ACS Synthetic Biology

pubs.acs.org/synthbio

Perspective

A Framework for a Standard-Enabled FAIR Data Management Workflow for Synthetic Biology

Carolus Vitalis,[§] Gonzalo Vidal,[§] Sai P. Samineni, Pedro Fontanarrosa, and Chris J. Myers*

Cite This: *ACS Synth. Biol.* 2026, 15, 1−8

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Synthetic biology laboratories generate diverse forms of data and metadata throughout a project's life cycle, such as sequences, models, protocols, images, and time-series measurements. Unfortunately, these assets are scattered across spreadsheets, proprietary exports, custom scripts, etc. found in varied locations such as shared drives. Inconsistent metadata and data formats hinder provenance, reuse, security, compliance, automation, and scale-up. The central gap is a coherent way to link data, metadata, and code so they remain findable, accessible, interoperable, and reusable (FAIR). This perspective considers current practices through semistructured interviews with synthetic biology researchers in laboratories across the United States, and the findings were used to provide guidance to create a framework for an integrated data management workflow. This framework maps common data types to community standards that allow machine-accessible metadata, version control, and standards-compliant repositories. This perspective also offers a catalog of potential software solutions and stepwise adoption guidelines that turn the proposed framework into a day-to-day practice, democratizing the generation of standardized data. The result is that users gain a template that raises data to FAIR status, strengthens traceability for regulatory or defense contexts, and provides a stronger foundation for training machine learning models.

1. INTRODUCTION

Synthetic biology is an interdisciplinary field that aims to engineer biological systems. This field began about 25 years ago[1,2] and promised to leverage the engineering principles of standards, abstraction, and decoupling to accelerate discovery and innovation.[3] Fundamental to achieving these goals is the collection of high-quality, well-curated data. Synthetic biologists generate a wide diversity of data and metadata throughout each project's life cycle, including computational models, sequences, protocols, experimental measurements, images, etc. However, these records are usually created ad hoc, often in spreadsheets, plain text, or proprietary outputs, without adherence to community standards, resulting in inconsistent metadata and formats that hinder reuse and interoperability. Indeed, not even the sequences of the genetic circuits are always available[4] or provided in machine-accessible formats.[5] These issues limit innovation and progress in the field. *Machine learning* (ML) techniques are not sufficient to overcome these issues, as they require high-quality data to be effective.[6]

Advancements in both data standards and software systems make it possible to adopt methodologies that will fulfill the promise of synthetic biology. This perspective presents an assessment of the current state of data management within synthetic biology projects, a proposed framework for a data management workflow that leverages community-developed standards to capture diverse data in a unified, machine-accessible format, and a path forward to deploying this framework within the synthetic biology community. If adopted, this framework leverages its capacity to represent data across the entire project life cycle to create a *findable, accessible, interoperable, and reusable* (FAIR)[7] data management workflow.

This perspective targets academic and government laboratories first while noting adaptations for industrial and proprietary contexts (e.g., private instances, access control, and regulatory data governance). It provides practical guidance with clear rationales, integration patterns, and added security and traceability rather than hypothesis-driven results. This perspective aligns with recent calls for stronger sequence and information publishing practices and lab information management system guidance.[8]

2. CURRENT DATA MANAGEMENT PROCESSES

In order to understand the current state of affairs, we interviewed 25 people from 14 U.S.-based research groups to gather information about their data management practices. In particular, we asked them about the types of data they produced, the techniques they used to process and analyze the data, and how their data was stored, published, and shared. Our interviews were conducted as part of a funded research project and, as such, focused on the U.S.-based collaborators on this project. While we do not claim statistical generalizability, we did find the information helpful in understanding

Received: November 2, 2025
Revised: December 30, 2025
Accepted: January 2, 2026
Published: January 7, 2026

Table 1. Data Types and Formats at Each Stage of the Synthetic Biology Workflow[a]

stage	data types	current formats used	proposed standard formats	proposed data storage
resource	characterized parts, media, chassis	Excel sheets	SBOL	SynBioHub
design	genetic designs (Sequences, etc.)	FASTA/GenBank	SBOL/SBOL Visual	SynBioHub
model	computational models	Python/MATLAB	SBOL/SBML/SED-ML	SynBioHub
build	assembly plans	plain text	SBOL	SynBioHub
test	metadata	Excel sheets	SBOL	SynBioHub
	measurements	Excel sheets	Tidy Data	Flapjack
	experimental protocols	plain text scripts	LabOP/documented code	SynBioHub/GitHub
learn	data processing software	scripts	documented code/executable code	GitHub/Docker Hub
	statistical analysis	Excel sheets	SBOL/Tidy Data	SynBioHub/Flapjack

[a]The first column shows the stage. The second column lists the data types that need to be standardized for each stage. The third and fourth columns show the current and proposed standard formats for the data types, respectively. The last column indicates the data storage platform that can be used to store the data in the standard format. At the **Resource** stage, SBOL captures characterization information along with provenance for traceability. At the **Design** stage, SBOL captures the hierarchical compositions of parts, while SBOL Visual provides a means to visualize them. At the **Model** stage, SBML and SED-ML capture models and simulation instructions, enabling reproducibility of results across a variety of SBML-compliant simulators.[9] At the **Build** stage, SBOL is used to encode the build plan, while LabOP encodes the protocol along with reagents and equipment, enabling human or machine execution, referencing SBOL entities inside protocol steps. At the **Test** stage, SBOL encodes the experimental metadata, LabOP encodes the experimental protocols, and Tidy Data captures the measurements to make downstream analysis reproducible. Finally, at the **Learn** stage, the source code should be shared on GitHub, while the executables should be containerized and shared on Docker Hub to ensure reproducibility. Generated metadata and data from this code should be represented in SBOL and Tidy Data, respectively. Finally, SynBioHub centralizes SBOL, SBML, SED-ML, and LabOP artifacts, while Flapjack holds Tidy Data.

typical data management processes, their limitations, and potential means to support them going forward.

