303} × {0, 8, 12, 16, 18} × {30}

where the low and high controls are grown in SC media and measured by a plate reader at 0, 8, 12, 16, and 18 hours and at

30 °C. These 60 data points correspond to the rows of an expanded table with one measurement per row, as in Table 3.

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

Table 4. Round 4.0 Parameters Table^a

| Parameter | Value | | | |
|-------------------------------------|-----------------------------|--|--|--|
| protocol | TimeSeries | | | |
| inoculation volume | 10 <i>µ</i> L | | | |
| inoculation media volume | 700 <i>µ</i> L | | | |
| | | | | |
| 'The parameters table identifies | the protocol id and several | | | |
| parameter-value pairs used to confi | gure the protocol. | | | |

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 parameter table values with runspecific 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

Table 5. Example Plate Reader Data Table^a

| Media | Replicate | Strain | Time point (h) | Temperature (°C) | Optical Density | Fluorescence |
|----------|-----------|-----------|----------------|------------------|-----------------|--------------|
| SC media | 1 | UWBF_6390 | 0 | 30 | 0.258 | 20.491 |
| SC media | 1 | UWBF_6390 | 8 | 30 | 0.260 | 26.066 |
| SC media | 1 | UWBF_6390 | 12 | 30 | 0.373 | 38.534 |
| | | | | | | |

^aData tables combine measurement metadata with raw data to support analysis tools.

1. **Experiment Inception**: The experiment designer determines the hypothesis and experimental method in conjunction with a data scientist.

- 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.
 Annotating the Experiment Request: The RT annotates the ER with hyperlinks from experiment terms to new or existing SBOL(2) definitions stored in SynBioHub (18) through a Data Dictionary.
 Structuring the Experiment Request: Linked constructs and experiment tables are converted into a structured request, formally defining a set of minimal requirements
- 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 ETL 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 4. RT facilitates several steps in the experimental process from experimental inception to the presentation of the results.

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 time points to calculate a growth rate. Its output is another table that eliminates the time point column and adds a growth rate column. The RT analyses either aggregate over columns such as time points or reagent concentrations to compute new columns containing their outputs or compute per data point output values that extend the table columns.

2.2. Round Trip Software Architecture. The RT automates metadata consolidation, experimental execution, data tagging, and analysis. Its implementation comprises several software components and intermediate data formats needed to implement the process. The primary existing software used by the RT includes SynBioHub,¹⁸ SBOL,² TASBE,²² and the Strateos platform³ (many of which were advanced as part of developing the RT). The remaining components were developed as part of the SD2 project.

Due to the nature by which the components interact, we describe the RT by the processing stages introduced in Figure 1. Figure 4 summarizes each stage. The following describes each stage in detail, highlighting the behavior of each relevant software component.

2.2.1. Inception. The inception stage is the experimental designer-driven part of the scientific method that involves generating a hypothesis and considering the experimental method needed to validate the hypothesis. While potentially capable of being extended to a closed-loop experimental tool, the RT does not play a direct role in experimental inception. Rather, the RT indirectly influences the inception through the type and nature of the experiments made available to the experimental designer. In particular, the RT implementation reported here currently includes five configurable experimental protocols, described in Section 4.2.

In the Round 4.0 experiment, the inception stage required the experimental designers to consider their original intent to identify sources of variation and the results from previous rounds. Round 3.0, the previous round, demonstrated more variation in strain behavior (circuit output signal) from time point 12 to 24 h than from 24 to 36 to 48 h (separated by 12 h intervals). The experimental designer developed Round 4.0 to better characterize variation in the time to reach log phase growth and peak expression by requesting finer time points between 8 and 24 h (separated by 6 h intervals). The experimental designer also omitted the slow media type used in Round 3.0 due to its relatively lower variation over this time period. While our experiment was designed manually, automated experimental design techniques (e.g., based upon Bayesian optimization²³) can use RT data sets to make similar determinations about when to measure.

2.2.2. Specification. The specification of an experiment involves creating an ER. In addition to the controls, measurements, and protocol parameter tables, the ER includes several pieces of information that describe the context of the experiment. The primary sections are the front matter (describing points of contact, date, *etc.*), the goal of the experiment, the rationale for the experimental method, a description of the protocol, the identified stakeholders, and any risks and mitigations.

The Round 4.0 ER configures a time series experiment with three media types, two controls, 24 strains, two measurement types, and five time points (*i.e.*, 1440 total reporting conditions and either three or six replicates per strain). Before the RT can process these data, it seeks to annotate the terms with their definitions. For example, to assess strain performance, the RT must know the input—output state expected for a specific logic circuit that strain UWBF_7299 implements.

