

Practical resources for enhancing the reproducibility of mechanistic modeling in systems biology

Michael L. Blinov^{1,a}, John H. Gennari^{2,a}, Jonathan R. Karr^{3,4,a},
 Ion I. Moraru^{1,a}, David P. Nickerson^{5,a} and Herbert M. Sauro^{6,a}

Abstract

Although reproducibility is a core tenet of the scientific method, it remains challenging to reproduce many results. Tiwari et al. recently found that they could only repeat 50% of published simulation results in systems biology. We identified several resources that investigators can leverage to make their research more accessible, executable, and comprehensible by others. Enhanced reproducibility would accelerate the development of more sophisticated models that could inform precision medicine and synthetic biology.

Addresses

¹Center for Cell Analysis and Modeling, UConn Health, Farmington, CT, USA

²Department of Biomedical Informatics and Medical Education, University of Washington, Seattle, WA, USA

³Icahn Institute for Data Science and Genomic Technology, Icahn School of Medicine at Mount Sinai, New York, NY, USA

⁴Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY, USA

⁵Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

⁶Department of Bioengineering, University of Washington, Seattle, WA, USA

Corresponding author: Sauro, Herbert M (hsauro@uw.edu)

^a All the authors contributed equally to this work.

Current Opinion in Systems Biology 2021, 27:100350

This review comes from a themed issue on **Mathematical Modelling (2021)**

Edited by **Stacey Deleria Finley** and **Vassily Hatzimanikatis**

For complete overview of the section, please refer the article collection - [Mathematical Modelling \(2021\)](#)

Available online 10 June 2021

<https://doi.org/10.1016/j.coisb.2021.06.001>

2452-3100/© 2021 Elsevier Ltd. All rights reserved.

Keywords

Systems biology, Mechanistic modeling, Repeatability, Reproducibility, Reusability.

Introduction

One of the central pillars of the scientific method is *reproducibility* [1], the ability of independent researchers to *reproduce* results *de novo* without the aid of their original investigators or their hardware and software. We believe that such reproducibility would accelerate

computational science by making it easier for peer reviewers to quality control reported results and by making it easier for investigators build upon reported results. In particular, integrating models of multiple biological subsystems into more comprehensive models of entire cells and organisms will only be feasible if there are high-quality components that are accessible and reusable. Although attaining this degree of reproducibility is challenging, we believe that computational scientists have access to all of the raw ingredients needed to report *repeatable* results that other researchers can recreate using the same data files, software tools, and a similar computational environment.

Despite the importance of repeatability, many computational results are not repeatable. For example, Tiwari et al. [2] recently found that they could only repeat 50% of simulation results reported in systems biology. Similar concerns have been reported in bioinformatics [3] and other computational domains [4–6]. Taken together, poor repeatability is currently an endemic problem to computational research.

To enable the collaboration needed to achieve the more comprehensive and more predictive models required for synthetic biology and precision medicine, we believe that each systems biology simulation result should, at a minimum, be repeatable by independent investigators using high-level descriptions of the models and simulations that produced the result and the same software that produced the result. Ideally, each step of the workflow responsible for each simulation result should be accessible and repeatable (Figure 1) [7]. This includes the data and knowledge used to inform the experiment; the models involved in the experiment and the processes used to construct and calibrate the models; the simulations involved in the experiment and their results; and the processes used to validate these models and simulations.

Here, we outline the key information needed to repeat each step of the typical systems biology simulation workflow and provide practical recommendations for conducting each step repeatably. Figure 1 summarizes our recommendations on top of a depiction of the typical systems biology simulation workflow. In the interest of providing modelers actionable recommendations, we

focus on the aspects of systems biology modeling and simulation, such as forward numerical simulation, which we believe modelers can conduct repeatably using readily available tools. For further discussion about aspects of modeling whose reproducibility remains in its infancy, such as the calibration and static analysis of models, we refer the reader to a recent more extensive review [8]. In the interest of brevity, our recommendations also focus on popular tools which we believe meet the needs of the largest number of modelers. More comprehensive lists of tools are available at <https://cellml.org>, <https://petab.readthedocs.io>, <http://sbml.org>, and <http://sed-ml/>.

