

## Perspective

# Best Practices for Making Reproducible Biochemical Models

Veronica L. Porubsky,<sup>1,5,\*</sup> Arthur P. Goldberg,<sup>2,4,5,\*</sup> Anand K. Rampadarath,<sup>3</sup> David P. Nickerson,<sup>3</sup> Jonathan R. Karr,<sup>2,4</sup> and Herbert M. Sauro<sup>1</sup>

<sup>1</sup>Department of Bioengineering, University of Washington, Seattle, WA 98105, USA

<sup>2</sup>Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA

<sup>3</sup>Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

<sup>4</sup>Icahn Institute for Data Science and Genomic Technology, Icahn School of Medicine at Mount Sinai, New York, NY 10029, USA

<sup>5</sup>These authors contributed equally

\*Correspondence: [verosky@uw.edu](mailto:verosky@uw.edu) (V.L.P.), [arthur.goldberg@mssm.edu](mailto:arthur.goldberg@mssm.edu) (A.P.G.)

<https://doi.org/10.1016/j.cels.2020.06.012>

## SUMMARY

Like many scientific disciplines, dynamical biochemical modeling is hindered by irreproducible results. This limits the utility of biochemical models by making them difficult to understand, trust, or reuse. We comprehensively list the best practices that biochemical modelers should follow to build reproducible biochemical model artifacts—all data, model descriptions, and custom software used by the model—that can be understood and reused. The best practices provide advice for all steps of a typical biochemical modeling workflow in which a modeler collects data; constructs, trains, simulates, and validates the model; uses the predictions of a model to advance knowledge; and publicly shares the model artifacts. The best practices emphasize the benefits obtained by using standard tools and formats and provides guidance to modelers who do not or cannot use standards in some stages of their modeling workflow. Adoption of these best practices will enhance the ability of researchers to reproduce, understand, and reuse biochemical models.

## INTRODUCTION

Recent recognition of the reproducibility obstacles in scientific research has led to calls for improved practices that ensure that published results can be reproduced by independent investigators (Mobley et al., 2013; Prinz et al., 2011; Golub et al., 1999; De Schutter, 2010; Woelfle et al., 2011; Casadevall and Fang, 2010). Computational models of biochemical system dynamics face the same criticism (Elofsson et al., 2019; Sandve et al., 2013; Peng, 2011; Medley et al., 2016; Waltemath and Wolkenhauer, 2016). Reproducible models confer important benefits: they are easier to understand, trust, modify, reuse, and compose. Thus, they facilitate collaboration among biochemical modelers. A collection of reproducible models could be reused to construct multi-scale models of larger, more complex systems. Achieving a dynamical, biochemical model with these traits requires that (1) the data, code, and decisions used to construct and simulate models be recorded by the modeler, (2) models be described in comprehensible languages, standard data formats, and nomenclature, and (3) the artifact produced by modeling be publicly shared and governed by open-source licenses (Rosen, 2005). To make it easier to conduct reproducible biochemical modeling, we and others are creating tools (Choi et al., 2018; Somogyi et al., 2015; Smith et al., 2009; Choi et al., 2016; Hucka et al., 2003; Hoops et al., 2006; Waltemath et al., 2011b; Olivier and Snoep, 2004; Watanabe et al., 2019) that simplify these activities. As a practical guide for the computational biochemical modeling community, this

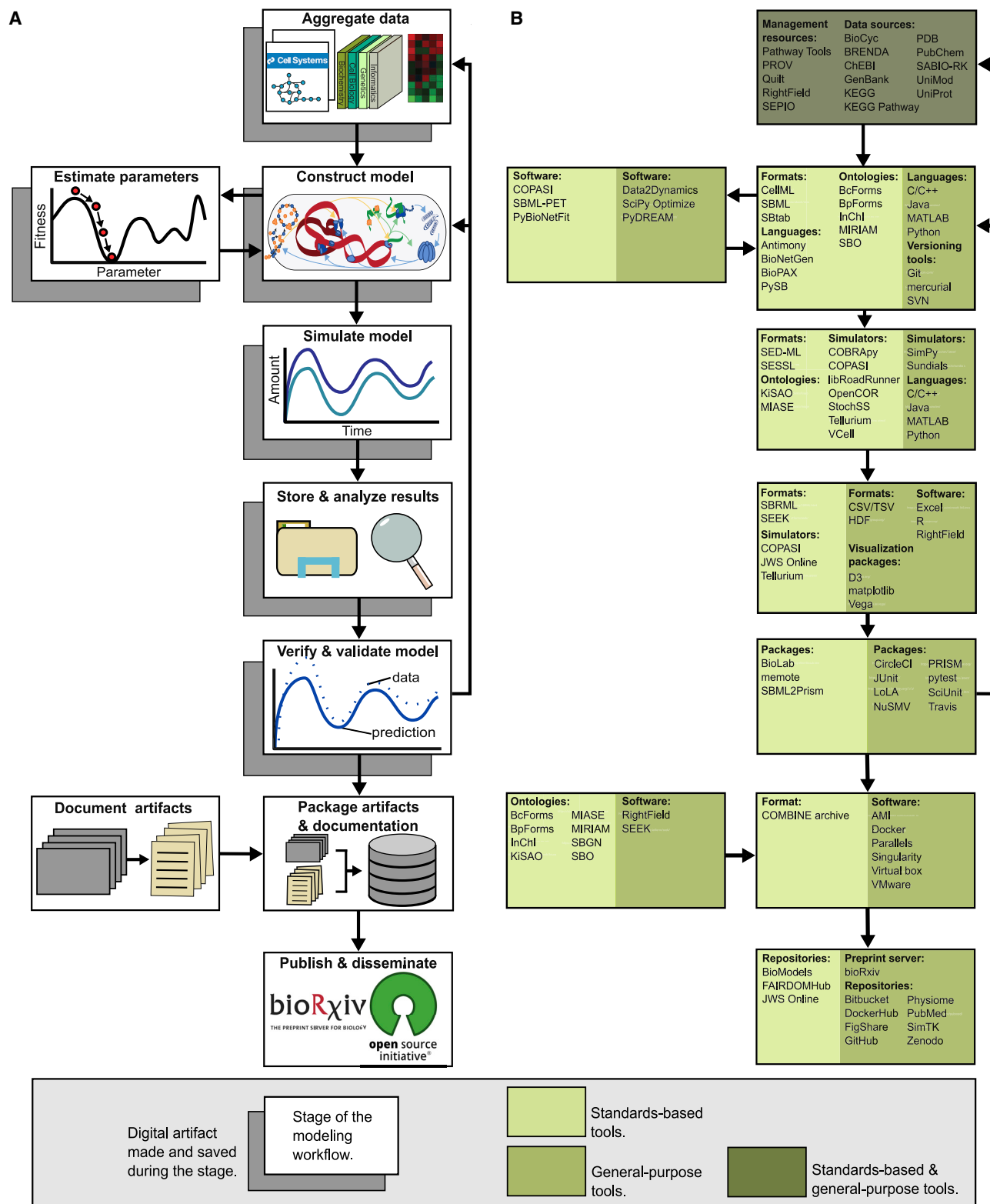
article lists nine of the most important best practices that researchers can use to make their models more reproducible.

We offer modelers these best practices as a guide for conducting reproducible modeling. We structure the best practices as advice for each stage of a typical biochemical modeling workflow: collect and aggregate data; construct a model, identify and estimate its parameters, define initial conditions and simulate the model, analyze the simulation results, validate the model, document all of the model artifacts, build a package that contains the artifacts and their documentation, and share the package while publishing the findings of the study (see Figure 1). This structure makes it easy to selectively implement a subset of the best practices.

An initiative with consistent goals developed the findability, accessibility, interoperability, and reusability (FAIR) principles (Wilkinson et al., 2016), which set forth goals and desiderata for good management and stewardship of scholarly data. To better support knowledge discovery and innovation, the FAIR principles urge all scholars who create digital data to ensure that it is findable, accessible, interoperable, and reusable. The best practices we present consistently support the FAIR principles, as enumerated in Table 1. In addition, our goals for achieving reproducible biochemical models focus more on creating reusable model artifacts, and our detailed practices provide specific guidelines that go beyond the scope of the FAIR principles.