2.2.3. Annotation. The annotation of the terms appearing in an ER is critical to avoid ambiguity and provide a longterm/reusable data set for the experiment. What the experimental designer describes as "SC media" may in fact correspond to what the laboratory technician describes as "synthetic complete media" and the LIMS indexes under the machine identifier rslb4uj7zbdy6m. Moreover, the data scientist that determines the impact of media upon growth rates may not be able to distinguish these terms from publicly available descriptions of the media ingredients. Similarly, automated analyses require additional information about the terms, such as the intended logic function implemented by a strain.

The RT solves this coreferencing and data linkage challenge with three software components: the Intent Parser,¹⁶ the Data Dictionary,¹⁷ and SynBioHub.¹⁸ The Intent Parser (assisted by its Google Docs add-on client) provides several features for annotating the ER. The Intent Parser identifies terms in the ER, flags them for annotation, and provides possible quick-fix actions. The Data Dictionary and SynBioHub store data that backs the Intent Parser business logic. The Data Dictionary lists *common terms* and synonyms, unique SynBioHub URIs defining the terms, and laboratory-specific nomenclature and identifiers. SynBioHub provides unique URIs for common terms that can be bare representations of the terms, SBOL² representations (*e.g.*, the genetic sequence, expected gene products, and regulatory model), or links grounding in external databases such as ChEBI or UniProt.

Through a simple process facilitated by the Intent Parser add-on, the experimental designer can hyperlink each term with a SynBioHub URI. For each term, the Intent Parser provides a list of possible common terms and URIs that are close matches. The experimental designer can either select one of these suggestions or add his/her own. The Intent Parser adds new common terms by automatically populating the Data Dictionary and SynBioHub for the experimental designer. The experimental designer can provide synonyms and SBOL representations of the term either at the creation time or at any later time. A full annotation of the terms present in the ER provides additional metadata for later analysis. For example, the RT helps annotate the strain identifier UWBF_7299 with a URI https://hub.sd2e.org/user/sd2e/design/UWBF_7299/1.

Annotation is the first step to ensuring that RT data products can not only be produced without ambiguity but also later be analyzed reliably. While there is opportunity for human error, it is isolated to the specification and annotation stages. Automation carries the metadata linked by the annotations through the RT so that no further compounding of human error can occur. Likewise, corrections to the annotations can be automatically pushed through the RT to avoid multiple, costly, human-driven edits. With a fully hyperlinked ER, the RT proceeds by generating a formal representation of the ER, called a Structured Request.

2.2.4. Structuring. When the experimental designer is ready to execute the experiment, they use the Intent Parser client to submit the experiment. The first step of submission involves converting the semistructured ER into a Structured Request. The Structured Request is a stand-alone JSON representation of the ER that relates the controls (C), measurements (M), and parameters (P) along with hyperlinks, common names, and synonyms for all terms.

The Intent Parser, Structured Request Generator, and Data Dictionary coordinate to populate the Structured Request. Through the Intent Parser client, the experimental designer invokes the Structured Request Generator to send the annotated ER as input. The Structured Request Generator populates the Structured Request with an expansion of the Measurement Table into a set of measurements M (similar to the content in Table 3) and further annotates the Structured Request with laboratory specific terms. While the ER contains common terms that are hyperlinked to their SBOL representations in SynBioHub, the lab (i.e., Strateos) chosen for experimental execution will have its own terms, often opaque LIMS identifiers. The Data Dictionary labels synonyms of common terms with those of the laboratory technician. (An added benefit is the ability to repurpose ERs for execution at multiple laboratories because the Data Dictionary resolves the terms.) With a complete Structured Request, the Structured Request Generator submits the experiment for planning.

2.2.5. Planning. The XPlan experimental planner¹⁹ determines how to configure the protocol to execute the experiment, over several experimental run launch requests. XPlan also produces additional metadata that describes how its decisions impact each measurement (*e.g.*, which lab container and well contain the sample). XPlan returns the metadata to the Structured Request Generator so that it can consolidate the metadata in the Structured Request.

XPlan decides several low-level details governing the experimental execution to reduce burden on the experimental designer. Most notably, XPlan: (1) partitions the measurements into experimental runs, (2) assigns reagent concentrations and replicate ids to container wells, and (3) enforces constraints to ensure the ER can be accomplished with the available lab resources.

XPlan encodes this decision problem as a Satisfiability Modulo Theories (SMT) problem²⁴ and applies the z3 solver.²⁵ The SMT problem seeks to assign a set of variables that satisfy several constraints, including: each requested measurement is produced by a well of a container of a run, each container in a run obeys experiment-wide constraints (*e.g.*, uses the same temperature because containers from multiple runs occupy the same incubator), each well in a column has the same reagent concentrations (*e.g.*, if the protocol uses multitip pipettes that address columns), and each well of a container produces the same measurements (*e.g.*, a plate read at 4 h, and flow cytometry read at 8 h).