Toward repeatability and reproducibility

Repeating experimental data and knowledge: popular data formats

Because computational models in systems biology are typically based on experimental data and knowledge, the ability to repeat simulation results requires this information to be published, ideally in a form that is both readily understandable by other investigators and machine-readable. In our opinion, there are many good formats for sharing different types of data and knowledge in different contexts. For example, the XLSX format is often a good choice for publishing small datasets because most researchers are familiar with the format and XLSX files are readily readable by both humans (such as with Excel, as well as with free, open-source tools such as LibreOffice and OpenOffice) and machines (e.g. using the open-source pandas Python package [9]). The comma-separated values (CSVs) format can be a good choice when a group of investigators need to use a source code management system such as Git to iteratively develop a dataset and track and merge their changes. Where possible, we recommend that investigators use formats such as HDF5 [10], xarray [11], and Zarr (<https://zarr.readthe>) that have greater capabilities to capture critical meta-data such as information about the semantic meaning of

the data, how it was processed, and the assumptions that the processing employed. HDF5, xarray, and Zarr are particularly well-suited for datasets that are large, hierarchical, and/or multidimensional. RDF (Resource Description Framework) -based formats such as BioPAX [12] can be good choices for qualitative and relational information, such as information about transcription factor binding sites and protein–protein interactions. Whichever format investigators choose, it is critical to include sufficient metadata for other investigators to be able to understand and meaningfully reuse the data.

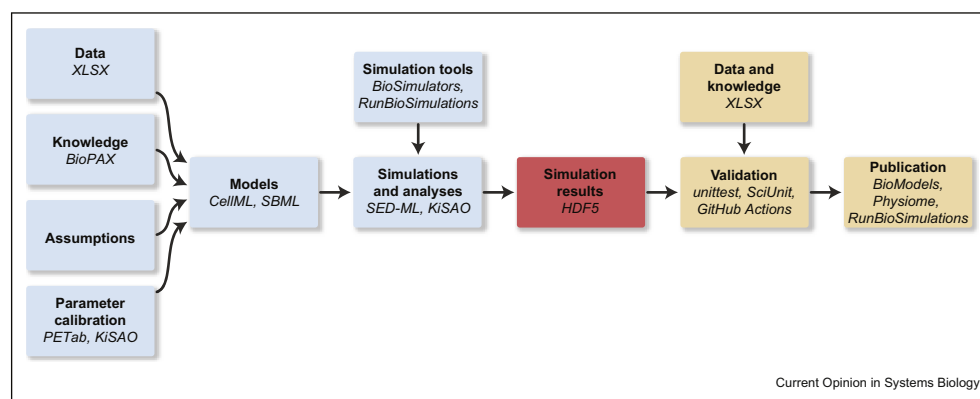
Repeating models: domain-specific model languages

Currently, models in systems biology are described using a broad range of formats including textual descriptions of equations in articles and supplementary materials; code for programming languages such as C/C++, Java, MATLAB, Python, and R; and domain-specific abstractions such as CellML [13] and the Systems Biology Markup Language (SBML) [14]. Many tools support these formats. For example, SBML is supported by tools such as BioNetGen [15], COPASI [16], PySB [17], Tellurium [18], and Virtual Cell [19]. An extensive list of tools for SBML models is available at <http://sbml.org>.

Although pioneering studies may need to use custom formats to describe innovative models, these special-purpose formats frustrate peer reviewers' abilities to review models and complicate investigators' abilities to reuse models. To facilitate quality control and collaboration, where possible, we recommend that investigators embrace formats such as CellML and SBML that provide valuable domain-specific abstractions.

We believe that the high degree of reusability that domain-specific abstractions such as CellML and SBML facilitate is nicely illustrated by the LaTeX document preparation system developed by Knuth in the late 1970s [20]. LaTeX is a powerful format for describing

Figure 1



Roadmap for conducting each step of the typical systems biology modeling and simulation workflow repeatably. For each step, we recommend formats, tools, and repositories that investigators can readily use.

textual documents, including complex documents that contain numerous figures, tables, equations, references, or meta sections such as tables of contents, glossaries, and indices. Mathematicians and physicists commonly use LaTeX to create journal articles and books. Typically, LaTeX files are compiled to PDF files, which can then be viewed by common PDF readers such as Acrobat. Although the software that Knuth developed to compile LaTeX files is no longer used, LaTeX files created over 40 years ago can still be edited and recompiled today due to the abstractions that Knuth developed. For example, a new paragraph could easily be inserted into an old LaTeX file, and modern software could easily recompile the revised LaTeX file to a new PDF. In this sense, LaTeX files have a high degree of reusability. Although the new PDF file would be easy to view, the PDF file would be less reusable than its parent LaTeX file in the sense that it would be more difficult to edit. For example, it would be difficult to insert a paragraph into the PDF file. Importantly, the abstractions provided by LaTeX have also enabled developers to replace Knuth's original software with more user-friendly versions.