We organize the best practices into two parallel sets of recommendations. The first provides guidance to biochemical modelers who employ a “standards-based approach,” which uses tools and data formats that were designed for biochemical modeling and





**Figure 1. Practical Recommendations of Tools for Reproducible Modeling across All Stages of the Typical Biochemical Modeling Workflow** (A) A typical workflow that creates and uses a dynamical model: in “aggregate data,” a modeler collects data from papers, public data sources and/or private experiments; in “construct model,” they use the data, their biological knowledge, assumptions, and modeling methods to create a model; in “estimate parameters,” the modeler produces a complete and self-consistent set of input parameters from the data; in “simulate model,” the modeler integrates the model over time; in “store and analyze results,” they store simulation results and analyze them; in “verify & validate model,” the modeler ensures that the model and its predictions are

(legend continued on next page)

have been adopted as community standards. The second advises modelers who use a “general-purpose approach” that employs computer languages, tools, and data formats that were designed to be used by many fields. While modelers who employ the general-purpose approach can make reproducible models, the standards-based tools and data formats expedite the construction of reproducible models. By facilitating the exchange of model artifacts between platforms for construction, simulation, analysis, and validation, and employing consistent ontology and minimum information standards, standards-based modeling makes it easier for other researchers to understand and reuse these models. Therefore, we recommend that the standards-based approach be followed whenever possible. However, modelers constructing models that require functionality that is not supported by the existing standards-based tools (Karr et al., 2012; Goldberg et al. 2018) will find the general-purpose approach to be more practical for some stages of the workflow.

Typical biochemical modeling workflow (A) and tools for achieving best practices for making reproducible biochemical models (B). Full caption on the following page.

## BEST PRACTICES FOR MAKING REPRODUCIBLE BIOCHEMICAL MODELS

### Best Practice 1: When Aggregating and Curating Data, Retain Its Metadata and Provenance

Most biochemical models require inputs gathered by data aggregation or the collection of data from multiple experiments, scientific papers, and online data sources. Appendix A of Goldberg et al. (Goldberg et al., 2018) provides an extensive list of data sources that store intracellular biochemical data. If experiments are conducted to obtain new data or inform conditions studied by the model, use reproducible experimental methods. We encourage experimentalists to use appropriate ontologies and minimum information standards when recording experimental methods, conditions, and the historical record of the data (Bandrowski et al., 2016; Kazic, 2015; Orchard et al., 2007; Deutsch et al., 2008; Bus-tin et al., 2009; Brazma et al., 2001; Taylor et al., 2007). Provide thorough descriptions of statistical analyses and estimated uncertainties in measurements that are due to instrument accuracy and other sources of noise (White, 2008; Miškovic and Hatzimanikatis, 2011). Data curation standardizes, normalizes, and links together the aggregated data to facilitate its use in models and manages its metadata (Goldberg et al., 2018). Metadata are data that describe data, in this case, data used by biochemical models (Deelman et al., 2010). Metadata about a measurement should include its units, estimates of its accuracy, annotations, and the identities of the ontologies that define the annotations. This will provide information that a modeler can use to evaluate whether data are suitable for their model. Data provenance is metadata that describes the historical record of data and should include

the lab that generated the data, the conditions under which it was obtained, the protocol used to make it, the paper that reported the measurement, and the online data source from which it was aggregated (Deelman et al., 2010). Provenance records should also describe transformations of the data following their collection. Follow the scientific evidence and provenance information OntologySEPIO to provide rich, computable representations of the evidence and provenance behind scientific assertions (Brush et al., 2016).

A limited set of existing tools can track the versions, metadata, and provenance of aggregated data. For example, Quilt provides version control features for data, and PROV (Moreau et al., 2015) is an extensive data model for tracking the provenance of online data, such as data sources. Over fifty tools have implemented parts of PROV (Huynh et al., 2013). When aggregating data, modelers should automate the processes that retain its metadata and provenance. If existing tools cannot perform the automation, then custom tools must be developed.

### Best Practice 2: Record the Model Construction Process

Construction (of a model) encodes the structure and dynamics of the biological system being modeled: its geometry; molecular species that participate in the system, the reactions that transform them, and the rate laws for these reactions, initial conditions, and parameters used by these model components. Other biological or biochemical features may also be represented. Document the construction process to ensure that the justification for design decisions, which are not explicitly encoded in the logic or mathematics of the model, is communicated to independent researchers. This includes simplifications and assumptions about the system and environmental context and decisions about which measurements to use.

Many of the artifacts created during model construction will change as the data and models are improved and altered. We recommend that modelers use version control systems to track changes in their data and code. This would support concurrent development by teams of modelers and help avoid unnecessary duplication of artifacts. Version control system tools include sub-version (SVN) and Git. For cloud-based storage of SVN and Git repositories that support version control, use GitHub (Brindescu et al., 2014).

### Standards-Based

Follow the minimal information required in the annotation of biochemical models (MIRIAM) standard to ensure that all model components are explained (Laibe and Le Novère, 2007). Use the systems biology graphical notation (SBGN) (Laibe and Le Novère, 2007) to visualize the model to help independent groups understand its components and interactions.

### General-Purpose

Record all the data and software used to construct the model and document the construction process. Help independent

consistent with experimental data; in “document artifacts,” the modeler annotates and provides human-readable descriptions (tan rectangles) for all model artifacts from each stage; in “package artifacts and documentation,” they combine all model artifacts and documentation into archive(s) to be shared publicly, and in “publish and disseminate,” the modeler publishes their novel scientific findings and shares the archive(s) by depositing them in open-source repositories that independent researchers can access to reproduce, understand, and reuse the model. Black arrows indicate the transitions between workflow stages.

(B) Software tools and data formats for reproducible modeling: Tools and data formats that enhance reproducibility are listed in a diagram that parallels the workflow in (A). These tools and data formats are split into recommendations for standards-based and general-purpose approaches to modeling, as presented in the text. Tools that are useful in multiple modeling stages are listed in those stages.

A table with links to the tools shown in Figure 1 is included in the Supplemental Information (see Table S1)."

**Table 1. Best Practices for Reproducibly Building and Simulating Biochemical Models Aligned with the Modeling Workflow Stages and the FAIR Principles They Implement**

Workflow Stage	Best Practices for Reproducible Biochemical Modeling	FAIR ID
Aggregate data	Best Practice 1: When aggregating and curating data, retain its metadata and provenance	F1–F4, A1 and A2, I1–I3, R1
Construct model	Best Practice 2: Record the model construction process Best Practice 3: Make model descriptions comprehensible by using structured formats and unambiguous names	F2, I1–I3, R1 F1, F4, A1, I1–I2, R1.3
Estimate parameters	Best Practice 4: If parameters are estimated, share the estimation algorithm and perform uncertainty quantification	A1
Simulate model	Best Practice 5: Record all simulation inputs and methods, including initial conditions, numerical integration algorithms, random number generator algorithms, and seed values	F2 and F3, A1, I1–I3
Store & analyze results	Best Practice 6: Save structured unprocessed simulation result and share the data presented in graphs and tables	A1, I1
Verify & validate model	Best Practice 7: Automate and document model verification and validation	A1, I1, R1.3
Package artifacts & documentation	Best Practice 8: Confirm that model predictions can be reproduced in an independent computing environment	A1, I1 and I2, R1.3
Publish & disseminate	Best Practice 9: Create packages that contain all model artifacts and documentation, and deposit them in public, version-controlled repositories	F1–F4, A1 and A2, I1–I3, R1

FAIR principle (Wilkinson et al., 2016) identifiers, which are most relevant to each best practice, are included to connect the practical concepts described in this article with policies adopted by the broader research community.

investigators understand the model by noting all assumptions and decisions made during construction, with comments in source code or as supplementary documentation for each artifact. When developing figures to visualize the model components and interactions, provide a detailed legend. Follow existing conventions for interaction maps when possible—for example, use standard arrowheads to represent mass transfer or activating interactions between biochemical species, and blunt-end arrowheads to represent repression.