If satisfiable, XPlan extracts a measurement to run mapping from the assignment found by the SMT solver. This additional metadata also helps XPlan create and submit the experimental run launch requests for execution at the lab. The lab responds with a run identifier for each, and XPlan attaches the measurement to run mappings to the Structured Request. At this point, the Structured Request has all the information needed to link the requested measurements with the data to be produced by the lab.

The Round 4.0 experiment uses the TimeSeries protocol, which implies several types of constraints. Experiment-level constraints state that each run uses the same temperature (30 $^{\circ}$ C) due to a shared incubator. Run-level constraints state that all measurements from the same run must use the same media due to liquid handler restrictions and that the measurement time points must fall within allowed ranges and frequencies. Sample-level constraints state that each measurement of the same replicate of a strain is made from the same microplate well and that there is one sample per well. While not exercised in the Round 4.0 experiment, the TimeSeries protocol has column- and row-level constraints that require all wells within a column or row to receive the same reagent concentration.

2.2.6. Execution. While an ER may require many measurements, XPlan partitions the measurements into runs (physical batches). In this way, the lab does not need to know about the ER and the experimental context. The RT provides the lab with all the information needed to initiate a run. In principle, this permits the RT to use multiple laboratories to produce the requested measurements. In practice, the RT is currently integrated with the Strateos robotic cloud laboratory and the extension to additional laboratories requires similar mappings from the ER to the protocol launch.

The run launch parameters specify the protocol, protocol configuration values, materials (*e.g.*, reagents and containers), and measurements. The lab uses the parameters to configure and run the experiment. The experiment results in two types of data. The first is a sample trace, describing the generated measurements (*e.g.*, the file corresponds to measuring a well at a given time). The second is the raw measurement data, such as CSVs of plate reader data or FCS files produced by a flow cytometer.

2.2.7. Acceptance. The Acceptance stage compares the lab results with the expected measurements that the RT prepared prior to the Execution stage. The Structured Request Generator computes the difference and produces a summary. This summary lists both unrequested and unfulfilled measurements in terms of their metadata. It also provides descriptive statistics, such as the counts of measurements fulfilling different experimental factors.

The RT proceeds automatically beyond the Acceptance stage if the Structured Request Generator is able to correctly align the expected and actual experiments. If not, it provides the summaries to the experimental designer and laboratory to correct the deviations. The correction process involves altering the expected samples by updating the ER or updating the experimental data produced by the lab. In both cases, the RT recomputes any intermediate data and repeats the measurement alignment process.

While not exercised in the Round 4.0 experiment, the Acceptance stage has been used to identify additional measurements produced by the lab. For example, if an ER omits a strain that is already present on the microplate used to initiate an experiment, the lab will report unexpected plate reader measurements for this strain. Similarly, the lab may omit measurements, such as those taken at a particular time point.

2.2.8. Extract Transform and Load. The ETL stage involves storing the experimental data in a cloud file system hosted at TACC through an S3 bucket, ingesting the measurement metadata into a Data Repository, and running data cleansing pipelines. Due to its size, the storage of the experimental data can be challenging. TACC stores data on its cloud platform and provides several important utilities, including backups, access protocols, and high performance computing nodes for preliminary data analysis. TACC also hosts the Data Repository that records all experimental metadata. The Data Repository is a MongoDB database (a NoSQL database) that represents the Structured Request and sample trace metadata provided by the lab, which can vary by protocol and experiment. The flexibility of the database supports easy data ingest. In contrast, the following Standardization stage supports ease of analysis by consolidating the data into a format that does not vary by protocol and experiment. The final ETL steps correspond to data cleansing software such as RNASeq pipelines and TASBE flow cytometry tools²² that produce additional data products.

2.2.9. Standardization. The Standardization step draws upon the ETL data products to create several AI-ready data tables, as demonstrated in Table 5. The provision of a consistent set of columns for each table, along with standard units and data value conventions, enables one to compare analysis techniques across experiments. It also reduces the burden of creating and applying automated analysis. The RT achieves standardization with the Data Converge component.

Upon ETL completion, Data Converge runs automatically. Data Converge also runs as needed as requests for additional data table columns arise. Data Converge also retains versions of every data table it produces to maintain backward compatibility. The standard data products it creates include: raw and log transformed flow cytometry event-level and summary data tables, metadata for each flow cytometry data table that describes the columns and their value statistics, plate reader measurements, and RNASeq data.

The metadata added by Data Converge originate in the Data Repository and include: the strain (common name and SynBioHub reference), input and output states (for genetic circuits), protocol, time point, replicate id, media, inducer concentrations, instrument settings, date of experiment, laboratory, and cell counts. This metadata aims to capture the essential parameters of an experiment along with the biological constructs, reagents, and measurements. Data Converge products are the single point of reference for all tools in the Analysis stage.