2.1. Data Types

Research groups and laboratories deal with diverse data types, including genetic designs, images, fluorescence measurements, molecular characterization data, assembly plans, and experimental protocols. Common file formats used include FASTA and GenBank for sequences, TIFF for images, and Excel for experimental measurements and related metadata, part libraries, and statistical analysis. Plain text is used to store information about the assembly plans and experimental protocols. Programming languages, such as Python and MATLAB, are used to encode models and automate build instructions. A list of the data types reported is shown in Table 1.

2.2. Analysis Tools

Reported workflows employ Excel spreadsheets and Jupyter notebook environments for basic tabulation and analyses and a set of specialized tools for particular bioinformatics tasks. USEARCH is used for amplicon read clustering, dereplication, and chimera filtering.[10] SPAdes is used for de novo assembly of short-read genomes and plasmids.[11] Bowtie2 performs fast gapped alignment of short reads to reference sequences.[12] antiSMASH detects and annotates biosynthetic gene clusters.[13] Geneious provides an integrated graphical environment for sequence visualization, alignment, and basic molecular biology tasks.[14] MATLAB and ad hoc Python scripts are used for numeric computation, scripting, and image or data manipulation.

2.3. Data Storage

Laboratories utilize various tools for data storage including Microsoft Teams, OneDrive, Dropbox, Zenodo, GitHub, and internal cloud storage. NCBI[15] and the Protein Data Bank[16] are used for data deposition and sharing. Some laboratories develop their own database platforms like the CAMII Biobank for microbial strain collections[17] and the MycoBank, which has fungal databases, nomenclatures, and species banks.[18]

2.4. Data Management Challenges

Laboratories commonly face challenges such as incomplete or missing metadata, nonstandardized data formats, and the need for data sharing and visualization platforms. Diverse data representation methods, including Excel, FASTA, GenBank, and plain text, lead to compatibility issues that hinder data interoperability across platforms, complicating communication among the stages of the workflow. Tight project deadlines discourage researchers from using standards properly due to the perceived steep learning curve. Often, the format used is dictated by the specific software tool or hardware in the wet lab, which can become part of group culture and increase resistance to change. This use of ad hoc formats and bespoke data management processes increases time and complexity due to the need to allocate resources for targeted training to standardize data management within a research group. Finally, the high volume of data produced—sometimes reaching terabytes per experiment—intensifies storage, organization, and retrieval challenges, especially in long-term projects. Indeed, ensuring data security while promoting collaboration poses another hurdle, particularly for laboratories reliant on custom storage solutions, which may lack robust security measures.

2.5. Proposed Data Management Workflow

At this time, synthetic biology has seen limited adoption of standards, with work typically done at the lowest level of abstraction and design still reserved for experts. While there has been some progress in the last 25 years, synthetic biology data is still not FAIR.[19] Furthermore, *posthoc curation* (i.e., after publication) of data is not sufficient, but rather *integrated curation* (i.e., during discovery) is needed to produce high-quality data for both current design methods and future AI/ML approaches.[20] Therefore, to increase the impact and accelerate the progress of synthetic biology discovery, a data management workflow that supports integrated curation is required.

2.6. Standards and Repositories

At the core of this workflow is the use of complementary standards. The main standard leveraged in our proposed

https://doi.org/10.1021/acssynbio.5c00813
ACS Synth. Biol. 2026, 15, 1−8

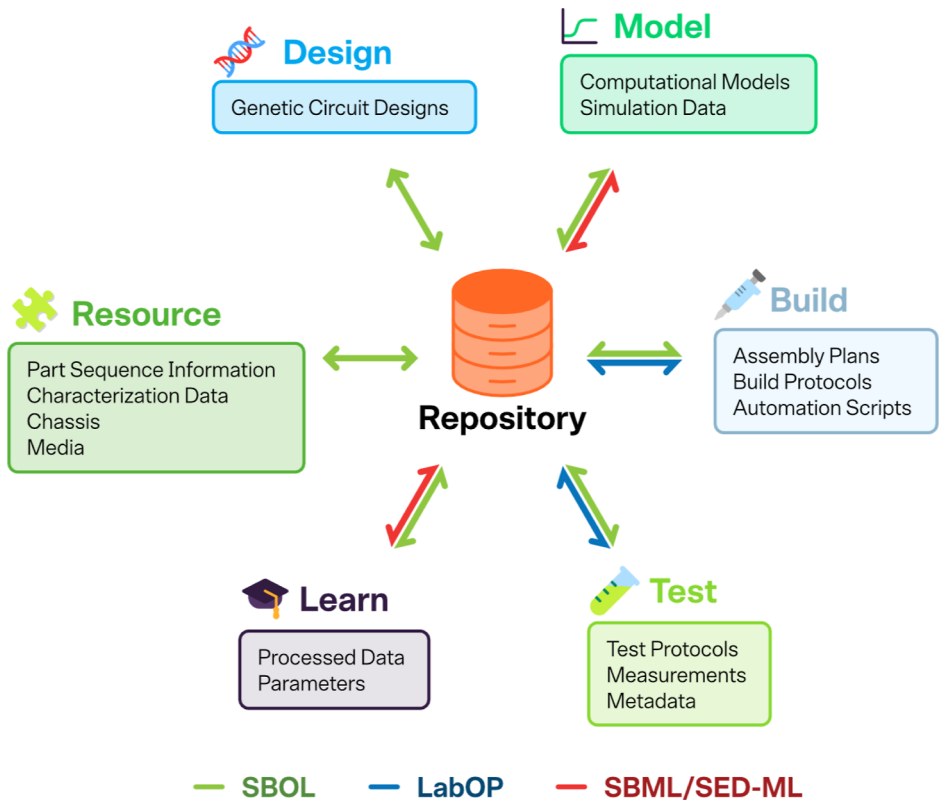

Figure 1. Proposed Framework for a Data Management Workflow. The framework has six stages (resource, design, model, build, test, and learn) that communicate using standards via a repository. In this framework, the Resource stage uploads information about resources used, encoded using SBOL, to a repository. In the Design stage, these resources are composed to produce genetic circuit designs that can also be stored as SBOL in the same repository. In the Model stage, computational models can be created using SBOL and SBML, with simulations encoded using SED-ML, and once again stored with the designs in the repository. In the Build stage, assembly plans encoded in SBOL and experimental protocols encoded in LabOP can be added to the repository. In the Test stage, metadata encoded in SBOL, experimental protocols encoded in LabOP, and measurement data encoded using Tidy Data can be added to the repository. Finally, in the Learn stage, the SBOL and SBML used for modeling can be updated with measured parameters.