We believe that the domain-specific abstractions provided by formats such as CellML and SBML are elegant solutions to similar challenges to the reuse of systems biology models. First, these abstractions often make models easier to understand by enabling modelers to describe models more similarly to their semantic biological meaning rather than describing models at a lower level in terms of their mathematical representation. We believe that this is key to facilitating model composition and collaboration. Second, these abstractions simultaneously both provide the community a common language for describing models and enable individual investigators to execute models using specific algorithms that are needed to simulate different scales or simulate models built with different kinds or amounts of data. We believe that this balances the community's needs both to collaborate and to model different systems and scales with different data and methods. For example, these abstractions enable software developers to use different algorithms to create simulation tools optimized for specific kinds of simulations (such as stiff sets of ordinary differential equations), while enabling most models to be executed with most simulation tools with similar results. Importantly, this ability to exchange most models among most tools ensures that most models will be reusable into the future, even if the original software tool used to develop the model is no longer available. For example, most SBML models developed in the 2000s with tools such as Gepasi [21] and Jarnac [22] and 2000's era simulation algorithms can still be executed with many of the modern tools cataloged at <http://sbml.org> such as COPASI [16], Tellurium

[18], and Virtual Cell [19] which implement modern algorithms.

Irrespective of which formats investigators choose to use, it is important to include metadata that enables others to evaluate and reuse models. At a minimum, models should include information about the meaning of each variable and equation and how the model was constructed, calibrated, and validated [23–25]. Where possible, this information should be provided in a structured form that is understandable by machines. The Systems Biology Graphical Notation [26,27] can be used to provide diagrammatic summaries of the meanings of models. In many cases, modelers can use tools such as SBMLsqueezer [28] and SemGen [29] and ontologies such as the Systems Biology Ontology [30] and the Ontology of Physics for Biology [31] to concretely describe the meaning of the biology represented by models. New formats and ontologies must be developed to help modelers better capture the data, assumptions, and design decisions used to construct models. PETab [32] is an emerging standard for capturing model calibrations.

Repeating simulations: simulation experiment description markup language, kinetic simulation algorithm ontology, and the COMBINE archive format

In most cases, it is important to not only share models, but to also share information about how models should be simulated and the software tools needed to execute such simulations. We recommend that modelers use the Simulation Experiment Description Markup Language (SED-ML) [33] and the Kinetic Simulation Algorithm Ontology (KiSAO) [30] to describe simulations of models. Recently, we expanded the application of SED-ML beyond kinetic simulations to flux balance, qualitative, and rule-based simulations [34]. When SED-ML cannot be used, instructions for simulating models can be provided as scripts or workflows.

In principle, information about the algorithm required for a simulation should be sufficient for its repetition. In practice, the large diversity of simulation algorithms used in systems biology, their complexity, and the lack of multiple implementations of many algorithms means that specific software tools are often needed to repeat specific studies. Therefore, we recommend that investigators use the BioSimulators registry of simulation tools to share and access software implementations of specific algorithms (<https://biosimulators.org>).

Because multiple files are often required to repeat a simulation, it is important for these files to be distributed together in a format that preserves their links. We encourage investigators to use the COMBINE (Computational Modeling in Biology Network) archive

format [35] to bundle such files. COMBINE archives are zip files that include manifest files that describe their contents. The COMBINE archive format is supported by a growing number of software tools and model repositories.

Validating models and simulations: software engineering methods and expert curation

Before data and models are shared with the community, it is important for authors to verify their findings to help ensure that other investigators focus their efforts on building upon correct results. For the most part, model verification in systems biology is still *ad hoc* and piecemeal. We believe that many investigators could enhance the quality of their data and models by adopting processes that are widely used in software engineering. Three powerful techniques that are used in software engineering to quality control complex systems are unit testing, test coverage analysis, and continuous integration (CI) [36,37].