2.2.10. Analysis. The Analysis stage automatically follows Data Converge completion and runs several analysis algorithms in parallel. Each analysis algorithm reads Data Converge data products and writes a variety of analysis results. Each result is a data table that includes the same metadata columns produced by Data Converge and additional columns that constitute the analysis output. Each analysis technique may also aggregate over Data Converge data table rows (*e.g.*, aggregate over rows corresponding to a time series of measurements for each sample).

The process of applying the analyses is overseen by the Precomputed Data Table component. The Precomputed Data Table matches data products with analyses, determining which pairs are compatible. For example, the Precomputed Data Table executes growth curve and doubling time analyses upon plate reader data and several fluorescence-based analyses upon

| 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 |

Table 6. Summary of the On-the-Loop Experimental Campaign^a

^aDuration and turnaround are measured in days. The round 2.0 turnaround is measured with respect to round 1.0.



Figure 5. RT produces data tables that report data points along multiple dimensions, including optical density and cells/mL. The respective box plots illustrate the per media and time point distributions of the replicates along these dimensions.

flow cytometry data. For each analysis and data pair, the Precomputed Data Table manages the execution, result storage, and dashboard reporting.

2.2.11. Presentation. The RT finishes its processing pipeline with visualizations of the Analysis stage results in the Presentation stage. The visualizations are organized by the Escalation software component.²⁰ Escalation uses a web interface to provide interactive data exploration with several predefined, interactive plots meant to address specific experimental designer concerns. In addition to the visualizations provided by Escalation, data scientists may extend Escalation or develop their own visualizations from the Data Converge and Precomputed Data Table data products. For example, the TACC environment provides Jupyter notebookbased interfaces to the data products for interactive scripting of custom visualizations.

2.3. Use of Round Trip in SD2. 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.

2.3.1. On-the-Loop Discovery and Identification of Sources of Variation. The RT supports a rapid experimental investigation paradigm where the experimental 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 experimental designer to plan the round, on the basis of data from the previous round. While the overall duration and turnaround were impacted by unrelated laboratory and experimental 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 experimental 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.

Within the campaign, we sought to determine the degree to which the strains and circuits: (1) perform under different conditions, (2) replicate performance within experimental conditions, and (3) are robust to experimental conditions. Figures 5, 6, and 7 illustrate results from several different



Figure 6. RT supports analyses such as Circuit Scoring that aggregate all measurement data across multiple ERs. Circuit Scoring computes the mean rank of each circuit: how well the observed circuit function implements the intended circuit (rank 1 indicates the circuit implements the intended circuit function better than all other possible alternative circuits).



Figure 7. RT automates several analyses, including Data Diagnosis. Data Diagnosis measures the association of performance with experimental variables, such as strain, time point, or media. Each plot shows the performance ratio data grouped by variable.

analysis techniques, enabled by the RT. The analyses, which were taken from Round 4.0, include optical density and estimated cells/mL per media and time point (Figure 5), circuit function ranking over all conditions (Figure 6), and variation in circuit performance aggregated by alternative experimental factors (Figure 7). The first two analyses were produced by data scientists from data tables within the data set created by the RT. Figure 5 illustrates how RT data tables, which record per data point measurement values, can readily identify per media and time point distributions over replicates. Figure 6 illustrates how more advanced analysis techniques can produce high-level summaries. The circuit scoring method ranks how well the truth table for each of the six Boolean logic circuits best matches the measurements for a given circuit. The figure illustrates the rank of the expected circuit against the measurements for the circuit across all strains, replicates, and conditions. This summary highlights that the OR and NOR circuits are not behaving as expected because their true and observed ranks are significantly different. Figure 7 illustrates three plots generated by the RT analysis called Data Diagnosis. It shows how a tool automated by the Precomputed Data Table can compute useful summaries of the data set. Data Diagnosis computes the distribution of the ratio of the fluorescence (relative to the low control) of each dimension of the data set, including strain, time point, and media. The pvalue for each distribution signifies how poorly the dimension explains variation in the performance.

The analyses helped to answer the questions about performance under different conditions, replicability, and robustness. These analyses not only helped our data scientists reach conclusions about the strains but also guided the rounds of experimentation. Most notably, they helped to identify the most promising growth conditions and time points to use for sampling. Analyses such as these impact the RT workflow by providing experimental designer feedback that can result in three outcomes: (1) develop a new ER to gather additional measurements, (2) end the experimental campaign, or (3) redesign the strains or biological materials, followed by gathering additional measurements with the RT.