### Best Practice 3: Make Model Descriptions Comprehensible by Using Structured Formats and Unambiguous Names

Models described in structured formats, which precisely identify model components, are easier to understand. All model components, such as a model geometry, species, and reactions, should be identified by an unambiguous name or annotation with a distinct semantic meaning. We also urge modelers to unambiguously describe the system context, that is, the biological entity being modeled, including its species, tissue, cell type or strain, and genotype. Also, describe the environmental context, such as the temperature, pressure, and external nutrients in the environment surrounding the biological entity. The physical units of all quantities represented in the model should be documented and propagated as quantities are transformed. Software packages that support units are available in multiple languages, including R and Python (Pebesma et al., 2016; Grecco and Thielen, 2020).

#### Standards-Based

To facilitate design and comprehension of their models, modelers should use standard systems biology formats for model descriptions, such as the systems biology markup language (SBML) (Hucka et al., 2003) and CellML (Cuellar et al., 2003). Antimony is a modular, text-based language that can describe

a model in simple statements and export models to SBML (Smith et al., 2009). BioPAX is a modeling language that represents biological pathways and can export them to SBML or CellML (Demir et al., 2010). BioNetGen (Harris et al., 2016) and PySB (Lopez et al., 2013) enable rule-based models and can also export them to SBML. SBTAB (Lubitz et al., 2016) and ObjTables (Karr et al., 2020) provide a set of standardized syntax rules and conventions for table-based data formats, to help modelers structure experimental measurements and enable automated data integration and model building.

Use the systems biology ontology (SBO) (Courtot et al., 2011) to precisely record and categorize the semantics of model components, including assumptions, the types of rate laws, and the roles of species in reactions and rate laws. The structures of small molecules can be described using the International Union of Pure and Applied Chemistry (IUPAC) International Chemical Identifiers (InChI) (Heller et al., 2015). BpForms and BcForms can precisely describe the structures of and modifications to bio-polymers and complexes (Lang et al., 2020).

#### General-Purpose

Models can be described using general-purpose programming languages. Document the code thoroughly with comments that describe the structure of all model components. If possible, describe the components of models as data rather than in code. For example, the types of molecules in a model could be described in a computer-readable table that contains a column for each molecular attribute. Many other model components can be described in similar tables. Storing components in spreadsheets or delimited files and annotating the meaning of all components and fields will help independent investigators comprehend the model. Standard identifiers for biological and chemical species identified in the standards-based section above can be used to name model entities in the general-purpose approach.

Avoid publishing a model as a system of ordinary differential equations alone, because converting a reaction network to this representation usually loses information, which obfuscates the underlying biochemistry. Instead, publish both the ordinary differential equations and a description of the model as a set of reactions and provide a computer-readable representation whenever possible.

#### **Best Practice 4: If Parameters Are Estimated, Share the Estimation Algorithm and Perform Uncertainty Quantification**

Unfortunately, aggregated measurements often fail to provide a complete, self-consistent set of parameters for a biochemical model. Therefore, parameter estimation is typically needed to infer the values of missing or inconsistent parameters. Parameter estimation solves for parameter values that minimize the divergence between experimental measurements of the system being modeled and the predictions of the model for that data. For non-identifiable models, common when representing biological systems, there are multiple sets of parameters that can minimize this divergence. In these cases, families of estimated parameters should be reported in machine-readable formats, to adequately capture their correlation structure. Many parameter estimation algorithms use well-established optimization methods and allow the user to tune inputs for effective estimation (Ashyraliyev et al., 2009). Use reusable programs instead of manually tuning parameter values.

Recognizing that biochemical measurements are imprecise and many biochemical properties, such as species concentrations, vary naturally; uncertainty quantification estimates the distributions of model input parameters and then propagates these distributions through model simulations to quantify their impacts on model predictions. When possible, initialize simulations by sampling inputs from their estimated distributions and execute multiple simulations to estimate the distributions of predictions. Algorithms and codes for parameter estimation and uncertainty quantification should be included in shared artifacts.

#### **Standards-Based**

Use a reusable program, such as COPASI (Hoops et al., 2006), SBML-PET (Zi and Klipp, 2006), or PyBioNetFit (Mitra et al., 2019), to perform parameter estimation on SBML models. COPASI accomplishes this by minimizing the least squares error between time course measurements and predictions of the model or by performing profile likelihood estimation (Hoops et al., 2006). SBML-PET estimates parameters for diverse types of experimental measurements (Zi and Klipp, 2006). PyBioNetFit provides both parameterization and uncertainty quantification protocols (Mitra et al., 2019).

#### **General-Purpose**

State the parameter estimation algorithm and all input values used to tune the protocol. If a custom algorithm is created, provide its code and documentation. Potential tools include Data2Dynamics (Raue et al., 2015), PyDREAM (Shockley et al., 2018), and the optimization library provided by SciPy (Jones et al., 2001). Data2Dynamics is a MATLAB toolbox that addresses parameter estimation challenges (Raue et al., 2015). In Python, PyDREAM (Shockley et al., 2018) performs parameter estimation

and uncertainty quantification for biochemical models and the SciPy (Jones et al., 2001) optimization package provides many gradient-based and global optimization approaches.

#### **Best Practice 5: Record All Simulation Inputs and Methods, Including Initial Conditions, Numerical Integration Algorithms, Random Number Generator Algorithms, and Seed Values**

Simulation (of a model) involves computational execution of the mathematics describing a model to generate predictions of its dynamic behavior. We urge modelers to implement numerical methods—such as a custom integration method—separately from representations of biological systems, so that each of them can be independently reused. When performing stochastic simulations that use a pseudo-random number generator algorithm, preserve a precise definition of the algorithm. Execute an ensemble of simulation runs with different seeds to estimate the distributions of species population trajectories and predictions that depend on them. Make the ensembles large enough to accurately characterize properties of the distributions. Record the seeds used by these simulations or a reproducible method for obtaining the seeds. If multiple distinct sets of input parameters are analyzed, repeat the process of estimating the distribution of predictions for each parameter set.

#### **Standards-Based**

Follow the minimum information about & simulation experiment (MIASE) guidelines to determine which software and data to archive (Waltemath et al., 2011a). The simulation experiment description markup language (SED-ML) can encode simulation descriptions, including simulator settings and parameter modifications, and facilitate exchange between standard-compatible tools (Waltemath and Le Novère, 2013). We recommend using the kinetic simulation algorithm ontology (KiSAO) (Courtot et al., 2011) to annotate SED-ML documents. Many simulators are compatible with these standards, including COPASI and Java web simulation online (JWS Online), an online platform that hosts models, simulation programs, and data (Olivier and Snoep, 2004). libRoadRunner provides high-performance simulation of multiple numerical integration algorithms (Somogyi et al., 2015), and Tellurium provides a Pythonic interface to access libRoadRunner, SED-ML, and additional analysis capabilities (Choi et al., 2018). OpenCOR is a modeling environment, which can be used to simulate models described using CellML (Garny and Hunter, 2015). Alternatively, with simulation experiment specification via & scala layer (SESSL) modelers can specify simulation experiments in a domain-specific language, import SBML model descriptions, and write additional specifications in Scala (Ewald and Uhrmacher, 2014).

#### **General-Purpose**

To ensure that published results can be regenerated, archive all software and data used to produce simulation results that may be used or referenced in publications. Follow the generic MIASE (Waltemath et al., 2011a) guidelines regarding documentation of model descriptions, simulators, and simulation experiments.

A simulation experiment simulates one or more models. It inputs initial conditions and parameters, and, optionally, perturbations. Perturbations can modify parts of the model or its parameters. A modeler can make a simulation experiment reproducible without archiving multiple executables by writing a small



program, often called a “script,” that executes all simulations. If the simulator has an application programming interface (API), then a script that uses the API can be written to run all simulation experiments. Strive to store the initial conditions and parameters used by the script in data files that can be easily understood by independent investigators. Variations on this approach should be devised if multiple simulators are required, if some simulators do not have APIs, or if the simulators depend on incompatible computing environments. For example, if a simulator does not have an API, then script could be written to output a sequence of commands in another script that executes the simulation experiments that use the simulator.

### **Best Practice 6: Save Structured Unprocessed Simulation Results and Share the Data Presented in Graphs and Tables**

To allow independent researchers to analyze published simulation results and perform new analyses and mathematical manipulations of the results without requiring that they reproduce the entire model, unprocessed results should be preserved for dissemination. Unprocessed results of simulations that might be used in published findings should be temporarily saved; results that are used in published findings should be archived so they can be shared with independent investigators.

