Article in Press

From FAIR to CURE: guidelines for computational models of biological systems

Herbert M. Sauro, Eran Agmon, Michael L. Blinov, John H. Gennari, Joseph L. Hellerstein, Adel Heydarabadipour, Bartholomew E. Jardine, Elebeoba May, David P. Nickerson, Lucian P. Smith, Gary D. Bader, Frank T. Bergmann, Patrick M. Boyle, Andreas Dräger, James R. Faeder, Song Feng, Juliana Freire, Fabian Fröhlich, James A. Glazier, Thomas E. Gorochowski, Tomas Helikar, Henning Hermjakob, Stefan Hoops, Peter Hunter, Princess I. Imoukhuede, Sarah M. Keating, Matthias König, Reinhard Laubenbacher, Leslie M. Loew, Carlos F. Lopez, William W. Lytton, Rahuman S. Malik-Sheriff, Andrew McCulloch, Pedro Mendes, Lealem Mulugeta, Chris J. Myers, Jerry G. Myers Jr, Anna Niarakis, David D. van Niekerk, Brett G. Olivier, Alexander A. Patrie, Ellen M. Quardokus, Nicole Radde, Johann M. Rohwer, Sven Sahle, James C. Schaff, Falk Schreiber, T. J. Sego, Janis Shin, Jacky L. Snoep, Rajanikanth Vadigepalli, H. Steven Wiley, Dagmar Waltemath & Ion I. Moraru

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

From FAIR to CURE: Guidelines for Computational Models of Biological Systems

Herbert M. Sauro[1,4*], Eran Agmon[10], Michael L. Blinov[10], John H. Gennari[21], Joseph L. Hellerstein[1,4], Adel Heydarabadipour[1], Bartholomew E. Jardine[1], Elebeoba May[19], David P. Nickerson[14], Lucian P. Smith[1], Gary D Bader[22], Frank T. Bergmann[33], Patrick M. Boyle[1,3,4,5], Andreas Dräger[18], James R. Faeder[26], Song Feng[13], Juliana Freire[20], Fabian Fröhlich[27], James A. Glazier[2], Thomas E. Gorochowski[6], Tomas Helikar[35], Henning, Hermjakob[40], Stefan Hoops[32], Peter Hunter[14], Princess I. Imoukhuede[1], Sarah M. Keating[30], Matthias König[29], Reinhard Laubenbacher[28], Leslie M. Loew[10], Carlos F. Lopez[17], William W. Lytton[7,8], Rahuman S. Malik-Sheriff [40], Andrew McCulloch[36], Pedro Mendes[10], Lealem Mulugeta[23,24], Chris J. Myers[25], Jerry G. Myers Jr.[9], Anna Niarakis[15,16], David D. van Niekerk[34], Brett G. Olivier[11], Alexander A. Patrie[10], Ellen M. Quardokus[2], Nicole Radde[42], Johann M. Rohwer[34], Sven Sahle[33], James C. Schaff[10], Falk Schreiber[38,39], T.J. Sego[28], Janis Shin[1], Jacky L. Snoep[34], Rajanikanth Vadigepalli[12], H. Steven Wiley[41], Dagmar Waltemath[37], Ion Moraru[10*]

[1*]Department of Bioengineering, University of Washington, Seattle, 98195-5061, WA, USA.

[2]Intelligent Systems Engineering and Biocomplexity Institute, Indiana University, Street, Bloomington, 47408, Indiana, USA.

[3]Center for Cardiovascular Biology, University of Washington, Seattle, 98195-5061, WA, USA.

[4]eScience Institute, University of Washington, Seattle, 98195-5061, WA, USA.

[5]Institute for Stem Cell and Regenerative Medicine, University of Washington, Seattle, 98195-5061, WA, USA.

[6]School of Biological Sciences, University of Bristol, 24 Tyndall Avenue, Bristol, BS8 1TQ, UK.

[7]Departments of Physiology & Pharmacology, Neurology, Downstate Health Science University, Brooklyn, 11203, NY, USA.

[8]Department of Neurology, Kings County Hospital, Brooklyn, 11203, NY, USA.

[9]NASA - Glenn Research Center, MS 110-3, 21000 Brookpark Road, Cleveland, 44135, Ohio, USA.

[10]Center for Cell Analysis and Modeling, UConn Health, 263 Farmington Avenue, Farmington, 06030-6406, Connecticut, USA.

[11]Amsterdam Institute for Life and Environment, Vrije Universiteit Amsterdam, De Boelelaan 1108, 1081 HZ, Amsterdam, Netherlands.

[12]Department of Pathology and Genomic Medicine, Thomas Jefferson University, 1020 Locust St, Philadelphia, 19107, Pennsylvania, USA.

[13]Biological Sciences Division, Pacific Northwest National Laboratory, 902 Battelle Blvd, Richland, 99354, WA, USA.

[14]Auckland Bioengineering Institute, University of Auckland, Auckland, 1010, New Zealand.

[15]Molecular, Cellular and Developmental Biology Unit (MCD), Center of Integrative Biology, University of Toulouse III-Paul Sabatier, 165 Rue Marianne Grunberg-Manago, Toulouse, 31400, France.

[16]Lifeware Group, Inria, Building Alan Turing, 1 Rue Honoré d'Estienne d'Orves, 91120, Palaiseau, France.

[17]Multiscale Modeling Group, Altos Labs, 94065, Redwood City, CA, USA.

[18]Martin Luther University Halle-Wittenberg, Data Analytics and Bioinformatics, Von-Seckendorff-Platz 1, 06120, Halle (Saale), Germany.

[19]Wisconsin Institute for Discovery, University of Wisconsin—Madison, 330 North Orchard Street, 53715, Madison, WI, USA.

[20]Department of Computer Science and Center for Data Science, New York University, New York, NY, 11201, New York, USA.

[21]Department of Biomedical Informatics & Medical Education, University of Washington, 1959 NE Pacific Street, 98195, Seattle, Washington, USA.

[22]The Donnelly Centre, University of Toronto, 160 College St, M5S 3E1, Toronto, Ontario, Canada.

[23]InSilico Labs LLC, 77027, Houston, Texas, USA.

[24]Medalist Performance, 77027, Houston, Texas, USA.

[25]Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, 425 UCB, Boulder, 80309, Colorado, USA.

[26]Department of Computational and Systems Biology, University of Pittsburgh, 3500 Fifth Avenue, 15213, Pittsburgh, Pennsylvania, USA.

[27]Dynamics of Living Systems Laboratory, The Francis Crick Institute, 1 Midland Road, NW1 1AT, London, UK.

[28]Department of Medicine, University of Florida, 1600 SW Archer Rd, 32610-0225, Gainesville, Florida, USA.

[29]Faculty of Life Science, Institute for Biology, Systems Medicine of Liver, Humboldt-University Berlin, Philippstraße 13, 10115, Berlin, Germany.

[30]Advanced Research Computing Centre, University College London, Gower Street, WC1E 6BT, London, UK.

[31]BioQUANT, Heidelberg University, Im Neuenheimer Feld 267, 69120, Heidelberg, Germany.

[32]Biocomplexity Institute, University of Virginia, Town Center Four, 3rd Floor, 994 Research Park Boulevard, 22911, Charlottesville, VA, USA.

[33]University of Heidelberg, Im Neuenheimer Feld 267, 69120, Heidelberg, Germany.

[34]Department of Biochemistry, University of Stellenbosch, Private Bag X1, 7602, Matieland, South Africa.

[35]Department of Biochemistry, University of Nebraska-Lincoln, Beadle Center, 68588-0664, Lincoln NE, USA.

[36]Departments of Bioengineering and Medicine, University of California San Diego, 9500 Gilman Drive, 92093-0412, La Jolla, CA, USA.

[37]Medical Informatics Laboratory, University Medicine Greifswald, D-17489, Greifswald, Germany.

[38]Department of Computer and Information Science, University of Konstanz, Universitaetsstrasse 10, 78464, Konstanz, Germany.

[39]Faculty of Information Technology, Monash University, 21 Chancellors Walk, 3800, Melbourne, VIC, Australia.

[40]European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Hinxton, Cambridgeshire, UK.

[41]Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, 902 Battelle Blvd, Richland, 99354, WA, USA.

[42]University of Stuttgart, Institute for Stochastics and Applications, Stuttgart, Germany, Germany.