2.3.2. 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 8 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 experimental data to produce data sets. Each experimental 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 to mid 2019), data was uploaded in a preliminary format. Figure 8 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 experimental designers creating samples to screen with the RT due to COVID-19 mandates. The ingest date depicted by Figure 8 over this period is much more closely aligned with the experimental date. The experimental frequency afforded by the RT was 0.29 experiments per day over Phase 2 and 0.39 experiments per



Figure 8. 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 experimental 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 experimental data upload by the program phase of SD2.

day over Phase 3, representing a $2.05 \times$ 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 8 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 the RT introduction, the rate of data product generation was 495.54 files per day, a 1.70× increase in the rate of data product generation. The second inflection point indicates the time at which the RT data set production was made operational for a large set of experiments.

Finally, the second right axis in Figure 8 illustrates the estimated time to produce an analysis of an experiment after the data was uploaded. The 4.20× speed-up from Phase 1 to Phase 2 corresponds to the introduction of the RT to automate metadata creation with manually triggered, yet standardized analysis. The final $30.10\times$ speed-up from Phase 2 to Phase 3 was achieved by automating all of the metadata creation, data standardization, and analysis. Overall, from Phase 1 to Phase 3, the SD2 RT elements contributed to a $126.10\times$ speed-up in producing the analyses of the experiments.

3. DISCUSSION AND FUTURE WORK

3.1. Discussion. 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 primary contributions of the RT are flexible knowledge representations and automated processing steps that support experimental 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.

The RT in its current form is available for use on the TACC infrastructure and through a public Github organization. Future projects can readily build upon the RT extensible architecture to include new protocols, laboratories, data types, and analyses. For example, we adapted the RT to a DBTL loop for screening cell-free riboswitches within two months, including a new protocol and related ER measurement table structure and sample trace format. As the RT is based on containerized software components that do not rely strongly on any particular cloud platform, future usage can also include transition to other public or private computational environments.

We have run several pilot experiments with prospective government transition partners (and are seeking industry and academic partners), which has motivated some of the future work items below. Broadening the adoption of the RT also requires additional standardization around the support for new laboratories, protocols, and data analyses. Currently, the integration of this new type of functionality requires software development rather than declarative interface specifications.

3.2. Future Work. There are many possible directions for future work, including alternative software hosting, protocol authoring, and increased experimental design automation.

While developed for cross-organization collaboration, the RT software can be adapted to benefit organizations that require local hosting of software components. The primary extensions include (1) replacing the Google Docs interface for the ER with a similar document sharing solution, such as Markdown and Git, and (2) extending the support for additional laboratory interfaces for launching protocol runs.

To support experimental designer authoring of protocols, we have developed the Protocol Activity Modeling Language (PAML).¹⁵ PAML is an open standard built upon the UML Activity Model.²⁶ It uses primitives that map onto Autoprotocol instructions and other protocol description formats. We have demonstrated automated submission and execution of end-user authored protocols that translate PAML into Autoprotocol for Strateos runs. Future work on PAML and the RT could combine protocol and ER authoring as well as simplify the Planning, Acceptance, ETL, and Standardization stages.

Finally, the RT has the potential to support increased experimental design automation by lifting the ER specification to make hypotheses and experimental intent machines processable. From these high-level specifications, an automated experimental design algorithm could identify the controls, measurements, and protocol parameter tables. Furthermore, the Analysis stage can be more finely tuned to address the hypothesis with the experimental data. Another potential improvement to the Analysis stage would be to automate or support users to chain analyses.

4. METHODS

4.1. Computational Infrastructure. Computational infrastructure for the RT is implemented using technologies from TACC and Strateos. The RT is implemented as an integrated set of software components that are deployed on the TACC high performance computing platform. The RT configures, initiates, and analyzes data from several experimental protocols on the Strateos robotic cloud laboratory platform.

TACC provides several high performance computing (HPC) clusters that are accessible through the SD2E portal (www.sd2e.org). RT components that run on the clusters are orchestrated through the Tapis platform.²⁷ Tapis brings together public, private, and shared HPC, high-throughput computing (HTC), Cloud, and Big Data resources under a single, web-friendly REST API. The RT uses Tapis to configure storage systems (for controlled access to data and metadata) and execution systems (for allocating jobs and applications to specialized hardware). Orchestration happens through the Tapis app and actor frameworks.

An app, in the context of Tapis, is an executable code available for invocation through the Tapis Jobs service on a specific execution system. Apps are language agnostic, providing an interface to containerized code. TACC provides an apps service, a central registry for all Tapis apps that provides discovery services and permissions, validation, archiving, and revision information about each app. The RT uses apps for compute-heavy operations, such as XPlan SMT solving, Data Converge data processing, and Precomputed Data Table component analyses.