Share the reduced data that are presented in published graphs and tables to enable independent analyses by other investigators. If these data are not shared, researchers must devote substantial effort to transcribe data from figures. Results and data should be stored in structured and space-efficient formats with annotations that clearly describe the data. Unlike traditional figures, interactive graphics provide access to the data presented by mouse over. However, to comprehensively share plotted data, archive formatted files containing the raw data, the graphics files, the code that generated the graphics, and documentation that relates the data to the figures. Providing the source data and the code used to generate the published figures ensures that the figure can be readily regenerated by independent researchers or altered to improve understanding of the data (EMBOpress, 2019).

#### **Standards-Based**

While standardized formats, such as the systems biology results markup language (SBRML) (Dada et al., 2010), have been developed for simulation results, they have not been widely adopted, leaving opportunities to develop additional standards. The SEEK platform helps address the challenges of managing model data by providing a suite of standards-compliant tools that link data with relevant metadata, facilitate exchange with independent modelers, and enable web-based simulation and plotting of experimental data stored on the platform (Wolstencroft et al., 2015). JWS Online directly links simulation predictions to online plots that display them and allows modelers to execute real-time web-based simulation of stored models to visualize interactive output (Olivier and Snoep, 2004).

#### **General-Purpose**

Annotate the semantic meaning and provenance of all simulation results. Save results in computer-readable formats, such as comma-separated values (CSV) or tab-separated values (TSV). The hierarchical data format (HDF) offers structured and efficient data storage that is especially useful for large datasets (Brown

et al., 1993), and RightField provides semantic data annotation features in excel spreadsheets (Wolstencroft et al., 2011). Export interactive graphics using MATLAB figures or web-based frameworks, such as Vega and D3.

### **Best Practice 7: Automate and Document Model Verification and Validation**

Verification (of a model and its tools) and validation (of a model) are concerned with whether a model, its tools, and its predictions are consistent with experimental data (Sargent, 2010). We recommend that modelers automate verification and validation as much as possible. Employ workflows, shell scripts, and similar techniques to automate processes that involve repeated execution of programs with different inputs. Document the verification and validation processes, especially the steps that are not automated, such as decisions made and conclusions reached during verification and validation. Record the algorithms, code, and data used.

Models that use stochastic simulations must employ stochastic validation methods, which statistically compare the distributions of model predictions with the distributions of measurements of the biological system phenotypes.

#### **Standards-Based**

Memote validates static properties of metabolic flux-balance constraint models and complies with SBML (Lieven et al., 2018). SBML2Prism can be used to make SBML models compatible with the PRISM model checker, which provides probabilistic model checking utilities that automate quantitative performance analyses in stochastic biochemical models (Kwiatkowska et al., 2011). BioLab uses statistical model checking to verify that rule-based biochemical models programed in the BioNetGen language exhibit expected temporal properties (Clarke et al., 2008).

#### **General-Purpose**

Models written in general-purpose programming languages should be designed, built, and verified using software engineering techniques, such as object-oriented programming, modularity, unit testing, and regression testing. Continuous integration services, such as CircleCI and Travis, automate regression testing. Evaluate whether simulation functions and the computational model have been correctly designed and implemented. Defining invariants and ensuring that they are satisfied can help verify modeling tools (Gries, 2012). For example, ensuring that chemical reactions conserve mass and that species populations are non-negative can detect subtle errors.

SciUnit is a framework for test-driven model validation (Omar et al., 2014). Modelers write a set of scientific unit tests that compare the predictions of the model with measured phenotypes of the system being modeled, and SciUnit runs the tests. We recommend SciUnit for validating general-purpose and standards-based models. NuSMV (Cimatti et al., 2002) and LoLA (Schmidt, 2000) may be used for formal model checking.

### **Best Practice 8: Confirm That Model Predictions Can Be Reproduced in an Independent Computing Environment**

Because modeling tools, models, and computing environments are all complex software, it can be difficult to re-execute a simulation experiment in a computing environment that differs from the original environment used to execute the experiment. To increase the likelihood that independent researchers can replicate

a simulation experiment in a different environment, we recommend that modelers replicate their own experiments in a different environment before disseminating the artifacts. Some journals now test the functionality of submitted models and may award badges to manuscript that attach data, source code and model artifacts, or attach artifacts that pass reproducibility tests performed by the journal (Donoho, 2010; Kidwell et al., 2016; AJPS, 2016). The Center for Open Science has compiled a list of journals that provide badges for sharing data and other materials (COS, 2019).

Traditionally, a programmer prepares to execute complex software in their computing environment by installing the software upon which the complex software depends. We strongly advise modelers to automate this software installation process and test it and their simulation experiments in popular computing environments. Given the challenges of this approach, we suggest that modelers build and disseminate a container or virtual machine, which contains all of the software and data required by their model and simulation experiments. Independent researchers can deploy the container or virtual machine to replicate the simulation experiments it contains. Popular types of containers include Docker, Amazon machine image and singularity, while common virtual machines include VMware, Parallels, and VirtualBox. In addition, cloud services, such as the Amazon elastic compute cloud, can be rented to test the modeling experiments deployed in a container or virtual machine.

### Best Practice 9: Create Packages that Contain All Model Artifacts and Documentations, and Deposit them in Public, Version-Controlled Repositories

We recommend that all model artifacts be publicly shared to ensure that they are findable, accessible, and reusable, as emphasized by the FAIR (Wilkinson et al., 2016) principles. A summary of the aims, design, assumptions, limitations, and structure of the model will help independent investigators find, understand, and reuse it. The relationships between model artifacts should be clearly described so that others can follow the dependencies within the data, model, code, predictions, and findings.

When possible, minimize barriers to accessibility. We recommend that the packages of artifacts be governed by an open-source license (Rosen, 2005) and deposited in public, version-controlled repositories. GitHub, Bitbucket, and Zenodo are popular repositories, which use version control and are easily accessed by the scientific community. Ideally, the packages would be shared by publishing a URL or DOI reference to them. Modelers who use version-controlled repositories should label and record public releases by tagging the release versions. Share findings on preprint servers, such as bioRxiv (Sever et al., 2019), and publish in peer-reviewed, open-access journals. The Physiome Project has developed an open-access journal created for the explicit purpose of publishing reproducible models of biological systems (Hunter and Borg, 2003).

### Standards-Based

The computational modeling in biology network (COMBINE) developed the open modeling exchange format (OMEX), which enables modelers to store all project data required for model comprehension, construction, and simulation in a single zip archive (Bergmann et al., 2014). Deposit the archive in a

modeling repository, such as the BioModels Database (Li et al., 2010) or the FAIRDOMHub (Wolstencroft et al., 2017). JWS Online also provides an integrated and standards-compliant storage and simulation platform that stores model components with their assumptions, parameter values, simulation results, and raw data. Independent investigators can execute simulations of shared models on the JWS Online platform and download model artifacts.

### General-Purpose

We urge modelers following a general-purpose approach to structure input data, model definitions, and other artifacts by creating directories that group the artifacts within an archive containing the complete modeling project. Provide thorough documentation for the contents of the archive, including a manifest that lists all files, a metadata file that describes the contents of the archive, and documents on how to execute the simulations and investigate the model. Upload archived artifacts to open-access repositories for scientific research, such as FigShare, SimTK, or Zenodo (Singh 2011; Sherman et al., 2005; Sicilia et al., 2017).

## DISCUSSION

### A Practical Guide to Reproducible Biochemical Modeling

Building reproducible biochemical models is essential; reproduction of results by independent investigators is a tenet of science. Therefore, we have comprehensively enumerated the best practices that researchers who build dynamical models of biochemical systems should follow to make their modeling workflows and their models reproducible. We urge modelers to systemize and automate their model construction processes and to record all data and software used by a workflow. When modelers publish models and findings, we encourage them to publicly share digital archives of their organized and documented artifacts, so that other researchers can reproduce their workflow and findings. These nine best practices can be easily integrated into a modeling workflow, because best practices are provided for each workflow stage, and useful software tools and data formats are recommended for each best practice (see Figure 1).