*Corresponding author(s). E-mail(s): hsauro@uw.edu; moraru@uchc.edu; nicole.radde@simtech.uni-stuttgart.de;

Contributing authors: agmon@uchc.edu; blinov@uchc.edu; gennari@uw.edu; joseph.hellerstein@gmail.com; adelhp@uw.edu; barthj@uw.edu; emay5@wisc.edu; d.nickerson@auckland.ac.nz; lpsmith@uw.edu; gary.bader@utoronto.ca; frank.bergmann@bioquant.uni-heidelberg.de; pmjboyle@uw.edu; andreas.draeger@informatik.uni-halle.de; faeder@pitt.edu; song.feng@pnnl.gov; juliana.freire@nyu.edu; fabian.frohlich@crick.ac.uk; jaglazier@gmail.com; thomas.gorochowski@bristol.ac.uk; thelikar2@unl.edu; hhe@ebi.ac.uk; shoops@virginia.edu; p.hunter@auckland.ac.nz; pii@uw.edu; s.keating@ucl.ac.uk; koenigmx@hu-berlin.de; reinhard.laubenbacher@medicine.ufl.edu; les@uchc.edu; clopez@altoslabs.com; billl@neurosim.downstate.edu; sheriff@ebi.ac.uk; amcculloch@ucsd.edu; pmendes@uchc.edu; lealem@insilico-labs.com; chris.myers@colorado.edu; jerry.g.myers@nasa.gov; anna.niaraki@univ-tlse3.fr; ddvniekerk@sun.ac.za; b.g.olivier@vu.nl; apatrie@uchc.edu; ellenmq@iu.edu; jr@sun.ac.za; sven.sahle@bioquant.uni-heidelberg.de; schaff@uchc.edu; falk.schreiber@uni-konstanz.de; timothy.sego@medicine.ufl.edu; jshin1@uw.edu; jls@sun.ac.za; rajanikanth.vadigepalli@jefferson.edu; steven.wiley@pnnl.gov; dagmar.waltemath@uni-greifswald.de;

Abstract

Guidelines for managing scientific data have been established under the FAIR principles requiring that data be Findable, Accessible, Interoperable, and Reusable. In many scientific disciplines, especially computational biology, both data and *models* are key to progress. For this reason, and recognizing that such models are a very special type of "data", we argue that computational models,

especially mechanistic models prevalent in medicine, physiology and systems biology, deserve a complementary set of guidelines. We propose the CURE principles, emphasizing that models should be Credible, Understandable, Reproducible, and Extensible. We delve into each principle, discussing verification, validation, and uncertainty quantification for model credibility; the clarity of model descriptions and annotations for understandability; adherence to standards and open science practices for reproducibility; and the use of open standards and modular code for extensibility and reuse. We outline recommended and baseline requirements for each aspect of CURE, aiming to enhance the impact and trustworthiness of computational models, particularly in biomedical applications where credibility is paramount. Our perspective underscores the need for a more disciplined approach to modeling, aligning with emerging trends such as Digital Twins and emphasizing the importance of data and modeling standards for interoperability and reuse. Finally, we emphasize that given the non-trivial effort required to implement the guidelines, the community should strive to automate as many of the guidelines as possible.

1 Introduction

Wilkinson et al. in 2016 [1] made a good case (dare we say a fair case) for establishing guidelines for the management of scientific data. They arrived at four guiding principles enshrined in the acronym FAIR, namely that data be Findable, Accessible, Interoperable, and Reusable. With the rapid growth of computational modeling, especially the development of mechanistic models of physiological and cellular systems, the question arises of how these principles can be extended so that they can succinctly describe best practices for model management (e.g., model development, deployment, and interpretation). In this perspective, we introduce a set of complementary guidelines to FAIR that address the specific needs for mechanistic models. We identify four principles: Credibility, Understandability, Reproducibility, and Extensibility. We refer to these as the **CURE** principles.

We focus on mechanistic models, thereby excluding the growing body of machine-learned (ML) and AI models that are based solely on data. The machine learning community has appropriately encouraged the use of FAIR principles when publishing ML models [2, 3], with an emphasis on ensuring that data are accessible to those who wish to repeat the study and models to be explainable. However, the reproducibility of these models is a separate topic that has its own special concerns. Thus, we will not consider ML/AI models in this perspective.

Although the focus of this proposal is on models from the systems biology and systems medicine community, the guidelines and sentiments we describe are broadly applicable to other modeling domains.

2 Why have Guidelines?

Models are indispensable in many scientific and engineering disciplines. Examples include circuit simulation in electrical engineering, models of fluid flow in mechanical engineering, and weather modeling in atmospheric science. In some cases, modeling has progressed to the development of Digital Twins [4, 5], in which a simulation model is designed to replicate and interact with the physical system, or virtual populations, which can be used for applications such as clinical trial design [6]. Biological modeling has not yet risen to that same level of sophistication, with the possible exception of areas such as protein folding [7] and molecular dynamics [8] where physics and chemistry play a more important role. Even so, biological modeling is a rapidly evolving field. As the field grows, guidelines in the spirit of FAIR will help modelers create more impactful and credible models. We believe that these guidelines will be of importance for models that ultimately reach the clinic, and particularly for the growing interest in developing biomedical Digital Twins [9, 10] or whole-cell models [11–13]. The Working Group 'Building Immune Digital Twins' tries to address these questions, having as a focus the human immune system and its responses in various pathological contexts [9, 10].

3 Who will use the Guidelines?

Not all groups may need to use the CURE guidelines. For example, researchers who create conceptual models as part of a basic research project are unlikely to use guidelines. For academic researchers who build models of real systems, the choice has to be made by the researcher. At minimum, we recommend that a published model should at least be reproducible [14–17]. This does not necessarily mean the use of open standards, but models that do not use open standards should at least be well documented and annotated. The use of open standards would allow the model to be stored on one of the open model repositories, as well as making the model much more reusable but ultimately, this decision is at the discretion of the researcher. For models that may eventually reach the clinic, the use of the CURE guidelines is highly recommended. For those involved in building large models, such as Digital Twins [9, 10] or Whole-cell models [11–13], we highly recommend considering the use of the CURE guidelines.

4 Existing Guidelines

Several groups have proposed guidelines to improve best practices in creating biological models over the last 20 years. Of particular note is the creation of standardised languages for biological models [18] such as Systems Biology Markup Language (SBML) [19, 20], CellML [21] and NeuroML [22]. These are machine-readable formats that are an *explicit representation* of the model. By explicit representation, we mean that the model representation only includes elements that are essential for the model; it does not include implementation details related to simulation. For example,

the essential characteristics of a mechanistic model of a well-mixed biochemical system include chemical species, reactions, and rate laws. It does not include software details such as file input/output and control logic for numerical solvers. An explicit representation is *independent of its implementation* in software.

The choice of an explicit representation for models is driven by the requirements of the communities that develop and use the models. SBML focuses on biochemical models where the representation is in terms of the biological processes. CellML focuses on a mathematical representation of models as differential-algebraic equations. NeuroML focuses on representations of neural systems. These representations have become popular among modelers and software developers. For example, all genome-scale models [23] are represented using SBML, and thousands of kinetic models are now stored on publicly accessible model repositories using these formats [24–26]. Standards such as SBML avoid the use of potentially confusing and unreusable *ad hoc* models, allowing models to persist in a reproducible form over long periods of time [27, 28]. Many authors, however, still publish models in executable formats such as MATLAB, Python, etc., which can pose problems for reproducibility and reuse, particularly when poorly documented [15, 29]. The logical modeling community, which uses SBML Qualitative format (SBML-Qual) to encode logical and Boolean models [30], made efforts to define a roadmap toward the annotation and curation of logical models (also known as the CALM initiative), including milestones for best practices and recommended standard requirements [31].

To promote data sharing and reuse, the FAIR principles recommend a data dictionary that specifies data types and semantics for data items [1]. An analogous requirement exists for models. For example, consider an SBML model with the reaction $A \to B$. Annotations are used to define A, B, and provide information about the reaction (e.g., the organism, cell type, and organelle in which the reaction takes place). Annotation can provide additional ontological and reference information about a model, including its submodels, processes, assumptions, and provenance. Genome-scale models are heavily annotated with process metadata [32, 33], and the curators at BioModels [24] regularly add annotations to curated models. Guidelines for harmonizing semantic annotations for computational models have been established [34]. As part of these efforts, the systems biology and physiology community developed the MIRIAM standard, which describes the "Minimum Information Required In the Annotation of Models" [35] and the OMEX metadata specification [34]. MIRIAM applies to structured information, such as SBML or CellML, where annotation information can be included in a machine-readable manner. Such information can be of great utility for searching, merging, or disassembling a model into its component parts [36].

5 Mechanistic Models

The focus of this perspective is on mechanistic models. Such models are sometimes called deductive models in contrast to inductive models, which are purely data-driven [37]. We define a mechanistic model as: a representation of a biological system that is described in terms of the constituent physical parts and processes that occur

between the parts. For example, in a mechanistic model of a cell signaling pathway, we would specify the various proteins and their phosphorylation states and the transformations between these states via enzymatic processes involving kinases and phosphatases.

Often, a mechanistic model is transformed into a computational model by invoking physical laws, such as the conservation of mass and chemical kinetic laws that govern individual transformations. In physiology and systems biology, models are commonly represented as a system of ordinary differential equations [38, 39], but other representations are also used, such as systems of Boolean functions, graphs, stochastic systems, and constraint-based models [40]. Such models are often shared via model repositories such as BioModels [24], JWS Online [41], or BiGG [25] and KBase [26] for constraint-based models.

Models can also be shared using raw executable formats such as Python or MATLAB that describe differential equations. Such 'raw' model code is typically provided as supplementary files to a published paper or stored in code repositories such as GitHub or specialized repositories such as ModelDB [42]. In some cases, the model may be described as part of the main text of the published article [43] or be absent altogether. This obviously makes the reproducibility of such work much more complex and sometimes impossible, given the frequency of typological errors in printing mathematical equations or code fragments. Models-as-programs paradigms such as those followed by PySB [44] and others encourage the use of best practices in Python coding and documentation through tools such as Sphinx [45], but these approaches rely on developer effort to document the code and model at an appropriate level of detail. They also tie a model to a specific language implementation, making reuse difficult and backward compatibility a problem, though PySB does offer some SBML support.