workflow is the *Synthetic Biology Open Language* (SBOL), which explicitly supports multiscale design representation and integration across iterative workflows.[21−23] SBOL can capture diverse metadata in a unified, machine-accessible standard. This workflow also utilizes SBOL Visual, a companion standard to SBOL that specifies how genetic components should be represented in genetic circuit diagrams.[24] For modeling, this workflow uses the *Systems Biology Markup Language* (SBML)[25] and *Simulation Experiment Description Markup Language* (SED-ML)[26] to provide standardized, exchangeable encoding for biochemical network models and for the description of simulation experiments, respectively, enabling reproducibility and interoperability. The *Laboratory Open Protocol Language* (LabOP)[27] provides a formal representation designed to simplify the exchange of processes between laboratories. This standard is based on SBOL, which provides native compatibility and can be used as instructions for laboratory automation. LabOP has specializations to support execution as manual "paper protocols," by Autoprotocol, and as "automated protocols" by Opentrons. Finally, Tidy Data is a standardized way to organize tabular data where each variable is a column, each observation is a row, and each value is a single measurement.[28] This structure makes data sets easier to manipulate, model, and visualize, streamlining data analysis and ML training pipelines.

Standard data and metadata should be stored in repositories that natively support these standards. SBOL and LabOP are encoded in the *Resource Description Framework* (RDF) format, which makes them compatible with triplestore repositories, such as SynBioHub.[29] SynBioHub is an open-source, standard-enabled repository for synthetic biology that can be used to store and share resources (parts, chassis, media, etc.), genetic designs, computational models, assembly plans, protocols, and experimental metadata. It provides a web-based user interface and programmatic access through its *Application Programming Interface* (API) to upload designs, browse collections, search across instances, and generate shareable links for collaboration and publication. Non-RDF information, such as SBML, SED-ML, images, etc., can be referenced as attachments via the SBOL data format on SynBioHub. Measurement and analysis data in the Tidy Format can be stored in the Flapjack repository,[30] an open-source platform for storing, plotting, and sharing measurement data and links to SBOL-encoded metadata stored in SynBioHub. Finally, documented source codes and executables can be stored in GitHub (https://github.com) and Docker Hub (https://hub.docker.com), respectively.

2.7. Proposed Framework for a Data Management Workflow

Our proposed framework for a data management workflow consists of six stages: resource, design, model, build, test, and learn (see Figure 1). Table 1 shows the data types used for each stage, current formats used, and proposed standard data

https://doi.org/10.1021/acssynbio.5c00813
ACS Synth. Biol. 2026, 15, 1−8

formats and storage methods that could be used instead in a FAIR data management workflow. Adoption of these standards will provide a FAIR, machine-accessible representation that spans the full synthetic biology workflow.

In the Resource stage, researchers collect information about resources used in a synthetic biology project (characterized parts with sequences, chassis, media, etc.). We propose the use of SBOL to represent these resources. SBOL can encode the structural and functional information on these resources in a hierarchical fashion, allowing for the linkage to other resources. Furthermore, SBOL leverages ontologies (i.e., controlled vocabularies) to make resources findable and reusable. We also propose that these resource libraries are stored in a repository, such as SynBioHub.[29]

In the Design stage, researchers compose parts from their libraries into genetic designs. They also specify the desired chassis and media that should be used. These genetic designs start as abstract designs, which are a simple composition of parts and their order. We propose the use of SBOL to represent genetic designs. SBOL can represent the use of parts in a defined order, referring to the definition of the parts and constraining their position in the composite design without the need to specify all of the sequences. Furthermore, the use of SBOL to create both resource libraries and genetic designs allows for easy connection or reference between them and their storage in the same repository, such as SynBioHub.

In the Model stage, researchers use abstract designs, related characterization data, and parameters for modeling. This information is used as input for model parameterization. For example, we could parameterize Hill functions in *ordinary differential equations* (ODE) models and predict genetic gate match/mismatch on bulk cultures; we could also parameterize the production and degradation propensities of chemical species in *stochastic simulation algorithms* (SSA) and predict their effect on single-cell gene expression.[31] In general, models are used to run simulations of the behavior or function of biological systems. There is usually a back and forth between the Design and Model stages, as a design might not produce the expected function, so the design has to be changed until it meets the expected function. We propose the use of SBOL to represent interactions among parts of the abstract design. These interactions can encode, for example, the production of a protein from a CDS template and the inhibition of a promoter by a protein. We propose the use of SBML[25] and SED-ML[32] to represent biochemical models and simulation descriptions, respectively. The SBOL, SBML, and SED-ML for the computational models should be stored in a data repository such as SynBioHub.

At the Build stage, researchers construct DNA and optionally use it to transform a host organism or chassis. This stage involves the definition of the sequence to be constructed, developing a plan to assemble this sequence, and executing experimental protocols to produce this sequence. These protocols can be manual (executed by a human) or automated (executed with the help of a machine). Concatenating part sequences is sufficient for their construction using DNA synthesis. DNA assembly requires additional information, such as parts used in the assembly, restriction enzymes, and assembly scars. We propose the use of SBOL to represent metadata information such as the target DNA sequence and the assembly plan to construct this sequence.[33] We propose the use of LabOP for the representation of experimental protocols that are utilized during the build process.[27] The main use of SBOL in LabOP is to define reagents, and then, LabOP has specialized classes to represent equipment, steps, and processes. Assembly plans in SBOL can be connected to assembly protocols encoded in LabOP, creating a seamless transition from the intent at Design to the process of constructing it at Build. The assembly plans encoded in SBOL and the protocols encoded in LabOP can be stored in a data repository such as SynBioHub. Any code used with laboratory automation should be shared using a code repository such as GitHub.