A unit test is a small computer program that evaluates whether a part of a software system functions as intended. Typically, a unit test for an individual part of a software system checks that the part produces the expected outputs for a range of inputs. To verify that a part of a software system functions as intended across all possible inputs, unit tests often focus on checking each edge case. For example, a unit test for a function that parses a CSV file into an array of numbers could check that the method creates a non-empty array for a non-empty CSV file and an empty array for an empty CSV file. To check whether a part of a software system is adequately tested, engineers typically use coverage analysis tools to identify the lines of their code which their tests did not execute. Engineers then add additional test cases to check that these lines function as intended. Because software, like models and simulations, is often developed iteratively, with additional capabilities slowly added over time, it is often important to ensure that software maintains its initial capabilities as new capabilities are added. To help ensure that software maintains its capabilities, engineers often use CI systems to automatically execute all of their unit tests each time they edit the code for their software. Used effectively, CI systems can help developers identify problems quickly, at a stage when they are easy to fix.

Because computational models have many similarities to software, and because there are many more tools for quality controlling software than for models, we encourage modelers to apply unit testing and CI to models. First, we encourage modelers to develop a list of tests that evaluate whether their model produces accurate predictions across all of the conditions that their model captures and across all of the predictions that their model can make. For example, tests for a model of

the metabolism of a bacterium could check that the model accurately predicts how quickly the bacterium grows and how quickly it consumes nutrients across a range of tested growth media. Such unit tests can be organized using domain-independent frameworks such as Python's unittest module or domain-specific frameworks such as MEMOTE [38] and SciUnit [39].

Second, we encourage investigators to use cloud-based CI systems linked to Git repositories such as CircleCI and GitHub Actions to automatically execute their tests each time they revise their data, models, or code [40–42]. Both CircleCI and GitHub Actions are free for most open-source projects.

Although we believe that authors should bear most of the responsibility for reporting repeatable results, we also believe that independent curation services and journals should also help authors publish reusable results. Ultimately, we hope that journals only accept articles that meet a minimum threshold, such as providing a publicly accessible and repeatable version of each simulation experiment.

Because of the complexity of modern computational research and the limited resources that investigators have for peer review, we believe that dedicated curation services are needed both to help investigators organize their work for reuse by others and help journals rigorously evaluate the repeatability of submitted work. For example, our Center for Reproducible Biomedical Modeling (<https://reproduciblebiomodels.org>) has begun to provide *PLoS Computational Biology* [43] reports of the repeatability of computational results submitted to the journal. These reports outline whether the results reported by the authors can be repeated and whether the authors provide sufficient instructions for others to utilize their work. Others have recently launched similar efforts with the *American Journal of Political Science* [44], *Biostatistics* [45], and *Physiome* [46]. The *Journal of Open Source Software* [47] provides similar services for scientific software.

Publishing data, models, and simulation tools: domain-specific repositories

Over the past decade, it has become easier to publish data, models, simulations, and the software needed to reuse them. Popular avenues for sharing data, models, and simulations include supplementary materials to papers; code repositories such as GitHub; domain-independent data repositories such as figshare [48], Dryad [49], Harvard Dataverse (<https://dataverse.harvard.edu>), and Zenodo [50]; and domain-specific repositories such as BioModels [51], OpenSeek [52,53], and the Physiome Model Repository (PMR) [54,55] for models, RunBioSimulations [34] for simulations, and BioSimulators for simulation tools. We

recommend that investigators use domain-specific repositories such as BioModels, the PMR, and RunBioSimulations that provide the community helpful interfaces for discovering and exploring models and simulations and that provide permanent storage with persistent identifiers.

Popular avenues for sharing the software needed to reuse data and models include code repositories such as GitHub and software package management systems such as CRAN and PyPI. Recently, Docker images and image repositories such as BioContainers [56] and Docker Hub have become a popular way to share software [57]. Such images make it easier for developers to share the often complex computational environments needed to use scientific software. Similar to our recommendations for data, model, and simulation repositories, we encourage developers to share simulation software tools through domain-specific repositories such as BioSimulators and bio.tools [58] which provide the community helpful interfaces for finding relevant tools.

Conclusion

Adoption of domain-specific formats such as CellML, SBML, and SED-ML over the past 20 years has advanced the repeatability of computational systems biology studies. At present, many models are publicly accessible from repositories such as BioModels and PMR, the models available from these repositories often include basic metadata about model elements, and these models can often be reused with multiple simulation software tools. As a result, by some estimates approximately 50% of published results can be repeated with reasonable effort.