It may be difficult to abstract the underlying physiological processes of some complex systems purely in explicit form, though there have been few efforts to attempt this. Multi-scale modeling tends to give rise to such challenging scenarios. The context of computational modeling of electrophysiological phenomena in the heart provides an illustrative example [46]. Ordinary differential equations describing cell-scale events like ion channel gating and the generation of action potentials can be encoded using CellML or as a biological description in SBML. However, representing the propagation of excitatory wavefronts requires spatial discretization of the governing partial differential equations via finite element analysis; explicit formats for such applications exist [47, 48] but have not seen widespread use. One approach to improving interoperability and reproducibility is to create tools for importing model components written in common data formats. For instance, the openCARP simulation environment [46] includes a CellML "translator" that facilitates on-the-fly incorporation of cell-scale representations of different types of cardiac electrophyiology (e.g., cardiac region, species, or disease-specific action potentials) within the fabric of the multi-scale simulation ecosystem.

6 The CURE Guidelines

As with FAIR, we define four specific sets of principles to improve best practices in developing mechanistic computational models of biological systems. There is no specific order in the principles but the acronym CURE seemed appropriate for the topic in question (Figure 1).
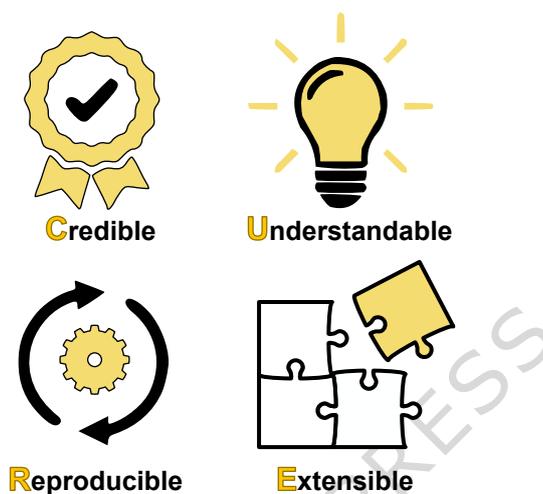

Fig. 1 Schematic representation of the CURE guidelines. The guidelines emphasize Credibility understandability, Reproducibility and Extensibility.

CURE [49] covers four key ideas, **Credibility**, **Understandability**, **Reproducibility**, and **Extensibility**, which we describe in the following sections. These guidelines are meant, where possible, to apply to both machine-readable formats, such as SBML, as well as models distributed in executable code, such as MATLAB, Python or Julia. They also align with previous community efforts to address barriers in comprehensiveness, accessibility, reusability, interoperability, and reproducibility of computational models in systems biology [50].

6.1 Credible

We use credibility to mean a perceived measure of believability [51, 52]. A credible model makes trustworthy and actionable predictions within the range of conditions it was intended to simulate. Prior work on model credibility dates back to 1979 [53], when the Society for Modeling & Simulation International (SCS) described many concepts associated with model credibility. More recently, several groups within the biological modeling community have discussed model credibility, most notably the ten rules developed by CPMS [54], devised by the Committee on Credible Practice of Modeling & Simulation in Healthcare (CMPS) in the US, and Musuamba et al. [55] in Europe who describe in some detail criteria and concepts that are important to

assess the model credibility. Of note, the CPMS working group considered credibility as a descriptor of the practice of modeling and simulation rather than that of a model. Accordingly, the assessment by the ten CPMS rules are geared towards evaluating the practices followed in the modeling efforts [56]. Two core concepts related to model credibility are verification and validation [57].

Verification is "the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model" [57]. Similar definitions can be found in many other documents [58]. In practice, verification means assessing the correctness of: (i) the model representation (e.g., detecting typographical errors); (ii) the numerical algorithms used to simulate the model (e.g., pseudo random numbers have the correct distributions); and (iii) the model implementation in software. Models implemented directly in programming languages such as Python may not have (i), and models implemented using standards such as SBML have few concerns about (iii).

When using standards such as SBML, verification involves ensuring that different software applications interpret the description of the model in the same way [59] and that the software has passed the parts of the SBML test suite relevant for the model [60]. It is worth noting the close correspondence of verification to software testing, including unit and system tests, and documentation [61]. Since models are almost always implemented in software, this should come as no surprise. Static tests can assess whether the model is correct, for example, that the biophysical laws have been entered correctly or that mass balance is not violated. Dynamic tests can offer more subtle checks to the correctness of a model. For example, a model whose variables are concentrations should not be able to reach negative values.

Validation is "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model" [57]. Of significant importance is the phrase "intended uses". All models have a limited scope, that is, they can only make useful predictions within their intended purpose and calibration. This is particularly important for models that might be used in a clinical setting where misuse of a model could have dire consequences. It is important, therefore, that there be a clear statement of the purpose of the model as well as the conditions under which the model is applicable. Validation involves comparing experimental data to predictions made by the model. In practice, given the nature of scientific models, not all model predictions can be validated, and validation is more a measure of confidence in a model's credibility to match reality than an absolute statement of truth.

Mechanistic models often have parameters whose values are determined through parameter fitting. This is a well-established area, which we will not cover it in detail here. Readers are referred to the following articles for further details [39, 62–73]

For parameter-free logic-based models in biology, the credibility lies in the causality of the statements used to build the logical rules and functions [74], as well as the binarized interpretation and use of small- and larger-scale experimental data [75].

Uncertainty Quantification. In recent years, there has been a growing interest in the biomedical community to use uncertainty quantification (UQ) as part of a credibility assessment [76]. UQ has, however, been well established in other scientific domains for many decades [77] and involves calculating how uncertainty in the experimental data and model parameters contributes to uncertainty in the model outputs. A model that generates highly uncertain outputs is seen as less credible. Verification, validation, and UQ have been collectively referred to by many practitioners using the acronym VVUQ [58]. A recent article by Colebank et al., reviews elements of this topic [72].

Another criterion that can enhance a model's credibility is the provenance of the data used to build the model. Where did the data come from, and was it modified in some way? Table 1 gives a summary of some of the main criteria that can be used to access credibility. However, not all criteria are equally important, with validation and verification usually being key in this regard. The first four rules in the 10 rules devised by the Committee on Credible Practice of Modeling & Simulation emphasize all these aspects [54].

Attribute	Criteria
Validation	How well do the model outputs match reality?
Verification	Has the model been constructed without error; are the simulation algorithms correctly encoded and operating without error?
Uncertainty	Have the effects of uncertainty in the model outputs been reported?
Provenance	Can the data that was used to calibrate or validate the model be traced to its original source?
Annotation	Have the inputs and output of the model been well defined?
Assumptions	Have the assumptions used in the model been made explicit?
Purpose	Has the purpose of the model been adequately described?
Scope	Has the scope, that is, the space within which the model can be used, been specified?
Unbiased	Was the model calibrated on unbiased data? This depends on the scope of the model, but if a model is to be used across a diverse population, then clearly, the calibration data needs to be diverse.

Table 1 A range of criteria that can establish the credibility of a model.

In many cases, modelers may opt not to consider some or all of these criteria in their work, primarily due to the burden of having to do the checks. We, therefore, recommend automating as many of these criteria as possible. For example, verification of SBML-based models can be achieved using the BioSimulation resource [78]. Validation tests could be provided in a standard format as part of the software modeling code just as software engineers often provide validation tests as part of their distributions [79]. More difficult is including information about data provenance and model assumptions. However, the use of model standards such as SBML or CellML allow models to be annotated with this information. It is more difficult to address the availability of experimental metadata and under what conditions the model was validated. However, work by the synthetic biology community has made progress in this area with the development of protocols to describe an experiment [80]. This is an important area that needs further work, and these guidelines may stimulate further research.

When models have an explicit representation (e.g., SBML) as opposed to just a software implementation, analysis tools can do deep dives into the model to examine its biophysical assumptions. Also, more in-depth verification can be done on the explicit model representation, such as detecting errors in the formulation of biophysical laws [81]. MEMOTE [82] is a successful example of automated software that can do deep dives into genome-scale models to assess the quality of the model.

Credibility is widely used in software development through the commenting of code, using version control for provenance, and unit and system testing for validation [61]. Verification is achieved through in-depth checking of the software compilers and runtime systems.

6.2 Understandable

One of the aims of the scientific method is to gain an understanding of how the world works by proposing models and theories that can be tested. We understand theories (to paraphrase [83]) to be bodies of knowledge that are broad in scope. Chemical reaction theory or the central dogma in biology are examples of broad scope. In contrast, models are instantiations of theories and, as a result, are narrower in scope and often represent a particular biological process of interest. This includes computational models of metabolism, protein signaling networks, etc. Both models and theories are the most important outcome of the scientific method. They provide a way to rationalize a set of observations and make predictions about the future state of the system. However, the act of "understanding" [84] a model or theory is not an easy concept to define and may encompass a number of different aspects. In particular, how might one quantify "understanding"? Philosophically, de Regt and Dieks [84] define understanding as a given phenomenon if there exists a theory that describes the phenomenon. However, such definitions are hard to quantify. Instead, we will focus on measures that can be used to provide some level of confidence in how understandable a model is.

Biological systems, even small ones, are challenging to understand because of the nonlinear interactions among the components. The problem becomes even more acute as the systems we study grow larger. Biological systems often interact in complicated and nonlinear ways that result in emergent behaviors [85]. For example, no amount of understanding of the components of DNA – purines and pyrimidines, or at a lower-scale carbon and hydrogen atoms – would lead us to predict its complex structure or understand its biological role. The role emerges from the way the components are assembled.