At the Test stage, researchers run wet-lab experiments to measure the behavior or function of the constructed genetic design. One of the most common tests is the measurement of the fluorescence and optical density of samples using plate readers. We propose the use of SBOL for representing metadata, such as media, chassis, and chemical supplements added to each sample. We propose the use of LabOP for representation of the experimental protocols used for the test. We propose the use of Tidy Data,[28] when possible, for the representation of experimental measurements. The metadata encoded in SBOL and the protocols encoded in LabOP can be shared using a data repository such as SynBioHub. Measurement data should be shared in an experimental data repository, such as Flapjack.[30] Other data types can be directly attached to SBOL objects, and documented code used with laboratory automation should be stored in a code repository such as GitHub. Finally, we encourage metadata collection following the *Investigation, Study, Assay* (ISA) metadata framework.[34] ISA is an open, community metadata framework for describing, managing, and exchanging experimental metadata across life sciences.

At the Learn stage, researchers process and analyze the experimental data and compare it with simulation results from the earlier Model stage. This stage can take many forms, as it depends on the models and experimental validation used from previous stages, as well as the data processing and analysis approach. Some differences that may arise in this process are the choice of background correction, calibration, data type, and characterization algorithm. Following the plate reader example, the fluorescence time series can be processed in different ways to obtain the gene expression rate. An indirect approach smooths the data and takes the derivative of the resulting signal.[35−37] This approach is not computationally intensive but is sensitive to noise. A direct approach takes data as is and processes it using linear inversion[38] or inverse problems.[39] This approach is more computationally intensive but is less sensitive to noise. Some of these data processing approaches are supported in the Flapjack software tool.[30] The algorithm used to process the data must be clear and shared to increase the reproducibility of this stage. We propose the use of SBOL to encode characterized parts or genetic designs with their sequences and link them to parameters and characterization information, and this information can be shared using SynBioHub or similar. The SBML models created in the Model stage can be updated with the measured parameters. Finally, any documented code developed to process the data should be shared using a code repository such as GitHub. Executables for this code should be shared as Docker containers on Docker Hub (https://hub.docker.com), keeping the code and analyses rerunnable in a fixed environment. This foundation enables AI/ML workflows (e.g., active learning, Bayesian optimization, mechanistic ML) that require clean,

https://doi.org/10.1021/acssynbio.5c00813
ACS Synth. Biol. 2026, 15, 1−8

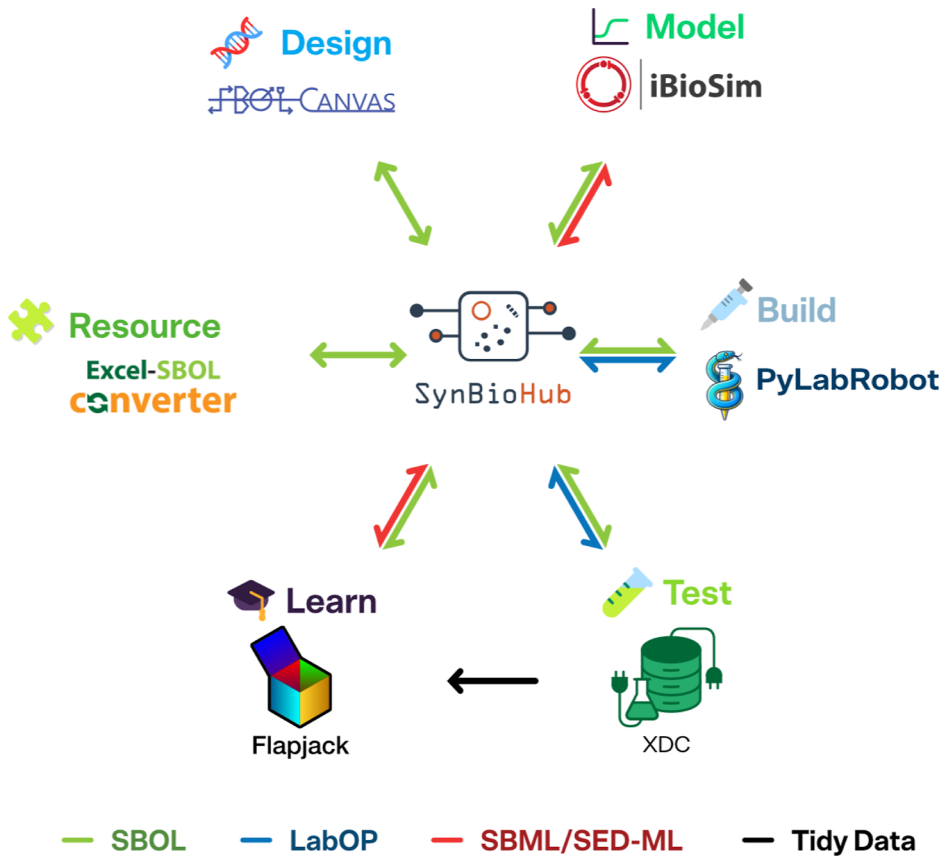

Figure 2. Software support for the proposed data management workflow. Resources can be selected by searching for them in a repository, such as SynBioHub. If the required resources are not found, they can be added to SynBioHub using the Excel-SBOL Converter. The selected resources can then be fetched from SynBioHub using a design tool, such as SBOLCanvas. Once the design is complete, it can be modeled using analysis tools, such as iBioSim, that support SBOL, SBML, and SED-ML. Once the modeling step indicates that the design is correct, assembly plans and assembly protocols can be developed using a build tool, such as PyLabRobot. After the genetic design is built, it can be tested in the laboratory, the metadata and protocols can be stored in SynBioHub, and the measurement data can be stored in Flapjack using a test tool, such as the XDC. Finally, the test data can be analyzed using a learn tool, such as Flapjack. The information learned can then be used to guide subsequent design and modeling steps.