Despite this progress, computational systems biology still has a long way to go to make many models reproducible and reusable. Key gaps in our ability to fully reuse models include limited tools for capturing the data and assumptions used to build models; limited adoption of newer formats such as PEtav, SED-ML, and the COMBINE archive format for describing model calibrations, specifying simulations, and bundling entire studies; and limited adoption of structured approaches to verifying models such as unit testing and CI. Furthermore, our scientific culture continues to under-prioritize and under-reward reproducible research [59–64].

We believe that many of these issues can be tackled by increased adoption of community standards by existing tools and the development of new tools that fill the gaps in between them. In addition, independent curation services and public model repositories could play a vital role in teaching investigators how to conduct research more reproducibly, helping researchers prepare their work for dissemination, and helping authors and journals

evaluate the reproducibility of results submitted for publication.

Finally, we feel that it is critically important to shift the culture of science to more strongly value reproducibility and reusability. Because computational systems biology is not an isolated field, this issue must be addressed systemically across science and throughout the world.

Together, increased adoption of existing domain resources, targeted development of new tools, expanded curation services, and a cultural shift toward reproducibility could substantially enhance the reproducibility and reusability of computational systems biology. In turn, more reusable scientific building blocks could dramatically accelerate systems biology and the attainment of ambitious goals such as comprehensive computational models of cells and organisms that could underpin personalized medicine in the future.

Funding

This work was supported by the National Institute for Biomedical Imaging and Bioengineering award P41GM109824 and National Science Foundation award 1933453. The content expressed here is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, the National Science Foundation, the Icahn School of Medicine at Mount Sinai, the University of Auckland, the University of Connecticut, or the University of Washington.

Conflict of interest statement

Nothing declared.

Acknowledgements

The authors thank the COMBINE (COmputational Modeling in BIOlogy NEtwork) community for many fruitful discussions that helped spur the ideas expressed here.

References

Papers of particular interest, published within the period of review, have been highlighted as:

- * of special interest
- ** of outstanding interest

1. Roger B: **Opus Majus, 1267**. In *Part 1. Opus Majus, volumes 1 and 2*. Edited by Bacon R, University of Pennsylvania Press; 2016. SN - 9781512814064.
2. Tiwari K, Kananathan S, Roberts MG, Meyer JP, Sharif Shohan MU, Xavier A, Maire M, Zyoud A, Men J, Ng S, *et al*: **Reproducibility in systems biology modelling**. *Mol Syst Biol* 2021, **17**, e9982.
- ** One of the first papers that systematically analyzed the irreproducibility of computational modeling in systems biology.
3. Hothorn T, Leisch F: **Case studies in reproducibility**. *Briefings Bioinf* 2011, **12**:288–300.
4. National Academies of Sciences, Engineering, and Medicine and others: **Reproducibility and replicability in science**. National Academies Press; 2019.