The Tapis actor framework is similar to that of apps but focuses on messaging capabilities and lightweight computing. For example, the Structured Request Generator, XPlan, and Precomputed Data Table use actors to send and respond to messages as well as dispatch jobs to apps. Both XPlan and Precomputed Data Table respond to requests to process new experiments and dispatch heavy-weight processing to apps. The actor and app interaction methodology provides the capability to respond to heterogeneous use cases and scale-up computing requirements as needed.

In addition to the TACC Tapis platform, the RT provides access to a project dashboard to gain better insight into experimental processing status. The dashboard lists each experiment and batch along with timestamps for important processing steps and links to intermediate data produced (*e.g.*, ERs, Structured Requests, laboratory data, ETL, Data Converge, and Precomputed Data Table products).

4.2. Protocols. Five protocols were used by the RT to produce the data reported in the results. These protocols are implemented by Strateos at their robotic cloud laboratory and named HarmonizedYeastGates, GrowthCurve, TimeSeries, ObstacleCourse, and Cell-Free-Riboswitches. There are several similarities between the protocols. Each protocol generates plate reader and/or flow cytometry measurements. Most protocols require an initial microplate that contains the experimental strains and begin with an "overnight" growth

phase to help samples recover from cold storage. This is followed by a growth phase with various time point measurements. The details of how to configure each protocol can be obtained through the Strateos API (https://github. com/strateos/transcriptic), which specifies a JSON-based schema for the available parameters.

The HarmonizedYeastGates protocol was developed early in the SD2 project to investigate cross-laboratory reproducibility through a common protocol executed at three geographically separated sites. The protocol begins with an overnight incubation of a glycerol stock microplate plus plate reader measurements to determine the optical density of each sample. The protocol uses the optical density to dilute each well to meet a configurable target optical density. After dilution, the protocol incubates the samples for a configurable duration and ends with a plate reader and flow cytometry measurement.

The GrowthCurve protocol provides a time series of plate reader measurements to characterize the growth rate of the samples arrayed on a microplate. The protocol begins with overnight growth, similar to the HarmonizedYeastGates protocol, followed by incubation with a configurable series of time points at which plate reader data is collected.

The TimeSeries protocol builds upon the GrowthCurve protocol capabilities by additionally offering flow cytometry measurements at configurable time points. The protocol also offers a configurable series of recovery steps that collect plate reader measurements. The recovery steps are followed by an induction step that permits addition of a reagent in a columnwise configurable concentration to the microplate. Following induction, the protocol takes plate reader and flow cytometry measurements at specified time points.

The ObstacleCourse protocol was developed to screen genetic circuits with one or two inputs. It collects data on the behavior of samples over several days (following an overnight growth). Each day, the protocol dilutes samples by a configurable factor and introduces configurable quantities of two reagents (inducers). Each day, the protocol takes two plate reader measurements (post-induction and end of day) and one flow cytometry measurement (end of day).

The Cell-Free-Riboswitches protocol arrays a number of DNA fragments and cell-free execution materials at configurable concentrations onto microplates along with a reagent for the riboswitches to sense at varying concentrations. The protocol conducts several plate reader measurements over a time series.

ASSOCIATED CONTENT

③ Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acssynbio.1c00305.

Experimental Requests: description of the sections of an ER, as developed as part of the SD2 program and adopted for use by the RT, and list of the Round 4.0 Experiment Request (ER); Demonstration: highlights of the RT with several annotated screenshots of how the RT components process an ER to generate data sets; Precomputed Data Table Analyses: details of the analysis tools integrated with the Precomputed Data Table; Presentation Reports: list of the Escalation reports generated for RT analyses (PDF)

Special Issue Paper

Invited contribution from the 12th International Workshop on Bio-Design Automation.

AUTHOR INFORMATION

Corresponding Author

Daniel Bryce – SIFT, LLC., Minneapolis, Minnesota 55401, United States; o orcid.org/0000-0001-5809-3658; Email: dbryce@sift.net