To see an example of many of these best practices at work in a single modeling workflow, we point readers to the executable simulation model (EXSIMO), recently published as a preprint in bioRxiv (König, 2020). This is an exemplary case study, which applies many of the concepts discussed in this text, to create a reproducible model of the liver, and can serve as a practical guide. For example, the EXSIMO platform encodes an executable simulation model of the liver using the SBML model description format and makes the simulation experiments compatible with SED-ML. While these steps alone would enable exchangeability across model construction and simulation platforms that support the standards, the EXSIMO platform also provides the entire simulation environment within a Docker image, allowing all validation tests and analyses to be executed within a container when run by an independent researcher. This step ensures that model and its simulation studies can be readily distributed. Finally, the EXSIMO platform provides extensive unit testing to verify and validate the model, and through GitHub

releases, is version controlled. Through these techniques, which are rigorously employed by the computer science community, the EXSIMO platform achieves a high level of quality control that will greatly benefit researchers interested in adapting this model or studying its predictions.

### Levels to Reproducible Biochemical Modeling

Following all best practices is an aspirational goal; this may become routine as the field grows and the importance of reproducible modeling becomes a more tangible concern, but adopting these guidelines is not an all-or-nothing challenge. Some of the recommended best practices may require additional training and effort to adopt new tools, and modelers may be concerned that this will distract from the scientific endeavor. There are costs associated with this transition. However, even adding just a few of these practices to a modeling project could provide notable benefits to reproducibility and enable long-term accessibility by the greater scientific community. We hope that modelers will consider these benefits and implement the best practices when possible. To further facilitate this goal, we have developed checklists that modelers may use to track their reproducibility progress as they execute a modeling workflow, encouraging them to work toward reproducibility (see Supplemental Information for checklists). These checklists are presented as levels of reproducible modeling, such that modelers can work toward an idealized modeling workflow, which most dramatically improves the ease of reproduction. We envision a future of biochemical modeling in which models undergo versioning and are iterated over, similar to semantic versioning, which is practiced within the software development community. In this way, models can be updated to facilitate reproducibility, and improve biological relevance and utility, but access to earlier versions can still be maintained to keep a complete provenance record for models derived from these versions. Using the provided checklists, a modeler could produce the first version of their model following easy-to-adopt reproducibility practices (see the general-purpose reproducible biochemical modeling checklist in the Supplemental Information). Over time, more rigorous changes could be implemented, adopting standardized formats whenever applicable to gradually move toward an idealized workflow (see the standards-based reproducible biochemical modeling checklist in the Supplemental Information). We encourage modelers to steadily improve the reproducibility of their models and modeling workflow.

### Standards in Biochemical Modeling: Progress and Limitations

We encourage modelers to employ standard data formats and tools that use these formats when possible because doing so greatly reduces the effort required to make reproducible models. As evidenced by the many standards and tools discussed in this paper, our field has made great strides toward automating and simplifying reproducible biochemical modeling in the last two decades. To evaluate this progress, we have collected data on the impact and adoption rates for the recommended scientific standards and tools (Table 2). All but one of these tools and standards were developed in the last two decades. Two types of evaluation data are provided, annual citation rates and adoption rates reported by a survey of the biochemical modeling commu-

nity (Szigeti et al., 2018a). The most influential and widely adopted biochemical modeling tools and standards include SBML, COPASI, SBGN, COBRApy, BioPAX, BioModels, Pathway Tools, InChI, BioNetGen, and SED-ML. A couple of tools and standards, notably SciPy and FAIR, score highly, because they are used broadly by science beyond biochemical modeling. Many other tools and standard have begun to develop a following and may become the leading approach in their domains in the future.

Nevertheless, some aspects of biochemical modeling still lack good standards or tools for reproducibility. For example, no standards or tools are available for aggregating and curating data, and the standardized methods available for saving simulation results lack important functionality. Because some models cannot be built using only a standards-based approach, we offer general-purpose approach recommendations for reproducible biochemical modeling that provide conceptual and practical guidance when standards-based tools are insufficient.

We have identified three core limitations to our standards-based guidance—limited domain coverage, limited functionality, and limited compatibility. Many of the recommended tools provided in these best practices (e.g., SBML, SED-ML, and COPASI) only support modeling of biochemical dynamics, limiting the domain coverage, or the types of biological processes and components that can be represented. However, modelers in the biochemical domain may want to study many other aspects of cells, such as evolution, motility, and replication. These domains of study must either build custom models or employ tools that are not standard-compatible. The tools we recommend also have functional limitations in the domains that they serve. For example, none of the recommended tools are suitable for modeling biochemical models of cells at the genome scale. Whole-cell models require tools that can scale to tens of thousands of species types and reactions, identify parameters for models of this size, and employ multi-algorithmic simulation to integrate pathways, which are characterized with variable levels of detail. Tools that address these limitations are under development (Medley et al., 2016; Goldberg et al., 2018; Schwab et al., 2000). The final chronic problem is that tools rarely support a standard in its entirety, limiting their compatibility. This problem grows worse when standards change frequently and when the tools are produced as academic projects with limited funding and high staff turnover. An example of this problem is that while the SBML standard has augmented functionality provided through packages for flux-balance analysis and hierarchical models, no SBML simulator supports all of the additional packages. Given these limitations, the use of general-purpose methods can be quite advantageous and even necessary for investigating certain biological inquiries.

### CONCLUSIONS

Although reproducibility is a core tenet of the scientific method, until recently it has been difficult to reproducibly construct biochemical models, because suitable standards and software tools did not exist. Over the last two decades, the biochemical modeling community has addressed this problem by developing and adopting standards and tools that make reproducible construction of many biochemical models feasible.



**Table 2. Influence of Standards and Tools**

Standard or Tool	Type of Standard or Tool	Most Cited Paper	Paper Year	PubMed (Cites per Year)	Scholar (Cites per Year)	Reported Use (%)
SciPy optimize, ODE solver, etc.	Simulator	<a href="#">Virtanen et al., 2020</a>	2020	39.8	881.4	–
FAIR	Modeling process guidelines	<a href="#">Wilkinson et al., 2016</a>	2016	142	590.7	–
SBML	Modeling language	<a href="#">Hucka et al., 2003</a>	2003	50	183.8	69.5
COPASI	Modeling application	<a href="#">Hoops et al., 2006</a>	2006	–	149.6	31.7
SBGN	Modeling visualization language	<a href="#">Le Novère et al., 2009</a>	2009	–	71	2.4
COBRApy	Simulator	<a href="#">Ebrahim et al., 2013</a>	2013	22.2	59.6	11.3
BioPAX	Biochemical data manager	<a href="#">Demir et al., 2010</a>	2010	24.8	56.5	–
BioModels	Model repository	<a href="#">Li et al., 2010</a>	2010	17.3	52.1	33.3
Pathway Tools	Biochemical data manager	<a href="#">Karp et al., 2002</a>	2002	–	39.6	–
InChI	Biochemical data standard	<a href="#">Heller et al., 2015</a>	2015	15.6	38.5	–
The ontology for biomedical investigations	Ontology	<a href="#">Bandrowski et al., 2016</a>	2016	–	33.4	–
Physiome	Model repository	<a href="#">Hunter and Borg, 2003</a>	2003	4.9	32.8	–
KisAO	Ontology	<a href="#">Courtot et al., 2011</a>	2011	–	23.9	–
SBO	Ontology	<a href="#">Courtot et al., 2011</a>	2011	–	23.9	–
Data2Dynamics	Parameter estimation tool	<a href="#">Raue et al., 2015</a>	2015	7.8	23.7	1.1
BioNetGen	Modeling language	<a href="#">Harris et al., 2016</a>	2016	7.7	22.5	8.5
SED-ML	Simulation description language	<a href="#">Waltemath et al., 2011b</a>	2011	8.2	20.6	–
JWS Online	Modeling application	<a href="#">Olivier and Snoep, 2004</a>	2004	4.6	19.9	5.4
PySB	Modeling language	<a href="#">Lopez et al., 2013</a>	2013	6.3	19.6	6.5
CellML	Modeling language	<a href="#">Cuellar et al., 2003</a>	2003	–	18.2	7.3
Vcell	Modeling application	<a href="#">Moraru et al., 2008</a>	2008	6	17.7	2.7
MIASE	Modeling process guidelines	<a href="#">Waltemath et al., 2011a</a>	2011	3.3	12.4	–
PROV	Reproducibility standard	<a href="#">Moreau et al., 2015</a>	2015	–	12.2	–
Memote	Validation tool	<a href="#">Lieven et al., 2018</a>	2018	–	12.1	–
libRoadRunner	Simulator	<a href="#">Somogyi et al., 2015</a>	2015	4.6	11.1	5.4
FAIRDOMHub	Model repository	<a href="#">Wolstencroft et al., 2017</a>	2017	3.2	10.6	–
SEEK	Model repository	<a href="#">Wolstencroft et al., 2015</a>	2015	3.1	10.6	3.8
SESSL	Simulation description language	<a href="#">Ewald and Uhrmacher, 2014</a>	2014	0	10.6	–
BioLab	Validation tool	<a href="#">Clarke et al., 2008</a>	2008	–	10.1	–
COMBINE	Biochemical data format	<a href="#">Bergmann et al., 2014</a>	2014	3.6	9.5	–
RightField	Model data annotation tool	<a href="#">Wolstencroft et al., 2011</a>	2011	2.4	8.6	–
OpenCOR	Simulator	<a href="#">Garny and Hunter, 2015</a>	2015	2.6	8.1	3.8
MIRIAM	Modeling process guidelines	<a href="#">Laibe and Le Novère, 2007</a>	2007	3.1	7.9	–
PyDREAM	Parameter estimation tool	<a href="#">Shockley et al., 2018</a>	2018	2.5	7.9	–
SBML-PET	Parameter estimation tool	<a href="#">Zi and Klipp, 2006</a>	2006	1.7	7.1	–
StochSS	Simulator	<a href="#">Drawert et al., 2016</a>	2016	2.7	7	–
Tellurium	Modeling application	<a href="#">Choi et al., 2018</a>	2018	2.1	6.7	4.8
SBRML	Model data annotation tool	<a href="#">Dada et al., 2010</a>	2010	1.7	6.2	–
PyBioNetFit	Validation tool	<a href="#">Mitra et al., 2019</a>	2019	2.1	5	–
Sbtabs	Biochemical data format	<a href="#">Lubitz et al., 2016</a>	2016	1.8	5	–
HDF	Data format	<a href="#">Brown et al., 1993</a>	1993	–	2.5	–
SciUnit	Validation tool	<a href="#">Omar et al., 2014</a>	2014	–	2.3	–