How can we measure understanding? One way is to divide a model's components and attributes into levels of perceived importance. For example, we could understand different aspects of a model such as knowing a model's purpose, its components, the biophysical laws that describe how the components interact, its inputs, outputs, assumptions, and limitations. Figure 2, depicts a pyramid that organizes such a hierarchy of 'understanding'. At the base of the pyramid is the most rudimentary understanding; successive layers indicate increased levels of understanding. At the most rudimentary level (1), we want to know the purpose and objectives of the
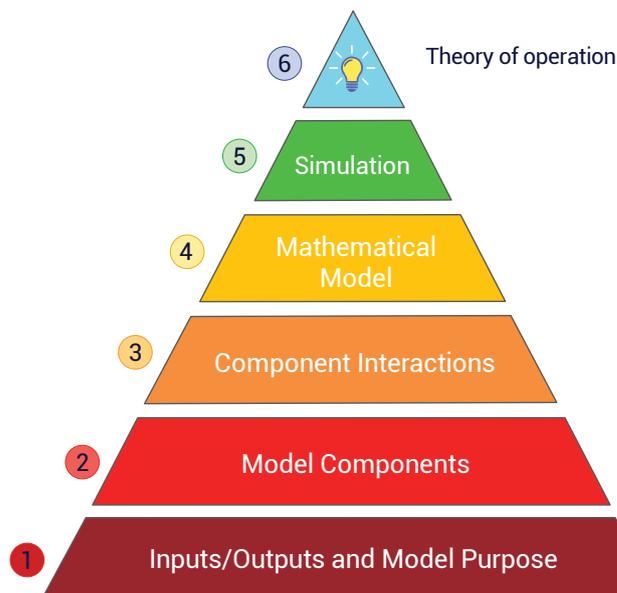
Fig. 2 Quantifying understanding through a hierarchy of criteria. The hierarchy begins with the simplest requirements, culminating with a general theory that explains the operation of the model. Not all hierarchical stages may be achieved in any given model.]

model as well as its inputs and outputs. Subsequent levels include the system components being modeled (2); the interactions between components that are modeled (3); a mathematical description of these interactions (4); a way to evaluate the mathematical model (e.g., solve a system of differential equations) (5); and finally, if possible, a general theory that explains the behavior of the model (6).

Technology is already available to assist in identifying model components (Level 2) and interactions (Level 3), as well as adding metadata to a model (Level 1). Such information can be provided through model annotations [86]. The mathematical model (Level 4) also includes the model's assumptions, which can be added as annotations using the SBO (Systems Biology Ontology) ontology [87]. For genome-scale metabolic models, the scientific community developed a list of detailed recommendations to annotate such models at Levels 1 to 4 following an extensive debate at their 2018 annual conference on constraint-based reconstruction and analysis (COBRA) [88]. Scientific models do not have to be mathematical, but for our purposes, most models are, and once a model is defined mathematically, it is often possible to convert the model into an executable form so that simulations can be carried out to generate predictions. Simulation information can be annotated using ontologies such as KiSAO (Kinetic Simulation Algorithm Ontology) [89]. The pinnacle of understanding is Level 6, which is the formulation of a theory that explains the behavior seen in the model.

Level 6 is the most difficult to quantify and involves explaining in terms of both fundamental and higher-order concepts how a given behavior emerges. Making this more difficult is that many biological systems show emergent behavior [85], where we observe

behavior that is not found in any individual component of the system. Even simple systems can reach this threshold. For example, a system of only two components and some non-linearity can show sustained energy dependent oscillations [39, 90]. Understanding the individual components is not sufficient to explain the origin of the oscillations [91] but requires a theory to help understand how, as a system, we observe oscillations. In this particular case, we need additional concepts, such as hysteresis and negative feedback, to understand the origin of the oscillations. For very complex systems, there may be multiple levels of abstraction that describe many levels of emergent behavior. Such abstractions are commonly used in electrical engineering and computer science. This is what makes it possible to engineer these highly complex systems. Reverse engineering abstraction layers in biology are, however, difficult [92, 93] due to the fact we are dealing with evolved systems that are not always as well structured as our own engineered systems. Unfortunately, no ontology exists to adequately annotate how a system operates. TEDDY (Terminology for the Description of Dynamics) is closest to such an ontology [87], which can be used to describe dynamics but does not, as yet, have the capacity to describe a theory of operation.

6.3 Reproducible

Reproducibility is a cornerstone of the scientific method, and over the last 10-15 years, much discussion has been devoted to this topic [94, 95]; mostly about the *lack* of reproducibility of many scientific results. Often, we think of wet lab experiments in biology as being difficult to reproducible because they are multifaceted and inherently variable. However, it has also been discovered that computational experiments are often not reproducible [15]. This is surprising given that computation involves well-defined and often deterministic procedures. One thing that has become clear is that community standards such as SBML have significantly improved the reproducibility of computational models by providing a machine-readable representation in a standard format [15, 96]. Moreover, recent evidence supports the opinion that reproducible models are more cited and more likely to be reused in subsequent studies [97]. Even if reproducibility is not a high priority for its own sake, verification, validation, and reuse of models require a model to be reproducible [98]. In recent years, we have seen the emergence of software tools and standards that offer support to assist modelers in creating reproducible models [81, 99–101]. One could even go as far as to say that in systems biology at least, the problem of reproducibility is solved [16]. We will not revisit the many issues and recommendations concerning the reproducibility of computational experiments, and readers are referred to the many publications that have appeared in recent years [16, 17, 27, 28, 52].

6.4 Extensible and reusable

Science builds on past efforts, standing on the proverbial shoulders of giants. This is no different when building computational models where past computational models can be enhanced, reused, and further validation applied. Unfortunately, many published

models cannot be easily reused [54]. This is because many models were published without an explicit representation of the model. Instead, the model is deeply embedded in a software implementation such as MATLAB, Python or Julia [102]. The software implementation adds many complexities (e.g., file interfaces and control logic for computing a solution), often in the form of multiple files with minimal documentation. Moreover, models deployed in this way are in a mathematical representation that loses considerable biological information. For example, when a biochemical pathway model is reduced to a set of differential equations the stoichiometric structure of the network is either lost or is difficult to reverse engineer. A further concern is that the mathematical representation greatly complicates the ability to query models to discover similar pathways, kinetics, and other characteristics. These considerations are some of the reasons why genome-scale models are published using SBML [19, 82] so as to preserve as much biological information as possible.

Hence, the primary concern with publishing a model solely as its software implementation is that it is difficult to reuse the model, either in part or in whole. For example, combining a model written in MATLAB with a model written in Python can be a costly exercise. Likewise, converting models from one format to another, for example, MATLAB to SBML, can also be error-prone and costly [29]. The use of standards such as SBML offers additional advantages, such as automated deconstruction [36] and reuse of models [103] through the use of model annotations or deep analysis using tools such as MEMOTE [82]. Models expressed in SBML are much easier to reuse or extend. When converted to a human-readable language like Antimony [104], reuse becomes even easier.

Other disciplines employ formats such as Modelica [105] or representations such as Simulink [106] to assist in reuse, but such techniques have not been widely used in developing biological models.

If executable code is used to represent models, then the model should ideally be partitioned into reusable program functions with ample documentation to illustrate how to reuse the model and what the various symbols in the mathematical equations represent.

In addition to technical aspects, licensing information is crucial for extensibility and reusability, as it defines if, how, and under what conditions models can be reused — only open and permissive licenses enable unrestricted reuse.

7 Recommended Requirements

The previous discussion provides a wide range of criteria that can be used to satisfy the CURE guidelines, and fulfilling all the requirements would be quite onerous. For most academic studies, it might be sufficient to meet a small number of criteria. For models that might be used in a clinical setting, it would be prudent to satisfy as many criteria as possible. Organizations such as the FDA (Food and Drug Administration) have already begun to issue guidelines for models used in medical devices [107]. Although these guidelines are not related to systems biology, such efforts point to a growing trend in the biomedical sciences.

For academic research, we propose a recommended set of requirements from each aspect of `CURE` that would be sufficient to significantly impact computational modeling. We provide a checklist in Figure 3, which also highlights the baseline requirements.

A key requirement is the need to develop standardized approaches to assess and communicate the extent to which a given model, or a modeling study, satisfies the recommended or baseline requirements. An example to consider is the approach taken by the CPMS working group to develop a rubric that considers the extent of outreach to various stakeholders in satisfying various credible practice guidelines [56]. Such evaluations, typically conducted as self-assessments by the respective study authors, can be shared with the community as part of supplementary material in published studies [108, 109]. Automation could greatly facilitate this assessment process, especially if provided in advance and used during model development.

7.1 Baseline Requirements

If all the recommended requirements are considered too onerous, it is possible to define a **baseline requirement**. These are the essential or foundational standards that are necessary for basic credibility, though they do not meet the full recommended requirements specified by `CURE`. The baseline requirement is similar in intent to the recent report [110] from the US National Academies, which emphasizes purpose, verification, validation, uncertainty quantification, and reproducibility, though their statement on reproducibility is vague. We include scope and model limitations into our baseline, which the National Academies documents do not explicitly mention, though it could be considered part of the statement of purpose. The baseline requirements are highlighted in Figure 3, and a more detailed summary is given in Table 2.

8 Conclusion

This paper introduces a set of principles for developing robust, credible biological models. These guidelines are meant to complement the existing but more generically applicable FAIR guidelines. As with FAIR, we propose a four-word moniker, `CURE`, that defines four core attributes of good modeling practice. These include credibility, understandability, reproducibility, and extensibility.

The need for `CURE` models is highlighted by recent interest in Biomedical Digital Twins, a technology that will rely on having robust models of biomedical systems [5, 110, 114], but it is clear that such guidelines would also benefit the wider biological modeling communities [115].