standardized inputs and reproducible training/evaluation; standardized metadata make model comparisons auditable.[20]

2.8. Software Support for the Proposed Data Management Workflow

The use of standards is not only useful to share data with others but also useful to connect software tools used by each stage of the data management workflow. Figure 2 depicts one potential set of software tools that could be used. The software tools presented here were selected because they support the standards that we propose and should not be seen as an exclusive list. They simply illustrate how, by using these standards, a seamless data management workflow can be constructed. If other tools support the standards being employed here, then these tools can be easily integrated into this workflow.

The first stage of the workflow is to select the resources (parts, chassis, media, etc.) needed for a given genetic design from repositories, such as SynBioHub.[29] If the required resources are not already found in SynBioHub, they can be easily uploaded using the Excel-SBOL Converter tool.[40] This converter accepts information about resources entered into an Excel template. These templates are extensible, so users can add any information that is important to the resources they are using. The converter translates the data into SBOL using rules specified in the template, so these data can be uploaded to SynBioHub for use in other stages of the workflow.

The design and model stages are facilitated by *genetic design automation* (GDA) tools,[41] such as Cello,[42] iBioSim,[43] LOICA,[44] SBOLCanvas,[45] and SynBioSuite.[46] In Figure 2, the cloud-based SBOLCanvas tool is suggested to allow users to design the layout of their genetic circuit by drag-and-drop, visualize their designs using SBOL Visual, and import relevant part information encoded in SBOL from SynBioHub collections. The SBOL for the resulting genetic circuit design can be automatically converted into an SBML model for simulation.[47,48] These simulations can then be performed using any SBML-compliant simulator such as iBioSim.

Currently, there is no complete solution for the Build stage that integrates all of the proposed standards. We are currently developing one that converts abstract designs to concrete build plans that can be executed on various robotic platforms. One promising solution for this last step is PyLabRobot,[49] a hardware agnostic Python interface with vendor back-ends that translate procedures into the instrument-specific command set, generate the executable protocol, and validate that the requested actions, volumes, lab-ware, and deck layouts are supported. These protocols are easy to create and can be customized to diverse laboratory requirements, which lowers the entry barrier and allows researchers to adapt them to their

https://doi.org/10.1021/acssynbio.5c00813
ACS Synth. Biol. 2026, 15, 1−8

needs. It would not be too difficult to add the standard support (SBOL/LabOP) to allow full integration into this workflow. Even if an ad hoc, nonstandard process is used to build a plasmid sequence, the SeqImprove[50] tool can be used to annotate the sequence with parts found in SynBioHub part libraries and convert the final build result back into SBOL.

The test stage can be supported by the Experimental Data Connector (XDC)[51] software tool. The inputs to this tool are Excel workbook templates that enable the capture of the experimental metadata and measurements. XDC transforms the experimental metadata into the SBOL standard and uploads them to SynBioHub. The measurement data is encoded in the Tidy Data format and uploaded to the Flapjack data repository[30] for further processing and analysis.

During the learn stage, the measurement data can be examined using the software tool Flapjack.[30] By utilizing the Flapjack platform, researchers can effectively integrate the test phase with the build and learn phases, facilitating the characterization and optimization of genetic circuits. Furthermore, Flapjack's API provides access to these data through code and provides a Python package to access it and use other analysis software.

At the conclusion of each stage, the gathered data are uploaded to SynBioHub. As a result, at the end of each cycle, a comprehensive collection of data is accumulated including metadata, designs, mathematical models, simulation results, and experimental measurements. This facilitates an integrated curation process that achieves traceability, reproducibility, and seamless sharing, fostering adherence to FAIR principles.

3. CRITICAL NEXT STEPS

Interviews highlighted the need for greater awareness of the practical benefits of standardization for data management among synthetic biology laboratories.[52] While researchers acknowledge the advantages that standards offer in terms of interoperability and reproducibility, they expressed concerns about the learning curve and time investment required for an effective implementation. We also identified a tendency toward ad hoc solutions, which suggests that any standardization effort should be accompanied by intuitive tools and programs adapted to the needs of each laboratory. We acknowledge that although any arbitrary files can be attached to SBOL objects in SynBioHub, dedicated tooling to handle them must be developed for a streamlined workflow. The link between the metadata in SynBioHub and the experimental databases is modular by design, allowing an easy connection to databases tailored for flow cytometry, genomic, image, and other data types. These findings emphasize the importance of a gradual strategy that combines consolidated standards with operational flexibility to guarantee an efficient transition to more structured and collaborative data structures. To achieve these goals and improve data management practices, we have the following recommendations.

3.1. Encourage the Use of Standard Data Formats and Software That Support Them

The use of standards is critical for an effective data management plan. They provide a consistent framework for collecting, storing, processing, and sharing data, which brings numerous benefits to research laboratories. These benefits include improved data quality, enforced completeness of data fields, and data integration by enabling interoperability between different users and groups to share and use data seamlessly. Standard formats, such as SBOL, SBOL Visual, LabOP, SBML, SED-ML, and Tidy Data, should be promoted to ensure data compatibility and interoperability between different laboratories and tools. Software tools compatible with these standards should be encouraged to facilitate their use and consistent encoding.

3.2. Establish a Centralized Data Repository

A centralized data repository provides a single source of truth. This ensures that everyone in the research group or laboratory has access to the same information, reducing inconsistencies and duplication efforts. It also facilitates the implementation of data governance frameworks, which include defining roles, responsibilities, and processes for managing data. Platforms like SynBioHub[29] and Flapjack[30] are examples of data repositories for metadata and experimental data, respectively. These software tools can create a unified location for storing, organizing, and sharing data within and between laboratories, promoting easy access and collaboration. Ideally, these repositories should be staffed with professional data curators to ensure that high-quality data is shared.