(Continued on next page)

**Table 2. Continued**

Standard or Tool	Type of Standard or Tool	Most Cited Paper	Paper Year	PubMed (Cites per Year)	Scholar (Cites per Year)	Reported Use (%)
SEPIO	Ontology	<a href="#">Brush et al., 2016</a>	2016	–	1.8	–
BpForms and BcForms	Biochemical data format	<a href="#">Lang et al., 2020</a>	2019	–	0.7	–
ObjTables	Biochemical data format	<a href="#">Karr et al., 2020</a>	2020	–	0	–

The influence and adoption rates of the scientific standards and tools recommended above are quantified by three measures. The annual citation rate for the primary publication is measured by Google Scholar and, where available, by PubMed. The latter reflects the impact of the work on biomedical research. Entries are sorted by their Google Scholar citation rates. Where available, adoption rates have been integrated from the 210 responses to a 2017 survey of 542 scientists in a broad range of biomodeling and related experimental disciplines ([Szigeti et al., 2018a](#)). The survey asked scientists which tools, resources, or languages they most frequently used.

This analysis employed reproducible methods. Two hand-curated tables were input: a list of scientific standards and tools containing the titles of their primary publications and a BibTeX bibliography containing the papers. The publication year and Google Scholar citation counts were obtained for each paper via a Google Scholar API. PubMed citation counts were obtained via the PubMed API ([NCBI Resource Coordinators, 2014](#)). Survey results were integrated from a spreadsheet in the ([Szigeti et al., 2018a](#)) public repository for the survey ([Szigeti et al., 2018b](#)). The analysis can be reproduced by executing two Python commands in the public repository that contains the analysis' hand-curated tables and source code ([Goldberg, 2020](#)).

We seek to further these advances by providing a comprehensive and practical set of best practices for reproducibly creating biochemical models. We recommend specific standards and tools for each stage of model development. But if some stages of model construction cannot employ the recommended standards and tools, a modeler can still implement our general-purpose guidelines, which can be applied to any method. Biochemical models constructed by following the recommended practices will be easier to understand, trust, and reuse.

We envision a biochemical modeling community that routinely publishes reproducible and reusable models, and which provides open access to their model artifacts. This would dramatically reduce the effort modelers must devote to making larger and more complex models by enabling reuse of models and data, and facilitating collaboration. Achieving this vision would accelerate the contributions made by modeling toward advancing our understanding of biology and medicine.

## SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at <https://doi.org/10.1016/j.cels.2020.06.012>.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the generous support given by the NIBIB of the National Institutes of Health under award number P41-EB023912. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

## REFERENCES

- Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J.A., and Blom, J.G. (2009). Systems biology: parameter estimation for biochemical models. *FEBS Journal* 276, 886–902.
- American Journal of Political Science (AJPS). (2016). American journal of political science qualitative data verification checklist. <https://ajps.org/wp-content/uploads/2019/01/ajps-qualdata-checklist-ver-1-0.pdf>.
- Bandrowski, A., Brinkman, R., Brochhausen, M., Brush, M.H., Bug, B., Chibucos, M.C., Clancy, K., Courtot, M., Derom, D., Dumontier, M., et al. (2016). The ontology for biomedical investigations. *PLoS One* 11, e0154556.
- Bergmann, F.T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., Hucka, M., Laibe, C., Miller, A.K., Nickerson, D.P., et al. (2014). COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics* 15, 369.
- Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., et al. (2001). Minimum information about a microarray experiment (MIAME)- toward standards for microarray data. *Nat. Genet.* 29, 365–371.
- Brindescu, C., Codoban, M., Shmarkatiuk, S., and Dig, D. (2014). How do centralized and distributed version control systems impact software changes? *Proceedings of the 36th international conference on Software Engineering*, 322–333.
- Brown, S.A., Folk, M., Goucher, G., Rew, R., and Dubois, P.F. (1993). Software for portable scientific data management. *Comput. Phys.* 7, 304.
- Brush, M.H., Shefchek, K., and Haendel, M. (2016). SEPIO: a semantic model for the integration and analysis of scientific evidence. *CEUR Workshop Proceedings* 1747. <http://ceur-ws.org/Vol-1747/>.
- Bustin, S.A., Benes, V., Garson, J.A., Hellemans, J., Huggett, J., Kubista, M., Mueller, R., Nolan, T., Pfaffl, M.W., Shipley, G.L., et al. (2009). The MIQE guidelines: minimum information for publication of quantitative real-time PCR experiments. *Clin. Chem.* 55, 611–622.
- Casadevall, A., and Fang, F.C. (2010). Reproducible science. *Infect. Immun.* 78, 4972–4975.
- Choi, K., Medley, J.K., König, M., Stocking, K., Smith, L., Gu, S., and Sauro, H.M. (2018). Tellurium: an extensible python-based modeling environment for systems and synthetic biology. *Biosystems* 171, 74–79.
- Choi, K., Smith, L.P., Medley, J.K., and Sauro, H.M. (2016). phraSED-ML: a paraphrased, human-readable adaptation of SED-ML. *J. Bioinform. Comp. Biol.* 14, 1650035.
- Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., and Tacchella, A. (2002). NuSMV 2: an OpenSource tool for symbolic model checking. In *Lecture Notes in Computer Science*, E. Brinksma and K.G. Larsen, eds. (Springer), pp. 359–364.
- Clarke, E.M., Faeder, J.R., Langmead, C.J., Harris, L.A., Jha, S.K., and Legay, A. (2008). Statistical model checking in BioLab: applications to the automated analysis of T-cell receptor signaling pathway. In *Computational Methods in Systems Biology*, M. Heiner and A.M. Uhrmacher, eds. (Springer), pp. 231–250.
- Center for Open Science (COS). (2019). Journals that issue open science badges. <https://cos.io/our-services/open-science-badges/>.
- Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., et al. (2011). Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.* 7, 543.
- Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P., and Hunter, P.J. (2003). An overview of CellML 1.1, a biological model description language. *Simulation* 79, 740–747.