5. Haibe-Kains B, Adam GA, Hosny A, Khodakarami F, Waldron L, Wang B, McIntosh C, Goldenberg A, Kundaje A, Greene CS, *et al.*: **Transparency and reproducibility in artificial intelligence**. *Nature* 2020, **586**:E14–E16.
6. McKinney SM, Karthikesalingam A, Tse D, Kelly CJ, Liu Y, Corrado GS, Shetty S: **Reply to: transparency and reproducibility in artificial intelligence**. *Nature* 2020, **586**:E17–E18.
7. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, da Silva Santos LB, Bourne PE, *et al.*: **The FAIR Guiding Principles for scientific data management and stewardship**. *Scientific Data* 2016, **3**.
8. Porubsky VL, Goldberg AP, Rampadarath AK, Nickerson DP, Karr JR, Sauro HM: **Best practices for making reproducible biochemical models**. *Cell Syst* 2020, **11**:109–120.
9. McKinney W: **pandas: a foundational Python library for data analysis and statistics**. In *Python for high performance and scientific computing*, seattle; 2011:1–9.
10. Folk M, Heber G, Koziol Q, Pourmal E, Robinson D: **An overview of the HDF5 technology suite and its applications**. In *Proceedings of the EDBT/ICDT 2011 workshop on Array databases*; 2011:36–47.
11. Hoyer S, Hamman J: **xarray: ND labeled arrays and datasets in Python**. *J Open Res Software* 2017, **5**.
12. Demir E, Cary MP, Paley S, Fukuda K, Lemer C, Vastrik I, Wu G, D'eustachio P, Schaefer C, Luciano J, *et al.*: **The BioPAX community standard for pathway data sharing**. *Nat Biotechnol* 2010, **28**:935–942.
13. Cuellar AA, Lloyd CM, Nielsen PF, Bullivant DP, Nickerson DP, Hunter PJ: **An overview of CellML 1.1, a biological model description language**. *Simulation* 2003, **79**:740–747.
- CellML is a one of the most commonly used format for describing models in systems biology. CellML is primarily used for models of animal physiology.
14. Keating SM, Waltemath D, König M, Zhang F, Dräger A, Chaouiya C, Bergmann FT, Finney A, Gillespie CS, Helikar T, *et al.*: **SBML Level 3: an extensible format for the exchange and reuse of biological models**. *Mol Syst Biol* 2020, **16**, e9110.
- SBML is the most widely used format for describing models in systems biology. SBML's abstractions facilitate the interpretation and reuse of models.
15. Harris LA, Hogg JS, Tapia J-J, Sekar JA, Gupta S, Korsunsky I, Arora A, Barua D, Sheehan RP, Faeder JR: **BioNetGen 2.2: advances in rule-based modeling**. *Bioinformatics* 2016, **32**:3366–3368.
16. Bergmann FT, Hoops S, Klahn B, Kummer U, Mendes P, Pahle J, Sahle S: **COPASI and its applications in biotechnology**. *J Biotechnol* 2017, **261**:215–220.
17. Lopez CF, Muhlich JL, Bachman JA, Sorger PK: **Programming biological models in Python using PySB**. *Mol Syst Biol* 2013, **9**:646.
18. Choi K, Medley JK, König M, Stocking K, Smith L, Gu S, Sauro HM: **Tellurium: an extensible Python-based modeling environment for systems and synthetic biology**. *Biosystems* 2018, **171**:74–79.
19. Loew LM, Schaff JC: **The Virtual Cell: a software environment for computational cell biology**. *Trends Biotechnol* 2001, **19**:401–406.
20. Knuth D: *TeXbook*. Addison-Wesley Reading; 1984.
21. Mendes P: **GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems**. *Comput Appl Biosci* 1993, **9**:563–571.
22. Sauro HM: **JARNAC: a system for interactive metabolic analysis**. In *9th int BioThermoKinetics Meeting*. Edited by Hofmeyr SJ, Rohwer JM, Stellenbosch University Press; 2000: 221–228.
23. Le Novère N, Finney A, Hucka M, Bhalla US, Campagne F, Collado-Vides J, Crampin EJ, Halstead M, Klipp E, Mendes P, *et al.*: **Minimum information requested in the annotation of biochemical models (MIRIAM)**. *Nat Biotechnol* 2005, **23**:1509–1515.