3.3. Provide Training on Data Management and Software Tools

Training and software tools are essential enablers of data management best practices. Training also ensures that data standards and the software used to use them are understood, adopted, and consistently applied. Training sessions and resources should be provided for the members of the research group to improve their data management, sharing, and analysis skills. This will promote the adoption of best practices and ensure that researchers have the necessary skills to effectively manage research data.

3.4. Ensure Secure Data Storage and Implement Safeguards

As synthetic biology projects increase in complexity and biological significance, particularly when working with *genetically modified organisms* (GMOs), it is essential that data repositories provide robust security measures. Secure storage fosters trust among researchers, encouraging them to deposit and share their data on centralized platforms. Repositories must implement access controls, encryption, and regular audits to protect sensitive information. Additionally, clear protocols for responding to data breaches are necessary to minimize risks and ensure rapid recovery. As the volume and importance of stored data increase, proactive management of security and contingency planning becomes critical to maintaining the integrity and reliability of the data management workflow.

We note that European biofoundries and other non-US consortia have emphasized shared infrastructure and coordinated data management frameworks, which can accelerate standards adoption. Therefore, we recommend that readers in these groups adapt the workflow by (i) aligning it with institutional policies, (ii) hosting private SynBioHub and Flapjack instances to meet GDPR-like requirements, and (iii) participating in biofoundry alliances to harmonize interfaces.

We believe that if these recommendations are adopted, then synthetic biology data management workflows will produce FAIR data, enhance reproducibility, and advance the field.

https://doi.org/10.1021/acssynbio.5c00813
ACS Synth. Biol. 2026, 15, 1−8

■ ASSOCIATED CONTENT

Data Availability Statement

This section provides links to primary documentation for the standards and software tools used in the data management workflow, along with worked examples that demonstrate implementation and typical usage. SBOL and SBOL Visual (https://sbolstandard.org). LabOP (https://github.com/Bioprotocols/LabOP-specification). SBML (https://sbml.org). SED-ML (https://sed-ml.org). Excel-to-SBOL converter (https://github.com/SynBioDex/Excel-to-SBOL). SBOLCanvas (https://github.com/SynBioDex/SBOLCanvas). iBioSim (https://github.com/MyersResearchGroup/iBioSim). PyLabRobot (https://github.com/PyLabRobot/pylabrobot). XDC (https://github.com/SynBioDex/Xperimental-Data-Connector). Flapjack (https://flapjacksynbio.github.io/flapjack_api). SynBioHub (https://wiki.synbiohub.org). Examples (https://github.com/MyersResearchGroup/SynBioWorkflowExamples)

■ AUTHOR INFORMATION

Corresponding Author

Chris J. Myers − *University of Colorado Boulder, Boulder, Colorado 80309, United States;* orcid.org/0000-0002-8762-8444; Email: chris.myers@colorado.edu

Authors

Carolus Vitalis − *University of Colorado Boulder, Boulder, Colorado 80309, United States;* orcid.org/0000-0003-3867-0395

Gonzalo Vidal − *University of Colorado Boulder, Boulder, Colorado 80309, United States;* orcid.org/0000-0003-3543-520X

Sai P. Samineni − *University of Colorado Boulder, Boulder, Colorado 80309, United States*

Pedro Fontanarrosa − *University College London, London, Greater London WC1E 6BT, U.K.;* orcid.org/0000-0002-0535-2684

Complete contact information is available at: https://pubs.acs.org/10.1021/acssynbio.5c00813

Author Contributions

[§]C.V. and G.V. contributed equally to this work. C.V., P.F., S.P.S., and G.V. performed the interviews and curated the data. C.V. and G.V. led preparation of the original draft. C.M. and G.V. supervised this project. All authors contributed to the writing of this manuscript.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank the collaborators from the Army Center for Synthetic Biology project for participating in the interviews. We would also like to thank the attendees of our tutorial workshops at the Synthetic Biology, Engineering, Evolution and Design (SEED) conference over the years for their extremely valuable feedback. This work was supported by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-22-2-0210. This work was also supported by DARPA grant number HR0011-24-C-0423. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

■ REFERENCES

(1) Elowitz, M. B.; Leibler, S. A synthetic oscillatory network of transcriptional regulators. *Nature* **2000**, *403*, 335−338.

(2) Gardner, T. S.; Cantor, C. R.; Collins, J. J. Construction of a genetic toggle switch in Escherichia coli. *Nature* **2000**, *403*, 339−342.

(3) Endy, D. Foundations for engineering biology. *Nature* **2005**, *438*, 449−453.

(4) Peccoud, J.; Anderson, J. C.; Chandran, D.; Densmore, D.; Galdzicki, M.; Lux, M. W.; Rodriguez, C. A.; Stan, G.-B.; Sauro, H. M. Essential information for synthetic DNA sequences. *Nat. Biotechnol.* **2011**, *29*, 22.

(5) Mante, J.; et al. Synthetic Biology Knowledge System. *ACS Synth. Biol.* **2021**, *10*, 2276−2285.

(6) Goshisht, M. K. Machine Learning and Deep Learning in Synthetic Biology: Key Architectures, Applications, and Challenges. *ACS Omega* **2024**, *9*, 9921−9945.

(7) Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J. W.; da Silva Santos, L. B.; Bourne, P. E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3* (1), 160018.

(8) Thuronyi, B. W.; Debenedictis, E.; Barrick, J. E. No assembly required: Time for stronger, simpler publishing standards for DNA sequences. *PLoS Biol.* **2023**, *21*, No. e3002376.