- Dada, J.O., Spasić, I., Paton, N.W., and Mendes, P. (2010). SBRML: a markup language for associating systems biology data with models. *Bioinformatics* 26, 932–938.
- De Schutter, E. (2010). Data publishing and scientific journals: the future of the scientific paper in a world of shared data. *Neuroinformatics* 8, 151–153.
- Deelman, E., Berriman, G., Chervenak, A., Corcho, O., Groth, P., and Moreau, L. (2010). Metadata and provenance management. In *Semantic Data Management: challenges, technology, and deployment* (CRC Press), pp. 433–467.
- Demir, E., Cary, M.P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J., et al. (2010). The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.* 28, 935–942.
- Deutsch, E.W., Ball, C.A., Berman, J.J., Bova, G.S., Brazma, A., Bumgarner, R.E., Campbell, D., Causton, H.C., Christiansen, J.H., Daian, F., et al. (2008). Minimum information specification for in situ hybridization and immunohistochemistry experiments (MISFISHIE). *Nat. Biotechnol.* 26, 305–312.
- Donoho, D.L. (2010). An invitation to reproducible computational research. *Biostatistics* 11, 385–388.
- Drawert, B., Hellander, A., Bales, B., Banerjee, D., Bellesia, G., Daigle, B.J., Jr., Douglas, G., Gu, M., Gupta, A., Hellander, S., et al. (2016). Stochastic simulation service: bridging the gap between the computational expert and the biologist. *PLoS Comput. Biol.* 12, e1005220.
- Ebrahim, A., Lerman, J.A., Palsson, B.O., and Hyduke, D.R. (2013). Cobrapy: constraints-based reconstruction and analysis for python. *BMC Syst. Biol.* 7, 74.
- Elofsson, A., Hess, B., Lindahl, E., Onufriev, A., van der Spoel, D., and Wallqvist, A. (2019). Ten simple rules on how to create open access and reproducible molecular simulations of biological systems. *PLoS Comput. Biol.* 15, e1006649.
- EMBOpress. (2019). Source data. <https://www.emboress.org/sourcedata>.
- Ewald, R., and Uhrmacher, A.M. (2014). SESSL: a domain-specific language for simulation experiments. *ACM Trans. Model. Comput. Simul.* 24, 1–25.
- Garny, A., and Hunter, P.J. (2015). Opencor: a modular and interoperable approach to computational biology. *Front. Physiol.* 6, 26.
- Goldberg, A.P. (2020). Reproducible standards and tools analysis. [https://github.com/KarrLab/paper\\_cell\\_sys\\_guidelines\\_4\\_repro\\_models\\_2020](https://github.com/KarrLab/paper_cell_sys_guidelines_4_repro_models_2020).
- Goldberg, A.P., Szigeti, B., Chew, Y.H., Sekar, J.A., Roth, Y.D., and Karr, J.R. (2018). Emerging whole-cell modeling principles and methods. *Curr. Opin. Biotechnol.* 51, 97–102.
- Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.
- Grecco, H., Thielen, J., and collaborators. (2020). Managing units of physical quantities in python. <https://github.com/hgrecco/pint>.
- Gries, D. (2012). *The Science of Programming* (Springer Science and Business Media).
- Harris, L.A., Hogg, J.S., Tapia, J.J., Sekar, J.A.P., Gupta, S., Korsunsky, I., Arora, A., Barua, D., Sheehan, R.P., and Faeder, J.R. (2016). BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics* 32, 3366–3368.
- Heller, S.R., McNaught, A., Pletnev, I., Stein, S., and Tchekhovskoi, D. (2015). InChI, the IUPAC international chemical identifier. *J. Cheminform.* 7, 23.
- Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI—a Complex Pathway Simulator. *Bioinformatics* 22, 3067–3074.
- Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., and Cornish-Bowden, A. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.
- Hunter, P.J., and Borg, T.K. (2003). Integration from proteins to organs: the physiome project. *Nat. Rev. Mol. Cell Biol.* 4, 237–243.
- Huynh, T.D., Groth, P., and Zednik, S. (2013). PROV implementation report, Technical Report <https://eprints.soton.ac.uk/id/eprint/358440>.
- Jones, E., Oliphant, T., and Peterson, P. (2001). SciPy: open source scientific tools for Python. <http://www.scipy.org/>.
- Karp, P.D., Paley, S., and Romero, P. (2002). The pathway tools software. *Bioinformatics* 18, S225–S232.
- Karr, J.R., Liebermeister, W., Goldberg, A.P., Sekar, J.A.P., and Shaikh, B. (2020). Structured spreadsheets with objtables enable data reuse and integration. *arXiv*, arXiv:2005.05227.
- Karr, J.R., Sanghvi, J.C., Macklin, D.N., Gutschow, M.V., Jacobs, J.M., Bolival, B., Jr., Assad-Garcia, N., Glass, J.I., and Covert, M.W. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell* 150, 389–401.
- Kazic, T. (2015). Ten simple rules for experiments' provenance. *PLoS Comput. Biol.* 11, e1004384.
- Kidwell, M.C., Lazarević, L.B., Baranski, E., Hardwicke, T.E., Piechowski, S., Falkenberg, L.S., Kennett, C., Slowik, A., Sonnleitner, C., Hess-Holden, C., et al. (2016). Badges to acknowledge open practices: a simple, low-cost, effective method for increasing transparency. *PLoS Biol.* 14, e1002456.
- König, M. (2020). Executable simulation model of the liver. *bioRxiv*. <https://doi.org/10.1101/2020.01.04.894873>.
- Kwiatkowska, M., Norman, G., and Parker, D. (2011). PRISM 4.0: verification of probabilistic real-time systems. In *Computer Aided Verification. CAV 2011. Lecture Notes in Computer Science*, 6806 (Springer), pp. 585–591.
- Laibe, C., and Le Novère, N. (2007). Miriam resources: tools to generate and resolve robust cross-references in systems biology. *BMC Syst. Biol.* 1, 58.
- Lang, P.F., Chebaro, Y., Zheng, X., Sekar, J.A.P., Shaikh, B., Natale, D., and Karr, J.R. (2020). BpForms and BcForms: tools for concretely describing non-canonical polymers and complexes to facilitate comprehensive biochemical networks. *Genome Biol.* 22, 117.
- Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., et al. (2009). The systems biology graphical notation. *Nat. Biotechnol.* 27, 735–741.
- Li, C., Donizelli, M., Rodriguez, N., Dharuri, H., Endler, L., Chelliah, V., Li, L., He, E., Henry, A., Stefan, M.I., et al. (2010). BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst. Biol.* 4, 92.
- Lieven, C., Beber, M.E., Olivier, B.G., Bergmann, F.T., Ataman, M., Babaei, P., Bartell, J.A., Blank, L.M., Chauhan, S., Correia, K., et al. (2018). Memote: a community driven effort towards a standardized genome-scale metabolic model test suite. *bioRxiv*. <https://doi.org/10.1101/350991>.
- Lopez, C.F., Muhlich, J.L., Bachman, J.A., and Sorger, P.K. (2013). Programming biological models in Python using PySB. *Mol. Syst. Biol.* 9, 646.
- Lubitz, T., Hahn, J., Bergmann, F.T., Noor, E., Klipp, E., and Liebermeister, W. (2016). SBtab: a flexible table format for data exchange in systems biology. *Bioinformatics* 32, 2559–2561.
- Medley, J.K., Goldberg, A.P., and Karr, J.R. (2016). Guidelines for reproducibly building and simulating systems biology models. *IEEE Trans. Bio Med. Eng.* 63, 2015–2020.
- Mišević, L., and Hatzimanikatis, V. (2011). Modeling of uncertainties in biochemical reactions. *Biotechnol. Bioeng.* 108, 413–423.
- Mitra, E.D., Suderman, R., Colvin, J., Ionkov, A., Hu, A., Sauro, H.M., Posner, R.G., and Hlavacek, W.S. (2019). Pybionetfit and the biological property specification language. *iScience* 19, 1012–1036.
- Mobley, A., Linder, S.K., Braeuer, R., Ellis, L.M., and Zwilling, L. (2013). A survey on data reproducibility in cancer research provides insights into our limited ability to translate findings from the laboratory to the clinic. *PLoS One* 8, e63221.
- Moraru, I.I., Schaff, J.C., Slepchenko, B.M., Blinov, M.L., Morgan, F., Lakshminarayana, A., Gao, F., Li, Y., and Loew, L.M. (2008). Virtual cell modelling and simulation software environment. *IET Syst. Biol.* 2, 352–362.
- Moreau, L., Groth, P., Cheney, J., Lebo, T., and Miles, S. (2015). The rationale of PROV. *J. Web Semant.* 35, 235–257.