For credibility, we recommend the use of verification, validation, and uncertainty quantification; for understandability, we discuss the clarity of model descriptions and the importance of annotations; for reproducibility, our focus is standards and open science practices; and for extensibility, we emphasize open standards and the use of modular models. We outline recommended requirements for each guideline and propose a baseline level below the recommended requirements that is largely in alignment with the National Academies report [110].

Credibility (Baseline requirement):

1. Clearly define the objectives and scope of the modeling study, including the biological question being addressed and the specific hypotheses to be tested.

2. Where possible, verify the model by checking the model with other simulators. Use model-checking tools to identify errors in the model [61].

3. Validate the model against experimental data using accepted statistical procedures to assess model accuracy and predictive power.

4. Where possible, assess how sensitive the model is to uncertainty (UQ) in parameter estimates, model structure, inputs, and assumptions.

5. Clearly document the limitations of the model, including areas where assumptions may not hold or where uncertainties exist, to provide context for interpreting results and guiding future research.

Understandability:

1. Provide a representation of the model that is both machine-readable and human-readable. Ideally, this should be in an open community standard and be an explicit representation of the model that is not intertwined with control logic, file input/output, and other implementation details.

2. Provide comprehensive documentation that explains the model structure, equations, and parameter values, e.g., by following the suggestions by Carey et al. [88]. When using open community standards such as SBML or CellML, submit the models to a recognized model repository. If the model is expressed in a programming language, deposit your model code at repositories such as GitHub, BioModels [24], or ModelDB [42].

3. Where possible, annotate the model to clarify any ambiguous terminology. When using a programming language to express a model, use commenting to annotate the model.

4. Use consistent notation and terminology to ensure consistency and clarity in model descriptions. Where possible, follow common notations used in the community.

5. Try to document all assumptions made during model development, including simplifications, approximations, and parameter values, to provide transparency and facilitate reproducibility.

6. Try to provide clear graphical illustration of the model. If the model is a biochemical network then use machine-readable formats such as SBGN [111], preferably using a community modeling standard such as SBML Layout [112] and Render [113].

Reproducibility (Baseline requirement):

1. Follow established standards and guidelines for model development, such as the Systems Biology Markup Language (SBML) and Minimum Information Requested in the Annotation of Models (MIRIAM), to enhance interoperability and facilitate model sharing and exchange. If using executable code, make sure best practices for code development are used.

2. Embrace and promote open science practices by openly sharing publicly funded models, data, and code with the scientific community, promoting transparency, reproducibility, and collaboration.

Extensibility and Reuse:

1. Use open modeling standards where possible. If using executable code such as Python, separate the model code from the runtime code so that, in principle, the model can be reused by other Python users. Care should be taken when describing the components of the model as comments. One approach is to provide the model as a software function that can be called by other code. Try to use open-source licensing so that there are no restrictions on the reuse of the research.

2. If a model is represented using a modeling format such as SBML, reuse should be much easier since the model is expressed in biological terms. If the model is annotated then automated systems can be devised to automate the merging and disassembly of models into individual parts of portions for reuse.
3. Provide licensing information and use open and permissive licenses.

Table 2 Summary of Recommended and Baseline Requirements.

Recommended Requirements for CURE

Credibility	Understandability	Reproducibility	Extensibility
☐ Define objectives and purpose	☐ Provide the model code[1]	☐ Use community standards	☐ Use community modeling standards
☐ Verify the model	☐ Annotate the model	Or	Or
☐ Validate the model	☐ Document model assumptions	☐ Use containerization [2]	☐ Use modular source code so that model components can be reused. Documentation is crucial in this situation.
☐ Include uncertainty quantification	☐ Use consistent notation	Or	
☐ Document model limitations		☐ Use best practices when writing executable code, minimize dependencies	

1. This could be an open standard such as SBML, CellML, or program code such as Python.

2. For example the Docker platform, although it is not known how well such technologies age over time.

Fig. 3 A suggested check-list for CURE that could be used for research-based models. The baseline requirements are highlighted in shaded pink boxes. A score of one out of ten can be given based on the number of criteria met. For example, the baseline requirement will yield a score of 6/10. Note that if publishing models via source code, it is essential to specify the version number of the software platform as well as version numbers of any dependency packages that were used. The use of containerization platforms such as docker can sometimes help in these situations.

While all of the CURE principles are important, we wish to highlight credibility, the first principle. Credibility is the degree to which a model can be trusted when applied to a given problem. In a biomedical application where concern is with patients and their well-being, the trustworthiness of a predictive model is paramount. Although tangential to models used in systems biology, the FDA recently published [107] a guidance document on assessing the credibility of computational models but applied only to medical devices. Many of their recommendations relate to model verification and validation but also contained aspects related to model plausibility, which considers the plausibility of the governing equations, assumptions, and model parameters. The FDA document also emphasizes UQ for estimating uncertainty in the model outputs. These are considered foundational for credible modeling. However, these have not yet significantly infiltrated the biomedical modeling community. The recent National Academies [110] report on Digital Twins emphasizes the same points. The report also stresses the critical need for data and modeling standards to enable interoperability and reuse. As an initial effort to support VVUQ, the BioSimulations resource [78] provides a verification service where a given model can be run against multiple independent simulators to help verify the simulation engines.

Finally, automation is key to making the CURE guidelines practical to reduce the burden on practitioners and accelerate wider adoption. To be clear, by automation we don't mean automating model construction, although one could envisage such a scheme. Instead, we are referring to automating model checking as outlined by CURE once the model has been built.

By automation we do not mean automating model construction although one could envisage such Automation of these guidelines on models that are published in arbitrary programming languages will be almost impossible unless modern AI can in

some way be leveraged. More realistic is automation when models are distributed using open standards that have a strict formal structure and are designed to be computer-readable. One project that has been successful in automating model checks is MEMOTE [82]. This software is able to do deep dives into genome-scale metabolic models (GEMs) to assess the quality of the model. This is possible because GEMs are stored in computer-readable form (SBML). Other efforts underway include verification of rate laws (https://github.com/sys-bio/ratesb_python) and, as yet, an unpublished service at (https://github.com/biosimulators/bsvs) that can be used to verify a SBML model by running a given model against multiple simulators. For model calibration, PEtab [116] can be used to describe the model calibration workflow, including uncertainty quantification. This allows model calibration to be reproducible and therefore automated. The PEtab community is currently developing version 2.0, which will expand its capabilities even further. PEtab will be an important component of an automated system. However, other open standards remain to be developed. In particular, automated model validation will be a key requirement to automate CURE guidelines.

Contributions

HMS conceived the study. HMS, EG, JHG, PH, BRJ, EM, GDB, PMB, AD, JF, TEG, CJM, RV, JH, CFL, WWL, HSW, AM, and PM wrote and edited the manuscript. The remaining authors, EA, MLB, JLH, AH, BEJ, DPN, LPS, FTB, JRF, SF, FF, JAG, TH, HH, SF, PI, SMK, MK, RL, LML, LM, JGM, AN, DDVB, BGO, AAP, EMQ, NR, JMR, SS, JCS, FS, TJS, JS, JLS, DW, IM read and approved the manuscript content.

Competing Interests

The authors declare no competing interests.

References

[1] Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L.B., Bourne, P.E., *et al.*: The FAIR guiding principles for scientific data management and stewardship. Scientific Data **3**(1), 1–9 (2016)

[2] Ravi, N., Chaturvedi, P., Huerta, E., Liu, Z., Chard, R., Scourtas, A., Schmidt, K., Chard, K., Blaiszik, B., Foster, I.: FAIR principles for AI models with a practical application for accelerated high energy diffraction microscopy. Scientific Data **9**(1), 657 (2022)

[3] Shiferaw, K.B., Balaur, I., Welter, D., Waltemath, D., Zeleke, A.A.: Califrame: a proposed method of calibrating reporting guidelines with fair principles to foster reproducibility of ai research in medicine. JAMIA open **7**(4), 105 (2024)

[4] Juarez, M.G., Botti, V.J., Giret, A.S.: Digital twins: Review and challenges. Journal of Computing and Information Science in Engineering **21**(3), 030802 (2021)

[5] Laubenbacher, R., Adler, F., An, G., Castiglione, F., Eubank, S., Fonseca, L.L., Glazier, J., Helikar, T., Jett-Tilton, M., Kirschner, D., *et al.*: Toward mechanistic medical digital twins: some use cases in immunology. Frontiers in Digital Health **6**, 1349595 (2024)

[6] Michael, C.T., Almohri, S.A., Linderman, J.J., Kirschner, D.E.: A framework for multi-scale intervention modeling: virtual cohorts, virtual clinical trials, and model-to-model comparisons. Front. Syst. Biol. **3** (2023)

[7] Dill, K.A., MacCallum, J.L.: The protein-folding problem, 50 years on. Science **338**(6110), 1042–1046 (2012)

[8] McCammon, J.A., Gelin, B.R., Karplus, M.: Dynamics of folded proteins. nature **267**(5612), 585–590 (1977)

[9] Laubenbacher, R., Niarakis, A., Helikar, T., An, G., Shapiro, B., Malik-Sheriff, R.S., Sego, T.J., Knapp, A., Macklin, P., Glazier, J.A.: Building digital twins of the human immune system: toward a roadmap. npj Digital Medicine **5**(1), 64 (2022) https://doi.org/10.1038/s41746-022-00610-z