Authors

- **Robert P. Goldman** SIFT, LLC., Minneapolis, Minnesota 55401, United States
- Matthew DeHaven SIFT, LLC., Minneapolis, Minnesota 55401, United States
- Jacob Beal Raytheon BBN Technologies, Cambridge, Massachusetts 02138, United States; © orcid.org/0000-0002-1663-5102
- Bryan Bartley Raytheon BBN Technologies, Cambridge, Massachusetts 02138, United States; © orcid.org/0000-0002-1597-4022
- Tramy T. Nguyen Raytheon BBN Technologies, Cambridge, Massachusetts 02138, United States; © orcid.org/0000-0002-8035-0259
- Nicholas Walczak Raytheon BBN Technologies, Cambridge, Massachusetts 02138, United States
- Mark Weston Netrias, Inc., Annapolis, Maryland 21409, United States
- **George Zheng** Netrias, Inc., Annapolis, Maryland 21409, United States
- Josh Nowak Strateos, Inc., Menlo Park, California 94025, United States
- **Peter Lee** *Ginkgo Bioworks, Inc., Boston, Massachusetts* 02210, United States
- Joe Stubbs Texas Advanced Computing Center, Austin, Texas 78758, United States
- Niall Gaffney Texas Advanced Computing Center, Austin, Texas 78758, United States
- Matthew W. Vaughn Texas Advanced Computing Center, Austin, Texas 78758, United States
- Chris John Myers Colorado University, Boulder, Colorado 80203, United States; o orcid.org/0000-0002-8762-8444
- Robert C. Moseley Duke University, Durham, North Carolina 27708, United States; Occid.org/0000-0002-4615-457X
- Steven Haase Duke University, Durham, North Carolina 27708, United States
- Anastasia Deckard Geometric Data Analytics, Inc., Durham, North Carolina 27701, United States
- Bree Cummins Montana State University, Bozeman, Montana 59717, United States
- Nick Leiby Two Six Technologies, Inc., Arlington, Virginia 22203, United States

Complete contact information is available at: https://pubs.acs.org/10.1021/acssynbio.1c00305

Author Contributions

Conceptualization: D.B., J.B., M.W., P.L., N.G., and S.H. Data curation: D.B., R.P.G., M.D., J.B., B.B., T.T.N., N.W., M.W., G.Z., J.N., P.L., C.J.M., R.C.M., S.H., A.D., and B.C. Formal analysis: D.B., R.P.G., J.B., M.W., G.Z., J.N., P.L., N.G., M.W.V., R.C.M., S.H., A.D., and B.C. Funding acquisition: D.B., M.W.V., C.J.M., and S.H. Investigation: D.B., J.B., M.W.,

and S.H. Methodology: D.B., J.B., B.B., T.T.N., N.W., M.W., G.Z., J.N., P.L., M.W.V., C.J.M., R.C.M., A.D., and B.C. Project administration: D.B., J.B., M.W., M.W.V., and S.H. Resources: J.N., J.S., N.G., M.W.V., and R.C.M. Software: D.B., R.P.G., M.D., J.B., B.B., T.T.N., N.W., M.W., G.Z., J.N., P.L., J.S., N.G., M.W.V., C.J.M., R.C.M., S.H., A.D., B.C., and N.L. Supervision: D.B., J.B., M.W., and S.H. Validation: D.B., M.W., and J.N. Visualization: D.B, J.B, M.W., N.G, R.C.M., A.D., B.C., and N.L. Writing, original draft: D.B. and J.B. Writing, review and editing: D.B., J.B., B.B., T.T.N., M.W., G.Z., R.C.M., A.D., and B.C.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by Air Force Research Laboratory (AFRL) and DARPA contracts FA875017CO184, HR001117C0092, HR001117C0094, FA8750-17-C-0229, and FA875017C0054. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations. Views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for public release, distribution unlimited (DISTAR Case 34715).

ABBREVIATIONS

DARPA, Defense Advanced Research Projects Agency; SD2, Synergistic Discovery and Design; RT, Round Trip; LIMS, Lab Inventory Management System; TACC, Texas Advanced Computing Center; ER, Experiment Request; GFP, green fluorescent protein

REFERENCES

(1) Jessop-Fabre, M. M.; Sonnenschein, N. Improving Reproducibility in Synthetic Biology. Frontiers in Bioeng. and Biotechnol. 2019, 7, 1-18.

(2) Roehner, N.; Beal, J.; Clancy, K.; et al. Sharing Structure and Function in Biological Design with SBOL 2.0. *ACS Syn. Bio.* **2016**, *5*, 498–506.

(3) Strateos Robotic Controlled Life Sciences Lab: Strateos Cloud Lab; https://www.strateos.com (accessed June 17, 2020).

(4) Klavins, E.; Keller, B.; Miller, A.; et al. aquariumbio/aquarium: Aquarium version 2.8.1; 2020; DOI: 10.5281/zenodo.4015532.

(5) Emerald Cloud Laboratory Emerald Cloud Lab: Remote Controlled Life Sciences Lab; https://www.emeraldcloudlab.com (accessed June 17, 2020).

(6) Edinburgh Genome Foundry *Edinburgh Genome Foundry*; The University of Edinburgh; https://www.ed.ac.uk/biology/research/facilities/edinburgh-genome-foundry (accessed June 17, 2020).

(7) DOE Joint Genome Institute DOE Joint Genome Institute: A DOE Office of Science User Facility of Lawrence Berkeley National Laboratory; https://jgi.doe.gov (accessed June 17, 2020).