- NCBI Resource Coordinators (2014). Database resources of the national center for biotechnology information. *Nucleic Acids Res* 42, D7–17.
- Olivier, B.G., and Snoep, J.L. (2004). Web-based kinetic modelling using JWS Online. *Bioinformatics* 20, 2143–2144.
- Omar, C., Aldrich, J., and Gerkin, R.C. (2014). Collaborative infrastructure for test-driven scientific model validation. In *Proceedings of the 36th international conference on Software Engineering*, Hyderabad, India, pp. 524–527. <http://www.cs.cmu.edu/~aldrich/papers/sciunit-icse14.pdf>.
- Orchard, S., Salwinski, L., Kerrien, S., Montecchi-Palazzi, L., Oesterheld, M., Stümpflen, V., Ceol, A., Chatr-aryamontri, A., Armstrong, J., Woollard, P., et al. (2007). The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nat. Biotechnol.* 25, 894–898.
- Pebesma, E., Mailund, T., and Hiebert, J. (2016). Measurement units in R. *R J* 8, 486.
- Peng, R.D. (2011). Reproducible research in computational science. *Science* 334, 1226–1227.
- Prinz, F., Schlange, T., and Asadullah, K. (2011). Believe it or not: how much can we rely on published data on potential drug targets? *Nat. Rev. Drug Discov.* 10, 712.
- Raue, A., Steiert, B., Schelker, M., Kreutz, C., Maiwald, T., Hass, H., Vanlier, J., Tönsing, C., Adlung, L., Engesser, R., et al. (2015). Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics* 31, 3558–3560.
- Rosen, L. (2005). *Open Source Licensing*, Volume 692 (Prentice Hall).
- Sandve, G.K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Comput Biol* 9, e1003285.
- Sargent, R.G. (2010). Verification and validation of simulation models. In *Proceedings of the 2010 Winter Simulation Conference*, IEEE, B. Johansson, S. Jain, J. Montoya-Torres, J. Huan, and E. Yücesan, eds., pp. 166–183. <https://www.informs-sim.org/wsc10papers/016.pdf>.
- Schmidt, K. (2000). LoLA a low level analyser. In *Application and Theory of Petri Nets 2000*. ICATPN 2000. *Lecture Notes in Computer Science*, vol. 1825, M. Nielsen and D. Simpson, eds. (Springer), pp. 465–474.
- Schwab, M., Karrenbach, N., and Claerbout, J. (2000). Making scientific computations reproducible. *Comput. Sci. Eng.* 2, 61–67.
- Sever, R., Eisen, M., and Inglis, J. (2019). Plan u: universal access to scientific and medical research via funder preprint mandates. *PLoS Biol* 17, e3000273.
- Sherman, M.A., Middleton, J.L., Schmidt, J.P., Paik, D.S., Blemker, S.S., Habib, A.W., Anderson, F.C., Delp, S.L., and Altman, R.B. (2005). The simtk framework for physics-based simulation of biological structures: preliminary design. In *Proceedings of the workshop on component models and frameworks in high performance computing*.
- Shockley, E.M., Vrugt, J.A., and Lopez, C.F. (2018). PyDREAM: high-dimensional parameter inference for biological models in python. *Bioinformatics* 34, 695–697.
- Sicilia, M.A., García-Barriocanal, E., and Sánchez-Alonso, S. (2017). Community curation in open dataset repositories: insights from Zenodo. *Procedia Comput. Sci.* 106, 54–60.
- Singh, J. (2011). Figshare. *J. Pharmacol. Pharmacother.* 2, 138–139.
- Smith, L.P., Bergmann, F.T., Chandran, D., and Sauro, H.M. (2009). Antimony: a modular model definition language. *Bioinformatics* 25, 2452–2454.
- Somogyi, E.T., Bouteiller, J.M., Glazier, J.A., König, M., Medley, J.K., Swat, M.H., and Sauro, H.M. (2015). libRoadRunner: a high performance SBML simulation and analysis library. *Bioinformatics* 31, 3315–3321.
- Szigeti, B., Roth, Y.D., Sekar, J.A., Goldberg, A.P., Pochiraju, S.C., and Karr, J.R. (2018b). Repository for Szigeti et al. survey. [https://github.com/KarrLab/paper\\_2018\\_curr\\_opin\\_sys\\_biol](https://github.com/KarrLab/paper_2018_curr_opin_sys_biol).
- Szigeti, B., Roth, Y.D., Sekar, J.A.P., Goldberg, A.P., Pochiraju, S.C., and Karr, J.R. (2018a). A blueprint for human whole-cell modeling. *Curr Opin Syst Biol* 7, 8–15.
- Taylor, C.F., Paton, N.W., Lilley, K.S., Binz, P.A., Julian, R.K., Jones, A.R., Zhu, W., Apweiler, R., Aebersold, R., Deutsch, E.W., et al. (2007). The minimum information about a proteomics experiment (MIAPE). *Nat. Biotechnol.* 25, 887–893.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* 17, 261–272.
- Waltemath, D., Adams, R., Beard, D.A., Bergmann, F.T., Bhalla, U.S., Britten, R., Chelliah, V., Cooling, M.T., Cooper, J., Crampin, E.J., et al. (2011a). Minimum information About a simulation experiment (MIASE). *PLoS Comput. Biol.* 7, e1001122.
- Waltemath, D., Adams, R., Bergmann, F.T., Hucka, M., Kolpakov, F., Miller, A.K., Moraru, I.I., Nickerson, D., Sahle, S., Snoep, J.L., and Novère, N.L. (2011b). Reproducible computational biology experiments with SED-ML – the simulation experiment description markup language. *BMC Syst. Biol.* 5, 198.
- Waltemath, D., and Le Novère, N. (2013). Simulation experiment description markup language (SED-ML). In *Encyclopedia of computational neuroscience* (Springer), pp. 1–4.
- Waltemath, D., and Wolkenhauer, O. (2016). How modeling standards, software, and initiatives support reproducibility in systems biology and systems medicine. *IEEE Trans. Bio Med. Eng.* 63, 1999–2006.
- Watanabe, L., Nguyen, T., Zhang, M., Zundel, Z., Zhang, Z., Madsen, C., Roehner, N., and Myers, C. (2019). iBioSim 3: a tool for model-based genetic circuit design. *ACS Synth. Biol.* 8, 1560–1563.
- White, G.H. (2008). Basics of estimating measurement uncertainty. *Clinical Biochemistry Reviews* 29, 53–60.
- Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* 3, 160018.
- Woelfle, M., Olhio, P., and Todd, M.H. (2011). Open science is a research accelerator. *Nat. Chem.* 3, 745–748.
- Wolstencroft, K., Krebs, O., Snoep, J.L., Stanford, N.J., Bacall, F., Golebiewski, M., Kuzyakiv, R., Nguyen, Q., Owen, S., Soiland-Reyes, S., et al. (2017). FAIRDOMHub: a repository and collaboration environment for sharing systems biology research. *Nucleic Acids Res* 45, D404–D407.
- Wolstencroft, K., Owen, S., Horridge, M., Krebs, O., Mueller, W., Snoep, J.L., du Preez, F., and Goble, C. (2011). RightField: embedding ontology annotation in spreadsheets. *Bioinformatics* 27, 2021–2022.
- Wolstencroft, K., Owen, S., Krebs, O., Nguyen, Q., Stanford, N.J., Golebiewski, M., Weidemann, A., Bittkowski, M., An, L., Shockley, D., et al. (2015). SEEK: a systems biology data and model management platform. *BMC Syst. Biol.* 9, 33.
- Zi, Z., and Klipp, E. (2006). SBML-PET: a systems biology markup language-based parameter estimation tool. *Bioinformatics* 22, 2704–2705.