[10] Niarakis, A., Laubenbacher, R., An, G., Ilan, Y., Fisher, J., Flobak, Å., Reiche, K., Rodríguez Martínez, M., Geris, L., Ladeira, L., Veschini, L., Blinov, M.L., Messina, F., Fonseca, L.L., Ferreira, S., Montagud, A., Noël, V., Marku, M., Tsirvouli, E., Torres, M.M., Harris, L.A., Sego, T.J., Cockrell, C., Shick, A.E., Balci, H., Salazar, A., Rian, K., Hemedan, A.A., Esteban-Medina, M., Staumont, B., Hernandez-Vargas, E., Martis B, S., Madrid-Valiente, A., Karampelesis, P., Sordo Vieira, L., Harlapur, P., Kulesza, A., Nikaein, N., Garira, W., Malik Sheriff, R.S., Thakar, J., Tran, V.D.T., Carbonell-Caballero, J., Safaei, S., Valencia, A., Zinovyev, A., Glazier, J.A.: Immune digital twins for complex human pathologies: applications, limitations, and challenges. npj Systems Biology and Applications **10**(1), 141 (2024) https://doi.org/10.1038/s41540-024-00450-5

[11] Karr, J.R., Sanghvi, J.C., Macklin, D.N., Gutschow, M.V., Jacobs, J.M., Bolival, B., Assad-Garcia, N., Glass, J.I., Covert, M.W.: A whole-cell computational model predicts phenotype from genotype. Cell **150**(2), 389–401 (2012)

[12] Macklin, D.N., Ruggero, N.A., Covert, M.W.: The future of whole-cell modeling. Current opinion in biotechnology **28**, 111–115 (2014)

[13] Waltemath, D., Karr, J.R., Bergmann, F.T., Chelliah, V., Hucka, M., Krantz, M., Liebermeister, W., Mendes, P., Myers, C.J., Pir, P., *et al.*: Toward community standards and software for whole-cell modeling. IEEE Transactions on Biomedical Engineering **63**(10), 2007–2014 (2016)

[14] Waltemath, D., Wolkenhauer, O.: How modeling standards, software, and initiatives support reproducibility in systems biology and systems medicine. IEEE Transactions on Biomedical Engineering **63**(10), 1999–2006 (2016)

[15] Tiwari, K., Kananathan, S., Roberts, M.G., Meyer, J.P., Sharif Shohan, M.U., Xavier, A., Maire, M., Zyoud, A., Men, J., Ng, S., *et al.*: Reproducibility in systems biology modelling. Molecular Systems Biology **17**(2), 9982 (2021)

[16] Porubsky, V.L., Goldberg, A.P., Rampadarath, A.K., Nickerson, D.P., Karr, J.R., Sauro, H.M.: Best practices for making reproducible biochemical models. Cell systems **11**(2), 109–120 (2020)

[17] Shin, J., Porubsky, V., Carothers, J., Sauro, H.M.: Standards, dissemination, and best practices in systems biology. Current opinion in biotechnology **81**, 102922 (2023)

[18] Vieira, L.S., Laubenbacher, R.C.: Computational models in systems biology: standards, dissemination, and best practices. Current opinion in biotechnology **75**, 102702 (2022)

[19] Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A., *et al.*: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics **19**(4), 524–531 (2003)

[20] Keating, S.M., Waltemath, D., König, M., Zhang, F., Dräger, A., Chaouiya, C., Bergmann, F.T., Finney, A., Gillespie, C.S., Helikar, T., Hoops, S., Malik-Sheriff, R.S., Moodie, S.L., Moraru, I.I., Myers, C.J., Naldi, A., Olivier, B.G., Sahle, S., Schaff, J.C., Smith, L.P., Swat, M.J., Thieffry, D., Watanabe, L., Wilkinson, D.J., Blinov, M.L., Begley, K., Faeder, J.R., Gómez, H.F., Hamm, T.M., Inagaki, Y., Liebermeister, W., Lister, A.L., Lucio, D., Mjolsness, E., Proctor, C.J., Raman, K., Rodriguez, N., Shaffer, C.A., Shapiro, B.E., Stelling, J., Swainston, N., Tanimura, N., Wagner, J., Meier-Schellersheim, M., Sauro, H.M., Palsson, B., Bolouri, H., Kitano, H., Funahashi, A., Hermjakob, H., Doyle, J.C., Hucka, M., Adams, R.R., Allen, N.A., Angermann, B.R., Antoniotti, M., Bader, G.D., Červený, J., Courtot, M., Cox, C.D., Dalle Pezze, P., Demir, E., Denney, W.S., Dharuri, H., Dorier, J., Drasdo, D., Ebrahim, A., Eichner, J., Elf, J., Endler, L., Evelo, C.T., Flamm, C., Fleming, R.M.T., Fröhlich, M., Glont, M., Gonçalves, E., Golebiewski, M., Grabski, H., Gutteridge, A., Hachmeister, D., Harris, L.A., Heavner, B.D., Henkel, R., Hlavacek, W.S., Hu, B., Hyduke, D.R., Jong, H., Juty, N., Karp, P.D., Karr, J.R., Kell, D.B., Keller, R., Kiselev, I., Klamt, S., Klipp, E., Knüpfer, C., Kolpakov, F., Krause, F., Kutmon, M., Laibe, C., Lawless, C., Li, L., Loew, L.M., Machne, R., Matsuoka, Y., Mendes, P., Mi, H., Mittag, F., Monteiro, P.T., Natarajan, K.N., Nielsen, P.M.F., Nguyen, T., Palmisano, A., Pettit, J.-B., Pfau, T., Phair, R.D., Radivoyevitch, T., Rohwer, J.M., Ruebenacker, O.A., Saez-Rodriguez, J., Scharm, M., Schmidt, H., Schreiber, F., Schubert, M., Schulte, R., Sealfon, S.C., Smallbone, K., Soliman, S., Stefan, M.I., Sullivan, D.P., Takahashi, K., Teusink, B., Tolnay, D., Vazirabad, I., Kamp, A.v., Wittig, U., Wrzodek, C., Wrzodek, F., Xenarios, I., Zhukova, A., Zucker, J.: SBML Level 3: an extensible format for the exchange and reuse of biological models. Molecular Systems Biology **16**(8), 9110 (2020) https://doi.org/10.15252/msb.20199110 https://www.embopress.org/doi/pdf/10.15252/msb.20199110

[21] Clerx, M., Cooling, M.T., Cooper, J., Garny, A., Moyle, K., Nickerson, D.P., Nielsen, P.M., Sorby, H.: CellML 2.0. Journal of Integrative Bioinformatics **17**(2-3), 20200021 (2020)

[22] Gleeson, P., Crook, S., Cannon, R.C., Hines, M.L., Billings, G.O., Farinella, M., Morse, T.M., Davison, A.P., Ray, S., Bhalla, U.S., *et al.*: NeuroML: a language for describing data driven models of neurons and networks with a high degree of biological detail. PLoS Computational Biology **6**(6), 1000815 (2010)

[23] Gu, C., Kim, G.B., Kim, W.J., Kim, H.U., Lee, S.Y.: Current status and applications of genome-scale metabolic models. Genome biology **20**, 1–18 (2019)

[24] Malik-Sheriff, R.S., Glont, M., Nguyen, T.V., Tiwari, K., Roberts, M.G., Xavier, A., Vu, M.T., Men, J., Maire, M., Kananathan, S., *et al.*: BioModels—15 years of sharing computational models in life science. Nucleic Acids Research **48**(D1), 407–415 (2020)

[25] King, Z.A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J.A., Ebrahim, A., Palsson, B.O., Lewis, N.E.: BiGG models: A platform for integrating, standardizing and sharing genome-scale models. Nucleic Acids Research **44**(D1), 515–522 (2016)

[26] Arkin, A.P., Cottingham, R.W., Henry, C.S., Harris, N.L., Stevens, R.L., Maslov, S., Dehal, P., Ware, D., Perez, F., Canon, S., *et al.*: KBase: the united states department of energy systems biology knowledgebase. Nature Biotechnology **36**(7), 566–569 (2018)

[27] Blinov, M.L., Gennari, J.H., Karr, J.R., Moraru, I.I., Nickerson, D.P., Sauro, H.M.: Practical resources for enhancing the reproducibility of mechanistic modeling in systems biology. Current Opinion in Systems Biology **27**, 100350 (2021)

[28] Porubsky, V., Smith, L., Sauro, H.M.: Publishing reproducible dynamic kinetic models. Briefings in Bioinformatics **22**(3), 152 (2021)

[29] Erdem, C., Mutsuddy, A., Bensman, E.M., Dodd, W.B., Saint-Antoine, M.M., Bouhaddou, M., Blake, R.C., Gross, S.M., Heiser, L.M., Feltus, F.A., *et al.*: A scalable, open-source implementation of a large-scale mechanistic model for single cell proliferation and death signaling. Nature Communications **13**(1), 3555 (2022)

[30] Chaouiya, C., Bérenguier, D., Keating, S.M., Naldi, A., van Iersel, M.P., Rodriguez, N., Dräger, A., Büchel, F., Cokelaer, T., Kowal, B., Wicks, B., Gonçalves, E., Dorier, J., Page, M., Monteiro, P.T., von Kamp, A., Xenarios, I., de Jong, H., Hucka, M., Klamt, S., Thierffry, D., Le Novère, N., Saez-Rodriguez, J., Helikar, T.: SBML Qualitative Models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. BMC Systems Biology **7**(1), 135 (2013) https://doi.org/10.1186/1752-0509-7-135 1309.1910