24. Mulugeta L, Drach A, Erdemir A, Hunt CA, Horner M, Ku JP, Myers Jr JG, Vadigepalli R, Lytton WW: **Credibility, replicability, and reproducibility in simulation for biomedicine and clinical applications in neuroscience**. *Front Neuroinf* 2018, **12**:18.
25. Erdemir A, Mulugeta L, Ku JP, Drach A, Horner M, Morrison TM, Peng GC, Vadigepalli R, Lytton WW, Myers JG: **Credible practice of modeling and simulation in healthcare: ten rules from a multidisciplinary perspective**. *J Transl Med* 2020, **18**:1–18.
- Guidelines for good practices in multi-scale modeling from the Inter-agency Modeling and Analysis Group.
26. Novère NL, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, Demir E, Wegner K, Aladjem MI, Wimalaratne SM, Bergman FT, Gauges R, Ghazal P, Kawaji H, Li L, Matsuoka Y, Villéger A, Boyd SE, Calzone L, Courtot M, Dogrusoz U, Freeman TC, Funahashi A, Ghosh S, Jouraku A, Kim S, Kolpakov F, Luna A, Sahle S, Schmidt E, Watterson S, Wu G, Goryanin I: **The Systems Biology graphical notation**. *Nat Biotechnol* 2009, **27**:735–741.
27. Rounay A: **sbgntikz—a TikZ library to draw SBGN maps**. *Bioinformatics* 2019, **35**:4499–4500.
28. Dräger A, Zielinski DC, Keller R, Rall M, Eichner J, Palsson BO, Zell A: **SBMLsqueezer 2: context-sensitive creation of kinetic equations in biochemical networks**. *BMC Syst Biol* 2015, **9**:1–17.
29. Neal ML, Thompson CT, Kim KG, James RC, Cook DL, Carlson BE, Gennari JH: **SemGen: a tool for semantics-based annotation and composition of biosimulation models**. *Bioinformatics* 2019, **35**:1600–1602.
30. 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**. *Mol Syst Biol* 2011, **7**:543.
31. Cook DL, Neal ML, Bookstein FL, Gennari JH: **Ontology of Physics for Biology: representing physical dependencies as a basis for biological processes**. *J Biomed Semant* 2013, **4**:1–8.
32. Schmiester L, Schälte Y, Bergmann FT, Camba T, Dudkin E, Ebert J, Fröhlich F, Fuhrmann L, Hauber AL, Kemmer S, *et al.*: **PETab—interoperable specification of parameter estimation problems in systems biology**. *PLoS Comput Biol* 2021, **17**, e1008646.
33. Waltemath D, Adams R, Bergmann FT, Hucka M, Kolpakov F, Miller AK, Moraru IL, Nickerson D, Sahle S, Snoep JL, Le Novère N: **Reproducible computational biology experiments with SED-ML - the Simulation Experiment Description Markup Language**. *BMC Syst Biol* 2011, **5**:198.
- SED-ML is the main standard for describing simulations in computational systems biology. SED-ML can be used with any model language such as CellML and SBML.
34. Shaikh B, *et al.*: **RunBioSimulations: an extensible web application that simulates a wide range of computational modeling frameworks, algorithms, and formats**. *Nucleic Acids Res* 2021, **49**.
35. Bergmann FT, Adams R, Moodie S, Cooper J, Glont M, Golebiewski M, Hucka M, Laibe C, Miller AK, Nickerson DP, Olivier BG, Rodriguez N, Sauro HM, Scharm M, Soiland-Reyes S, Waltemath D, Yvon F, Le Novère N: **COMBINE archive and OME format: one file to share all information to reproduce a modeling project**. *BMC Bioinf* 2014, **15**:369.
36. Hellerstein JL, Gu S, Choi K, Sauro HM: **Recent advances in biomedical simulations: a manifesto for model engineering**. *F1000Res* 2019, **8**.
37. Schölzel C, Blesius V, Ernst G, Dominik A: **The impact of mathematical modeling languages on model quality in systems biology: a software engineering perspective**. *bioRxiv* 2020:1–28, <https://doi.org/10.1038/s41540-021-00182-w>.