(9) Shaikh, B.; Smith, L. P.; Vasilescu, D.; Marupilla, G.; Wilson, M.; Agmon, E.; Agnew, H.; Andrews, S. S.; Anwar, A.; Beber, M. E.; et al. BioSimulators: a central registry of simulation engines and services for recommending specific tools. *Nucleic Acids Res.* **2022**, *50*, W108−W114.

(10) Alloui, T.; Bousseboughi, I.; Chaoui, A.; Nouar, A. Z.; Chettah, M. C. U. *A Meta Search Engine based on a new result merging strategy*. In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, 2015; pp 531−536.

(11) Bankevich, A.; et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *J. Comput. Biol.* **2012**, *19* (5), 455−477.

(12) Langmead, B.; Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **2012**, *9*, 357−359.

(13) Blin, K.; Shaw, S.; Augustijn, H. E.; Reitz, Z. L.; Biermann, F.; Alanjary, M.; Fetter, A.; Terlouw, B. R.; Metcalf, W. W.; Helfrich, E. J. N.; van Wezel, G. P.; Medema, M. H.; Weber, T. antiSMASH 7.0: new and improved predictions for detection, regulation, chemical structures and visualisation. *Nucleic Acids Res.* **2023**, *51*, W46−W50.

(14) Kearse, M. D.; Moir, R.; Wilson, A.; Stones-Havas, S.; Cheung, M.; Sturrock, S. S.; Buxton, S.; Cooper, A.; Markowitz, S.; Duran, C.; Thierer, T.; Ashton, B.; Meintjes, P. L.; Drummond, A. J. Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics* **2012**, *28*, 1647−1649.

(15) Sayers, E. W.; et al. Database resources of the national center for biotechnology information. *Nucleic Acids Res.* **2022**, *50*, D20−D26.

(16) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The protein data bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.

(17) Huang, Y.; Sheth, R. U.; Zhao, S.; Cohen, L.; Dabaghi, K.; Moody, T. U.; Sun, Y.; Ricaurte, D.; Richardson, M.; Velez-Cortes, F.; Blazejewski, T.; Kaufman, T. C.; Ronda, C.; Wang, H. H. High-throughput microbial culturomics using automation and machine learning. *Nat. Biotechnol.* **2023**, *41*, 1424−1433.

(18) Robert, V.; et al. MycoBank gearing up for new horizons. *IMA fungus* **2013**, *4*, 371−379.