[31] Niarakis, A., Kuiper, M., Ostaszewski, M., Malik Sheriff, R.S., Casals-Casas, C., Thieffry, D., Freeman, T.C., Thomas, P., Touré, V., Noël, V., Stoll, G., Saez-Rodriguez, J., Naldi, A., Oshurko, E., Xenarios, I., Soliman, S., Chaouiya, C., Helikar, T., Calzone, L.: Setting the basis of best practices and standards for curation and annotation of logical models in biology—highlights of the [bc]2 2019 colomoto/sysmod workshop. Briefings in Bioinformatics **22**(2), 1848–1859

(2020) https://doi.org/10.1093/bib/bbaa046

[32] Blais, E.M., Chavali, A.K., Papin, J.A.: Linking genome-scale metabolic modeling and genome annotation. Systems Metabolic Engineering: Methods and Protocols, 61–83 (2013)

[33] Passi, A., Tibocha-Bonilla, J.D., Kumar, M., Tec-Campos, D., Zengler, K., Zuniga, C.: Genome-scale metabolic modeling enables in-depth understanding of big data. Metabolites **12**(1), 14 (2021)

[34] Neal, M.L., König, M., Nickerson, D., Mısırlı, G., Kalbasi, R., Dräger, A., Atalag, K., Chelliah, V., Cooling, M.T., Cook, D.L., Crook, S., de Alba, M., Friedman, S.H., Garny, A., Gennari, J.H., Gleeson, P., Golebiewski, M., Hucka, M., Juty, N., Myers, C., Olivier, B.G., Sauro, H.M., Scharm, M., Snoep, J.L., Touré, V., Wipat, A., Wolkenhauer, O., Waltemath, D.: Harmonizing semantic annotations for computational models in biology. Briefings in Bioinformatics **20**(2), 540–550 (2018) https://doi.org/10.1093/bib/bby087 https://academic.oup.com/bib/article-pdf/20/2/540/28834058/bby087.pdf

[35] Novère, N.L., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J., Crampin, E.J., Halstead, M., Klipp, E., Mendes, P., *et al.*: Minimum information requested in the annotation of biochemical models (miriam). Nature Biotechnology **23**(12), 1509–1515 (2005)

[36] Neal, M.L., Galdzicki, M., Gallimore, J., Sauro, H.M.: A c library for retrieving specific reactions from the biomodels database. Bioinformatics **30**(1), 129–130 (2014)

[37] Größler, A., Milling, P.: Inductive and deductive system dynamics modeling. In: The 2007 International Conference of the System Dynamics Society (2007)

[38] Ingalls, B.P.: Mathematical Modeling in Systems Biology: an Introduction. MIT press, 255 Main Street, NE18. Floor 9. Cambridge, MA 02142 (2013)

[39] Sauro, H.M.: Systems Biology: Introduction to Pathway Modeling. Ambrosius Publishing, Seattle (2020)

[40] Kim, O.D., Rocha, M., Maia, P.: A review of dynamic modeling approaches and their application in computational strain optimization for metabolic engineering. Frontiers in Microbiology **9**, 378885 (2018)

[41] Peters, M., Eicher, J.J., Van Niekerk, D.D., Waltemath, D., Snoep, J.L.: The jws online simulation database. Bioinformatics **33**(10), 1589–1590 (2017)

[42] McDougal, R.A., Morse, T.M., Carnevale, T., Marenco, L., Wang, R., Migliore, M., Miller, P.L., Shepherd, G.M., Hines, M.L.: Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience. Journal

of computational neuroscience **42**, 1–10 (2017)

[43] Choi, J.-S., Waxman, S.G.: Physiological interactions between nav1. 7 and nav1. 8 sodium channels: a computer simulation study. Journal of Neurophysiology **106**(6), 3173–3184 (2011)

[44] Lopez, C.F., Muhlich, J.L., Bachman, J.A., Sorger, P.K.: Programming biological models in python using pysb. Molecular systems biology **9**(1), 646 (2013)

[45] Brandl, G.: Sphinx documentation. URL http://sphinx-doc. org/sphinx. pdf (2021)

[46] Plank, G., Loewe, A., Neic, A., Augustin, C., Huang, Y.-L., Gsell, M.A., Karabelas, E., Nothstein, M., Prassl, A.J., Sánchez, J., *et al.*: The opencarp simulation environment for cardiac electrophysiology. Computer methods and Programs in Biomedicine **208**, 106223 (2021)

[47] Britten, R.D., Christie, G.R., Little, C., Miller, A.K., Bradley, C., Wu, A., Yu, T., Hunter, P., Nielsen, P.: Fieldml, a proposed open standard for the physiome project for mathematical model representation. Medical & biological engineering & computing **51**, 1191–1207 (2013)

[48] Schaff, J.C., Lakshminarayana, A., Murphy, R.F., Bergmann, F.T., Funahashi, A., Sullivan, D.P., Smith, L.P.: SBML level 3 package: spatial processes, version 1, release 1. Journal of Integrative Bioinformatics **20**(1), 20220054 (2023)

[49] Another organization devoted to the promotion of curation practices of research compendia also uses the acronym CURE:"Curating for Reproducibility". However, there is little overlap between the two usages. There is also a UNESCO CuRe curriculum, but this is also unrelated.

[50] Niarakis, A., Waltemath, D., Glazier, J., Schreiber, F., Keating, S.M., Nickerson, D., Chaouiya, C., Siegel, A., Noël, V., Hermjakob, H., Helikar, T., Soliman, S., Calzone, L.: Addressing barriers in comprehensiveness, accessibility, reusability, interoperability and reproducibility of computational models in systems biology. Briefings in Bioinformatics **23**(4), 212 (2022) https://doi.org/10.1093/bib/bbac212

[51] Yilmaz, L., Liu, B.: Model credibility revisited: Concepts and considerations for appropriate trust. Journal of Simulation **16**(3), 312–325 (2022)

[52] Tatka, L.T., Smith, L.P., Hellerstein, J.L., Sauro, H.M.: Adapting modeling and simulation credibility standards to computational systems biology. Journal of Translational Medicine **21**(1), 501 (2023)

[53] Schlesinger, S.: Terminology for model credibility. Simulation **32**(3), 103–104

(1979)

[54] Erdemir, A., Mulugeta, L., Ku, J.P., Drach, A., Horner, M., Morrison, T.M., Peng, G.C., Vadigepalli, R., Lytton, W.W., Myers Jr, J.G.: Credible practice of modeling and simulation in healthcare: ten rules from a multidisciplinary perspective. Journal of Translational Medicine **18**(1), 369 (2020)

[55] Musuamba, F., Skottheim Rusten, I., Lesage, R., *et al.*: Scientific and regulatory evaluation of mechanistic in silico drug and disease models in drug development: building model credibility. CPT Pharmacometrics and Systems Pharmacology **10**, 804–825 (2021)

[56] Vadigepalli, R., Manchel, A., Erdemir, A., Mulugeta, L., Ku, J.P., Rego, B.V., Horner, M., Lytton, W.W., Myers, J.G.: A rubric for assessing conformance to the ten rules for credible practice of modeling and simulation in healthcare. medRxiv, 2024–10 (2024)

[57] Thacker, B., Doebling, S., Hemez, F., Anderson, M., Pepin, J., Rodriguez, E.: Concepts of model verification and validation (No. LA-14167). Los Alamos National Lab., Los Alamos, NM (US) (2004)

[58] Council, N.R., Engineering, D., Sciences, P., Mathematical Sciences, B., Applications, T., Verification, C., Quantification, U.: Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification. National Academies Press, ??? (2012)

[59] Bergmann, F.T., Sauro, H.M.: Comparing simulation results of SBML capable simulators. Bioinformatics **24**(17), 1963–1965 (2008)

[60] Hucka, M., Independent, A.F., He, E., Keating, S.: SBML Test Suite (2022). https://doi.org/10.5281/zenodo.1112521

[61] Hellerstein, J.L., Gu, S., Choi, K., Sauro, H.M.: Recent advances in biomedical simulations: a manifesto for model engineering. F1000Research **8** (2019)

[62] Mendes, P., Kell, D.: Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics **14**(10), 869–883 (1998)

[63] Marino, S., Hogue, I.B., Ray, C.J., Kirschner, D.E.: A methodology for performing global uncertainty and sensitivity analysis in systems biology. Journal of theoretical biology **254**(1), 178–196 (2008)

[64] Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J.A., Blom, J.G.: Systems biology: parameter estimation for biochemical models. The FEBS Journal **276**(4), 886–902 (2009)

[65] Dräger, A., Kronfeld, M., Ziller, M.J., Supper, J., Planatscher, H., Magnus, J.B., Oldiges, M., Kohlbacher, O., Zell, A.: Modeling metabolic networks in *C. glutamicum*: a comparison of rate laws in combination with various parameter optimization strategies. BMC Systems Biology **3**(5), 5 (2009) https://doi.org/10.1186/1752-0509-3-5

[66] Kirk, P., Thorne, T., Stumpf, M.P.: Model selection in systems and synthetic biology. Current opinion in biotechnology **24**(4), 767–774 (2013)

[67] Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C.P., Stumpf, M.P.: A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. Nature protocols **9**(2), 439–456 (2014)

[68] Gábor, A., Banga, J.R.: Robust and efficient parameter estimation in dynamic models of biological systems. BMC Systems Biology **9**, 1–25 (2015)

[69] Mitra, E.D., Hlavacek, W.S.: Parameter estimation and uncertainty quantification for systems biology models. Current opinion in systems biology **18**, 9–18 (2019)