(8) Broad Institute *Broad Institute;* https://www.broadinstitute.org (accessed June 17, 2020).

(9) Pendleton, I. M.; Cattabriga, G.; Li, Z.; et al. Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): a software pipeline for automated chemical experimentation and data management. *MRS Commun.* **2019**, *9*, 846–859. (10) King, R. D.; Whelan, K. E.; Jones, F. M.; et al. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* **2004**, *427*, 247–252.

(11) Riffyn Nexus Riffyn Nexus, The World's First Process Data System; https://www.riffyn.com (accessed September 20, 2020). (12) Synthace Synthace: Enabling life science, the way it should be done; https://www.synthace.com (accessed September 20, 2020).

(13) Benchling Cloud-Based Informatics Platform for Life Sciences R&D; https://www.benchling.com (accessed September 20, 2020).

(14) Radix Bio Complex Lab Work, Radically Simple; https://www.radix.bio (accessed September 20, 2020).

(15) Beal, J.; Bartley, B.; Biggers, V.; et al. Motivation and progress towards building a common protocol representation. In *COMBINE* 2021 *Community Meeting*; 2021.

(16) Nguyen, T.; Walczak, N.; Sumorok, D.; Weston, M.; Beal, J. Intent Parser: A Tool for Codification and Sharing of Experimental Design. *ACS Syn. Bio.* **2021**, in press DOI: 10.1021/acssynbio.1c00285.

(17) Beal, J.; Sumorok, D.; Bartley, B.; Nguyen, T. Collaborative Terminology: SBOL Project Dictionary. In *Proceedings of the International Workshop on Bio-Design Automation (IWBDA)*, 2020; pp 68–69.

(18) McLaughlin, J. A.; Myers, C. J.; Zundel, Z.; et al. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Syn. Bio.* **2018**, *7*, 682–688.

(19) Kuter, U.; Goldman, R. P.; Bryce, D. XPLAN: Experiment Planning for Synthetic Biology In *Proceedings of the 1st ICAPS Workshop on Hierarchical Planning*, 2018; pp 48–52.

(20) Leiby, N.; Zaitzeff, A. twosixlabs/Escalation: 1.0.0, 2021; DOI: 10.5281/zenodo.4660286.

(21) Gander, M. W.; Vrana, J. D.; Voje, W. E.; et al. Digital logic circuits in yeast with CRISPR-dCas9 NOR gates. *Nat. Commun.* **2017**, *8*, 1–11.

(22) Beal, J.; Overney, C.; Adler, A.; et al. TASBE Flow Analytics: A Package for Calibrated Flow Cytometry Analysis. *ACS Syn. Bio.* **2019**, *8*, 1524–1529.

(23) Mockus, J. Bayesian Approach to Global Optimization: Theory and Applications; Springer: Netherlands, 2012; DOI: 10.1007/978-94-009-0909-0.

(24) Barrett, C.; Tinelli, C. Satisfiability modulo theories. In *Handbook of Model Checking*; Springer, 2018; pp 305–343; DOI: 10.1007/978-3-319-10575-8.

(25) De Moura, L.; 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; Springer-Verlag: Berlin, 2008; pp 337–340; DOI: 10.1007/ 978-3-540-78800-3 24.

(26) Dumas, M.; ter Hofstede, A. H. M. UML Activity Diagrams as a Workflow Specification Language. In UML 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools; Springer, Berlin, 2001; pp 76–90; DOI: 10.1007/3-540-45441-1_7.

(27) Cleveland, S. B.; Jamthe, A.; Padhy, S.; et al. Tapis API Development with Python: Best Practices In Scientific REST API Implementation: Experience Implementing a Distributed Stream API. *Practice and Experience in Advanced Research Computing* **2020**, 181–187.

Recommended by ACS

A Network Approach to Genetic Circuit Designs

Matthew Crowther, Ángel Goñi-Moreno, et al. AUGUST 31, 2022 ACS SYNTHETIC BIOLOGY

READ 🗹

PASIV: A Pooled Approach-Based Workflow to Overcome Toxicity-Induced Design of Experiments Failures and Inefficiencies

Alexis Casas, Richard Kitney, et al. MARCH 09, 2022 ACS SYNTHETIC BIOLOGY

READ 🗹

Discovering Content through Text Mining for a Synthetic Biology Knowledge System

Bridget T. McInnes, Mai H. Nguyen, et al. JUNE 07, 2022 ACS SYNTHETIC BIOLOGY

READ 🗹

Intent Parser: A Tool for Codification and Sharing of Experimental Design

Tramy Nguyen, Jacob Beal, et al. DECEMBER 09, 2021 ACS SYNTHETIC BIOLOGY

READ 🗹

Get More Suggestions >