(19) Mante, J.; Myers, C. J. Advancing Reuse of Genetic Parts: Progress and Remaining Challenges. *Nat. Commun.* **2023**, *14*, 2953.
(20) Palacios, S.; Collins, J. J.; Del Vecchio, D. Machine learning for synthetic gene circuit engineering. *Curr. Opin. Biotechnol.* **2025**, *92*, 103263.
(21) Madsen, C.; et al. Synthetic Biology Open Language (SBOL) Version 2.3. *J. Integr. Bioinform.* **2019**, *16*, 20190025.
(22) Buecherl, L.; Mitchell, T.; Scott-Brown, J.; Vaidyanathan, P.; Vidal, G.; Baig, H.; Bartley, B.; Beal, J.; Crowther, M.; Fontanarrosa, P.; et al. Synthetic Biology Open Language (SBOL) Version 3.1.0. *J. Integr. Bioinform.* **2023**, *20*, 20220058.
(23) Brown, B.; Bartley, B.; Beal, J.; Bird, J. E.; Goñi-Moreno, Á.; McLaughlin, J. A.; Mısırlı, G.; Roehner, N.; Skelton, D. J.; Poh, C. L.; et al. Capturing Multicellular System Designs Using Synthetic Biology Open Language (SBOL). *ACS Synth. Biol.* **2020**, *9*, 2410−2417.
(24) Quinn, J. Y.; et al. SBOL visual: a graphical language for genetic designs. *PLoS Biol.* **2015**, *13*, No. e1002310.
(25) Keating, S. M.; Waltemath, D.; König, M.; Zhang, F.; Dräger, A.; Chaouiya, C.; Bergmann, F. T.; Finney, A.; Gillespie, C. S.; Helikar, T.; et al. SBML Level 3: an extensible format for the exchange and reuse of biological models. *Mol. Syst. Biol.* **2020**, *16*, No. e9110.
(26) Smith, L.; Bergmann, F.; Garny, A.; Helikar, T.; Karr, J.; Nickerson, D.; Sauro, H.; König, M. Simulation Experiment Description Markup Language (SED-ML): Level 1 Version 4. *J. Integr Bioinform* **2021**, *18*, 20210021.
(27) Bartley, B.; Beal, J.; Rogers, M.; Bryce, D.; Goldman, R. P.; Keller, B.; Lee, P.; Biggers, V.; Nowak, J.; Weston, M. Building an Open Representation for Biological Protocols. *ACM J. Emerg. Technol. Comput. Syst.* **2023**, *19*, 1−21.
(28) Wickham, H. Tidy data. *Journal of statistical software* **2014**, *59*, 1−23.
(29) McLaughlin, J. A.; Myers, C. J.; Zundel, Z.; Mısırlı, G.; Zhang, M.; Ofiteru, I. D.; Goñi-Moreno, A.; Wipat, A. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synth. Biol.* **2018**, *7*, 682−688.
(30) Yáñez Feliú, G.; Earle Gómez, B.; Codoceo Berrocal, V.; Muñoz Silva, M.; Nuñez, I. N.; Matute, T. F.; Arce Medina, A.; Vidal, G.; Vitalis, C.; Dahlin, J.; Federici, F.; Rudge, T. J. Flapjack: Data Management and Analysis for Genetic Circuit Characterization. *ACS Synth. Biol.* **2021**, *10*, 183−191.
(31) Myers, C. J. *Engineering Genetic Circuits*; CRC Press, 2016.
(32) Waltemath, D.; Adams, R.; Bergmann, F. T.; Hucka, M.; Kolpakov, F.; Miller, A. K.; Moraru, I. I.; Nickerson, D.; Sahle, S.; Snoep, J. L.; Le Novere, N. Reproducible computational biology experiments with SED-ML - The Simulation Experiment Description Markup Language. *BMC Syst. Biol.* **2011**, *5*, 198.
(33) Beal, J.; Selvarajah, V.; Chambonnier, G.; Haddock, T.; Vignoni, A.; Vidal, G.; Roehner, N. Standardized Representation of Parts and Assembly for Build Planning. *ACS Synth. Biol.* **2023**, *12*, 3646−3655.
(34) Johnson, D.; Batista, D.; Cochrane, K.; Davey, R. P.; Etuk, A.; Gonzalez-Beltran, A.; Haug, K.; Izzo, M.; Larralde, M.; Lawson, T. N.; et al. ISA API: An open platform for interoperable life science experimental metadata. *GigaScience* **2021**, *10*, giab060.
(35) Ronen, M.; Rosenberg, R.; Shraiman, B. I.; Alon, U. Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 10555−10560.
(36) Aïchaoui, L.; Jules, M.; Le Chat, L.; Aymerich, S.; Fromion, V.; Goelzer, A. BasyLiCA: a tool for automatic processing of a Bacterial Live Cell Array. *Bioinformatics* **2012**, *28*, 2705−2706.
(37) De Jong, H.; Ranquet, C.; Ropers, D.; Pinel, C.; Geiselmann, J. Experimental and computational validation of models of fluorescent and luminescent reporter genes in bacteria. *BMC Syst. Biol.* **2010**, *4*, 55.
(38) Zulkower, V.; Page, M.; Ropers, D.; Geiselmann, J.; De Jong, H. Robust reconstruction of gene expression profiles from reporter gene data using linear inversion. *Bioinformatics* **2015**, *31*, i71−i79.
(39) Vidal, G.; Vitalis, C.; Muñoz Silva, M.; Castillo-Passi, C.; Yáñez Feliú, G.; Federici, F.; Rudge, T. J. Accurate characterization of dynamic microbial gene expression and growth rate profiles. *Synth. Biol.* **2022**, *7*, ysac020.
(40) Mante, J.; Abam, J.; Samineni, S. P.; Potzsch, I. M.; Beal, J.; Myers, C. J. Excel-SBOL Converter: Creating SBOL from Excel Templates and Vice Versa. *ACS Synth. Biol.* **2023**, *12*, 340−346.
(41) Myers, C. J.; Barker, N.; Kuwahara, H.; Jones, K.; Madsen, C.; Nguyen, N.-P. D. *Proceedings of the 2009 International Conference on Computer-Aided Design*; Genetic Design Automation: New York, NY, USA, 2009; pp 713−716.
(42) Jones, T. S.; Oliveira, S. M. D.; Myers, C. J.; Voigt, C. A.; Densmore, D. Genetic Circuit Design Automation with Cello 2.0. *Nat. Protoc.* **2022**, *17*, 1097−1113.
(43) Watanabe, L.; Nguyen, T.; Zhang, M.; Zundel, Z.; Zhang, Z.; Madsen, C.; Roehner, N.; Myers, C. IBIOSIM 3: A Tool for Model-Based Genetic Circuit Design. *ACS Synth. Biol.* **2019**, *8*, 1560−1563.
(44) Vidal, G.; Vitalis, C.; Rudge, T. J. L. O. I. C. A. LOICA: Integrating Models with Data for Genetic Network Design Automation. *ACS Synth. Biol.* **2022**, *11*, 1984−1990.
(45) Terry, L.; Earl, J.; Thayer, S.; Bridge, S.; Myers, C. J. SBOLCanvas: a visual editor for genetic designs. *ACS Synth. Biol.* **2021**, *10*, 1792−1796.
(46) Sents, Z.; Stoughton, T. E.; Buecherl, L.; Thomas, P. J.; Fontanarrosa, P.; Myers, C. J. SynBioSuite: A Tool for Improving the Workflow for Genetic Design and Modeling. *ACS Synth. Biol.* **2023**, *12*, 892−897.
(47) Misirli, G.; Nguyen, T.; McLaughlin, J. A.; Vaidyanathan, P.; Jones, T. S.; Densmore, D.; Myers, C.; Wipat, A. A computational workflow for the automated generation of models of genetic designs. *ACS Synth. Biol.* **2019**, *8*, 1548−1559.
(48) Zilberzwige-Tal, S.; Fontanarrosa, P.; Bychenko, D.; Dorfan, Y.; Gazit, E.; Myers, C. J. Investigating and modeling the factors that affect genetic circuit performance. *ACS Synth. Biol.* **2023**, *12*, 3189−3204.
(49) Wierenga, R. P.; Golas, S. M.; Ho, W.; Coley, C. W.; Esvelt, K. M. PyLabRobot: An open-source, hardware-agnostic interface for liquid-handling robots and accessories. *Device* **2023**, *1*, 100111.
(50) Mante, J.; Sents, Z.; Britt, D.; Mo, W.; Liao, C.; Greer, R.; Myers, C. J. SeqImprove: Machine-Learning-Assisted Curation of Genetic Circuit Sequence Information. *ACS Synth. Biol.* **2024**, *13*, 3051−3055.
(51) Samineni, S. P.; Vidal, G.; Vitalis, C.; Feliú, G. Y.; Rudge, T. J.; Myers, C. J.; Mante, J. Experimental Data Connector (XDC): Integrating the Capture of Experimental Data and Metadata Using Standard Formats and Digital Repositories. *ACS Synth. Biol.* **2023**, *12*, 1364−1370.
(52) Berezin, C.-T.; Aguilera, L. U.; Billerbeck, S.; Bourne, P. E.; Densmore, D.; Freemont, P.; Gorochowski, T. E.; Hernandez, S. I.; Hillson, N. J.; King, C. R.; et al. Ten simple rules for managing laboratory information. *PLoS Comput. Biol.* **2023**, *19*, No. e1011652.

https://doi.org/10.1021/acssynbio.5c00813
ACS Synth. Biol. 2026, 15, 1−8