[70] Read, M.N., Alden, K., Timmis, J., Andrews, P.S.: Strategies for calibrating models of biology. Briefings in Bioinformatics **21**(1), 24–35 (2020)

[71] Wieland, F.-G., Hauber, A.L., Rosenblatt, M., Tönsing, C., Timmer, J.: On structural and practical identifiability. Current Opinion in Systems Biology **25**, 60–69 (2021)

[72] Colebank, M.J., Oomen, P.A., Witzenburg, C.M., Grosberg, A., Beard, D.A., Husmeier, D., Olufsen, M.S., Chesler, N.C.: Guidelines for mechanistic modeling and analysis in cardiovascular research. American Journal of Physiology-Heart and Circulatory Physiology (2024)

[73] Villaverde, A.F., Pathirana, D., Fröhlich, F., Hasenauer, J., Banga, J.R.: A protocol for dynamic model calibration. Briefings in Bioinformatics **23**(1), 387 (2021) https://doi.org/10.1093/bib/bbab387 https://academic.oup.com/bib/article-pdf/23/1/bbab387/42230172/bbab387.pdf

[74] Touré, V., Flobak, Å., Niarakis, A., Vercruysse, S., Kuiper, M.: The status of causality in biological databases: data resources and data retrieval possibilities to support logical modeling. Briefings in Bioinformatics **22**(4), 390 (2020) https://doi.org/10.1093/bib/bbaa390

[75] Hall, B.A., Niarakis, A.: Data integration in logic-based models of biological mechanisms. Current Opinion in Systems Biology **28**, 100386 (2021) https://doi.org/10.1016/j.coisb.2021.100386

[76] Viceconti, M., Pappalardo, F., Rodriguez, B., Horner, M., Bischoff, J., Tshinanu, F.M.: In silico trials: Verification, validation and uncertainty quantification of predictive models used in the regulatory evaluation of biomedical products. Methods **185**, 120–127 (2021)

[77] Booker, J.M., Ross, T.J.: An evolution of uncertainty assessment and quantification. Scientia Iranica **18**(3), 669–676 (2011)

[78] 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 Research **50**(W1), 108–114 (2022)

[79] unittest — Unit testing framework. https://docs.python.org/3/library/unittest.html

[80] 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 Journal on Emerging Technologies in Computing Systems **19**(3), 1–21 (2023)

[81] Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI—a complex pathway simulator. Bioinformatics **22**(24), 3067–3074 (2006)

[82] 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.*: MEMOTE for standardized genome-scale metabolic model testing. Nature Biotechnology **38**(3), 272–276 (2020)

[83] Fried, E.I.: Theories and models: What they are, what they are for, and what they are about. Psychological Inquiry **31**(4), 336–344 (2020)

[84] Regt, D., W, H.: Understanding Scientific Understanding. Oxford University Press, Oxford (2017)

[85] Nicolis, G., Prigogine, I.: Exploring complexity: An introduction (1989)

[86] Le Novère, N.: Model storage, exchange and integration. BMC Neuroscience **7**, 1–9 (2006)

[87] Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., *et al.*: Controlled vocabularies and semantics in systems biology. Molecular Systems Biology **7**(1), 543 (2011)

[88] Carey, M.A., Dräger, A., Beber, M.E., Papin, J.A., Yurkovich, J.T.: Community

standards to facilitate development and address challenges in metabolic modeling. Molecular Systems Biology **16**(8), 9235 (2020) https://doi.org/10.15252/msb.20199235

[89] Zhukova, A., Waltemath, D., Juty, N., Laibe, C., Le Novère, N.: Kinetic simulation algorithm ontology. Nature Precedings, 1–1 (2011)

[90] Kholodenko, B.N.: Cell-signalling dynamics in time and space. Nature reviews Molecular cell biology **7**(3), 165–176 (2006)

[91] Reijenga, K.A., Megen, Y.M., Kooi, B.W., Bakker, B.M., Snoep, J.L., Verseveld, H.W., Westerhoff, H.V.: Yeast glycolytic oscillations that are not controlled by a single oscillophore: a new definition of oscillophore strength. Journal of Theoretical Biology **232**(3), 385–398 (2005)

[92] Csete, M.E., Doyle, J.C.: Reverse engineering of biological complexity. Science **295**(5560), 1664–1669 (2002)

[93] Andrews, S.S., Wiley, H.S., Sauro, H.M.: Design patterns of biological cells. BioEssays, 2300188 (2024)

[94] Baker, M.: 1,500 scientists lift the lid on reproducibility. Nature **533**(7604) (2016)

[95] Plesser, H.E.: Reproducibility vs. replicability: a brief history of a confused terminology. Frontiers in neuroinformatics **11**, 76 (2018)

[96] Mendes, P.: Reproducible research using biomodels. Bulletin of Mathematical Biology **80**(12), 3081–3087 (2018)

[97] Höpfl, S., Pleiss, J., Radde, N.E.: Bayesian estimation reveals that reproducible models in systems biology get more citations. Scientific Reports **13**(1), 2695 (2023)

[98] Coveney, P.V., Groen, D., Hoekstra, A.G.: Reliability and reproducibility in computational science: Implementing validation, verification and uncertainty quantification in silico. The Royal Society Publishing (2021)

[99] Moraru, I.I., Schaff, J.C., Slepchenko, B.M., Blinov, M., Morgan, F., Lakshminarayana, A., Gao, F., Li, Y., Loew, L.M.: Virtual Cell modelling and simulation software environment. IET systems biology **2**(5), 352–362 (2008)

[100] Choi, K., Medley, J.K., König, M., Stocking, K., Smith, L., Gu, S., Sauro, H.M.: Tellurium: an extensible python-based modeling environment for systems and synthetic biology. Biosystems **171**, 74–79 (2018)

[101] Medley, J.K., Choi, K., König, M., Smith, L., Gu, S., Hellerstein, J., Sealfon, S.C., Sauro, H.M.: Tellurium notebooks—an environment for reproducible

dynamical modeling in systems biology. PLoS Computational Biology **14**(6), 1006220 (2018)

[102] Shichkova, P., Coggan, J.S., Kanari, L., Boci, E., Favreau, C., Antonel, S.M., Keller, D., Markram, H.: Breakdown and repair of metabolism in the aging brain. Frontiers in Science **3**, 1441297 (2025)

[103] Neal, M.L., Cooling, M.T., Smith, L.P., Thompson, C.T., Sauro, H.M., Carlson, B.E., Cook, D.L., Gennari, J.H.: A reappraisal of how to build modular, reusable models of biological systems. PLoS Computational Biology **10**(10), 1003849 (2014)

[104] Smith, L.P., Bergmann, F.T., Chandran, D., Sauro, H.M.: Antimony: a modular model definition language. Bioinformatics **25**(18), 2452–2454 (2009)

[105] Mattsson, S.E., Elmqvist, H., Otter, M.: Physical system modeling with modelica. Control Engineering Practice **6**(4), 501–510 (1998)

[106] Dabney, J.B., Harman, T.L.: Mastering Simulink vol. 230. Pearson/Prentice Hall Upper Saddle River, New Jersey (2004)

[107] Food and Drug Administration, et al.: Assessing the credibility of computational modeling and simulation in medical device submissions. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/assessing-credibility-computational-modeling-and-simulation-medical-device-submissions (2021)

[108] Verma, A., Antony, A.N., Ogunnaike, B.A., Hoek, J.B., Vadigepalli, R.: Causality analysis and cell network modeling of spatial calcium signaling patterns in liver lobules. Frontiers in physiology **9**, 1377 (2018)

[109] Gee, M.M., Lenhoff, A.M., Schwaber, J.S., Ogunnaike, B.A., Vadigepalli, R.: Closed-loop modeling of central and intrinsic cardiac nervous system circuits underlying cardiovascular control. AIChE Journal **69**(4), 18033 (2023)

[110] National Academies of Sciences, Engineering and Medicine, et al.: Foundational research gaps and future directions for digital twins. https://nap.nationalacademies.org/catalog/26894/foundational-research-gaps-and-future-directions-for-digital-twins (2023)

[111] Novère, N.L., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., *et al.*: The systems biology graphical notation. Nature biotechnology **27**(8), 735–741 (2009)

[112] Gauges, R., Rost, U., Sahle, S., Wegner, K.: A model diagram layout extension for sbml. Bioinformatics **22**(15), 1879–1885 (2006)

[113] Bergmann, F.T., Keating, S.M., Gauges, R., Sahle, S., Wengler, K.: Sbml level 3 package: render, version 1, release 1. Journal of Integrative Bioinformatics **15**(1), 20170078 (2018)

[114] Bhagirath, P., Strocchi, M., Bishop, M.J., Boyle, P.M., Plank, G.: From bits to bedside: Entering the age of digital twins in cardiac electrophysiology. Europace **in press** (2024)

[115] Karr, J., Malik-Sheriff, R., Osborne, J., Gonzalez-Parra, G., Forgoston, E., Bowness, R., Liu, Y., Thompson, R., Garira, W., Barhak, J., et al.: Model integration in computational biology: the role of reproducibility, credibility and utility. front. Syst. Biol **2**(10.3389) (2022)

[116] Schmiester, L., Schälte, Y., Bergmann, F.T., Camba, T., Dudkin, E., Egert, J., Fröhlich, F., Fuhrmann, L., Hauber, A.L., Kemmer, S., *et al.*: Petab—interoperable specification of parameter estimation problems in systems biology. PLoS computational biology **17**(1), 1008646 (2021)