38. Lieven C, Beber ME, Olivier BG, Bergmann FT, Ataman M, Babaei P, Bartell JA, Blank LM, Chauhan S, Correia K, *et al.*: **MEMOTE for standardized genome-scale metabolic model testing.** *Nat Biotechnol* 2020, **38**:272–276.
 39. Omar C, Aldrich J, Gerkin RC: **Collaborative infrastructure for test-driven scientific model validation.** *Proc 36th Int Conf Softw Eng* 2014:524–527.
 40. Meyer M: **Continuous integration and its tools.** *IEEE Softw* 2014, **31**:14–16.
 41. Beaulieu-Jones BK, Greene CS: **Reproducibility of computational workflows is automated using continuous analysis.** *Nat Biotechnol* 2017, **35**:342–346.
 42. Krafczyk M, Shi A, Bhaskar A, Marinov D, Stodden V: **Scientific tests and continuous integration strategies to enhance reproducibility in the scientific software context.** In *Proc 2nd int workshop pract reproducible eval comput syst*, 2019:23–28.
 43. Papin JA, Mac Gabhann F, Sauro HM, Nickerson D, Rampadarath A: **Improving reproducibility in computational biology research.** *PLoS Comput Biol* 2020, **16**:1–3, <https://doi.org/10.1371/journal.pcbi.1007881>. 1007881.
- This paper describes one of the first pilot studies to use outside curation to improve modeling papers in systems biology.
44. **Editors of American Political Science Review.** Notes from the editors *Am Pol Sci Rev* 2019, **113**:iii–vii.
 45. Peng RD: **Reproducible research and biostatistics.** *Biostatistics* 2009, **10**:405–408.
 46. Hunter P: **The Virtual Physiological Human: the Physiome project aims to develop reproducible, multiscale models for clinical practice.** *IEEE Pulse* 2016, **7**:36–42.
 47. Smith AM, Niemeyer KE, Katz DS, Barba LA, Githinji G, Gymrek M, Huff KD, Madan CR, Mayes AC, Moerman KM, *et al.*: **Journal of Open Source Software (JOSS): design and first-year review.** *PeerJ Comput Sci* 2018, **4**:e147.
 48. Singh J: **FigShare.** *J Pharmacol Pharmacother* 2011, **2**:138–139.
 49. White H, Carrier S, Thompson A, Greenberg J, Scherle R: **The Dryad Data Repository: a Singapore framework metadata architecture in a DSpace environment.** In *Dublin core conference*; 2008:157–162.
 50. Peters I, Kraker P, Lex E, Gumpenberger C, Gorraiz JI: **Zenodo in the spotlight of traditional and new metrics.** *Front Res Metr Anal* 2017, **2**:13.
 51. Malik-Sheriff RS, Glont M, Nguyen TV, Tiwari K, Roberts MG, Xavier A, Vu MT, Men J, Maire M, Kananathan S, *et al.*: **BioModels–15 years of sharing computational models in life science.** *Nucleic Acids Res* 2020, **48**:D407–D415.
- BioModels is one of the largest repositories of systems biology models. BioModels is primarily used to publish models that are encoded in SBML. Importantly, BioModels' team work to ensure that each curated model can reproduce at least one reported result.
52. Wittig U, Rey M, Weidemann A, Müller W: **Data management and data enrichment for systems biology projects.** *J Biotechnol* 2017, **261**:229–237.
 53. Wolstencroft K, Owen S, Krebs O, Nguyen Q, Stanford NJ, Golebiewski M, Weidemann A, Bittkowski M, An L, Shockley D, *et al.*: **SEEK: a systems biology data and model management platform.** *BMC Syst Biol* 2015, **9**:1–12.
 54. Lloyd CM, Lawson JR, Hunter PJ, Nielsen PF: **The CellML model repository.** *Bioinformatics* 2008, **24**:2122–2123.
 55. Yu T, Lloyd CM, Nickerson DP, Cooling MT, Miller AK, Garny A, Terkildsen JR, Lawson J, Britten RD, Hunter PJ, *et al.*: **The Physiome Model Repository 2.** *Bioinformatics* 2011, **27**:743–744.
- PMR is one of the largest repositories of systems biology models. PMR is primarily used to publish models described using CellML.
56. da Veiga Leprevost F, Grüning BA, Alves Aflitos S, Röst HL, Uszkoreit J, Barsnes H, Vaudel M, Moreno P, Gatto L, Weber J, *et al.*: **BioContainers: an open-source and community-driven framework for software standardization.** *Bioinformatics* 2017, **33**:2580–2582.
 57. Nüst D, Sochat V, Marwick B, Eglen SJ, Head T, Hirst T, Evans BD: **Ten simple rules for writing Dockerfiles for reproducible data science.** *PLoS Comput Biol* 2020, **16**:1–24, <https://doi.org/10.1371/journal.pcbi.1008316>. e1008316.
 58. Ison J, Ienasescu H, Chmura P, Rydzka E, Ménager H, Kalas M, Schwämmle V, Grüning B, Beard N, Lopez R, *et al.*: **The bio.-tools registry of software tools and data resources for the life sciences.** *Genome Biol* 2019, **20**:1–4.
 59. Schnell S: **"Reproducible" research in mathematical sciences requires changes in our peer review culture and modernization of our current publication approach.** *Bull Math Biol* 2018, **80**:3095–3105.
 60. Samota EK, Davey RP: **Knowledge and attitudes among life scientists towards reproducibility within journal articles.** *BioRxiv* 2019:581033.
 61. Hrynaskiewicz I: **Publishers' responsibilities in promoting data quality and reproducibility.** In *Good research practice in non-clinical pharmacology and biomedicine*. Cham: Springer; 2019:319–348.
 62. Heesen R: **Why the reward structure of science makes reproducibility problems inevitable.** *J Philos* 2018, **115**:661–674.
 63. Quan W, Chen B, Shu F: **Publish or impoverish: an investigation of the monetary reward system of science in China (1999–2016).** *Aslib J Inform Manag* 2017.
 64. Cowley Jr AW: **A journalistic break through.** *Function* 2020, **1**.