Automated Credibility Assessment of Computational Models in Systems Biology

Lillian Tatka

Submitted in partial fulfillment
of the requirements for the
Bioengineering Ph.D. General Exam

Supervisory Committee:
Herbert Sauro, Chair
Patrick Boyle
Joseph Hellerstein
Lucian Smith
Wendy Thomas
Georg Seelig, GSR

Department of Bioengineering
University of Washington
May 2023

Abstract

Computational models are increasingly used in high-impact decision making in science, engineering, and medicine. It is crucial that computational models meet a standard of credibility when using them in high-stakes decision making. For this reason, institutes including NASA, the FDA, and the EMA have developed standards to promote and assess the credibility of computational models and simulations. However, due to the breadth of models these institutes assess, these credibility standards are mostly qualitative and avoid making specific recommendations. On the other hand, modeling and simulation in systems biology is a narrower domain and several standards are already in place. These factors facilitate the development of a quantitative credibility standard. A systems biology credibility standard is essential given the rise in complexity and influence of models. This proposal describes the development of a testing suite to standardize and automate credibility testing and scoring of computational systems biology models. The test suite will include dynamic tests that examine the output of a simulated model, verification and validation tests, and a preliminary assessment of parameter sensitivity to gauge uncertainty in the model's predictions.

Contents

Chapter 1

Introduction

This document describes my completed work to date and outlines research goals for the remainder of my PhD. Research can be split into two major projects: research on biochemical oscillators (completed) and research on computational model credibility (proposed). The projects described focus on computational models of (bio)chemical reaction networks (CRNs). Although several modeling paradigms exist to describe CRNs, this work uses CRNs represented by systems of ordinary differential equations. A CRN is a set of chemical reactions R_i where i is the index over the range 1 to n, the total number of reactions in the network . Reactions involve the chemical species of the network, S_j where j is the index of each species from 1 to the total number of species. A reaction is then represented as follows:

$$Ri : \sum_{j \in S} \alpha_{ij} S_j \rightarrow \sum_{j \in S} \beta_{ij} S_j$$

where α_{ij} and β_{ij} are non-negative integers, stoichiometry coefficients. For example, a simple 3-species, 2-reaction network can be represented by the following reactions:

$$\begin{gathered} S1 + S2 \rightarrow S3 \\ S3 \rightarrow S1 + S2 \end{gathered} \tag{1.1}$$

However, these reactions alone are insufficient to completely describe the small network. The rate constants and ratelaws for the reaction are unspecified. This research focuses on mass-action kinetics, where the rate of a reaction is dictated by the concentration of its substrates and a rate constant:

$$\begin{aligned} S1 + S2 \rightarrow S3; \qquad & k_1 * S1 * S2 \\ S3 \rightarrow S1 + S2; \qquad & k_2 * S3 \end{aligned} \tag{1.2}$$

The rate of change of each species can then be put in the form of a differential equation (equation 1.3) and solved numerically.

$$\begin{aligned} \frac{dS1}{dt} &= k_2S3 - k_1S1S2 \\ \frac{dS2}{dt} &= k_2S3 - k_1S1S2 \\ \frac{dS3}{dt} &= k1S1S2 - k_2S3 \end{aligned} \tag{1.3}$$

The first chapter of this document covers the exploration of oscillating chemical reaction networks. This work was intended as a small initial step towards understanding the function of biochemical oscillators and identifying common design patterns that enable oscillation. This work involved computationally evolving a large population of oscillators and comparing their reaction characteristics to non-oscillating systems. The library of reaction networks used for this work was made publicly available to enable further oscillator/design pattern research as well as to serve as a standardized data set for testing novel software and algorithms. This research is described in more detail in chapter 1, which is adapted from the following publication:

> **Tatka, L.**, Luk, W., Hellerstein, J., Elston, T., Sauro, H. "Cesium: A Public Database of Evolved Oscillatory Reaction Networks." Biosystems, vol. 224, 2023, p.104836., https://doi.org/10.1016/j.biosystems.2023.104836.

The use of the oscillator database as a means of benchmarking and testing future software led to the exploration of credibility in computational modeling. Although the Center for Reproducible Biomedical Modeling (CRBM) at the University of Washington is well-known for creating extensive standards for model reproducibility, the concept of model credibility in systems biology has yet to be thoroughly explored. Credibility in computational modeling has only risen in importance as computational models become more common and are increasingly used to make impactful decisions such as approving a drug or moving forward with an expensive experiment. The need for more comprehensive credibility standards was recently highlighted during the COVID pandemic where models of questionable quality and credibility were relied upon to make public safety decisions [1].

As the use and reliance on computational modeling grows, institutions such as the National Aeronautics and Space Administration (NASA) and the Food and Drug Administration (FDA) (among others) have began to develop standards to encourage model credibility. These standards are largely qualitative owing to the variety of models and simulations in these fields. Systems biology has a unique advantage in that the scope of models is relatively well-defined extensive standards are already in place for describing, annotating, and simulating model. This work serves as the background for the proposed credibility research. Credibility standards in other fields and current standards in systems biology are reviewed in chapter 2, which is adapted from the following publication:

> **Tatka, L.**, Smith, L., Hellerstein, J., Sauro, H. "Adapting Modeling and Simulation Credibility Standards to Computational Systems Biology." Journal of Translational Medicine, submitted.

The proposed research aims to partially address the lack of credibility standards and testing in computational systems biology by developing credibility standards and tests. Unlike the budding credibility standards developed by NASA and the FDA, the proposed research seeks to create concrete and quantitative measures of credibility to partially eliminate assessment subjectivity. **The proposed research aims to create a software package to automatically assess the credibility of systems biology models.** The proposed software would automate dynamic tests, verification & validation, and sensitivity analysis of systems biology models and produces a quantitative score or model credibility. The target users for this software are (1) modelers who may use this tool during the development process to iteratively improve their model (2) by journals or reviewers who may wish to verify a model is sufficiently credible for publication and (3) third parties who want to assess a previously published model's credibility prior to using it in their own studies or decisions.

The aims of the proposed research are defined as follows and explained in detail in chapters 4 - 7.

- **Aim 1:** Develop a python package to perform dynamic tests on systems biology models in order to assess their credibility

- **Aim 2:** Create tools to automatically verify and validate models

- **Aim 3:** Build software to automate simple parameter sensitivity analyses

This research will result in two first author publications, the titles of which are to be determined. The first publication will describe the design and purpose of the software package and include some preliminary examples of implementation. The second publication will present the full software package and demonstrate its use on several existing models. I hope to finish this work by June 2024.

Chapter 2

Previous Work:

A Public Database of Evolved Oscillatory Reaction Networks

2.1 Background

With the constant emergence of new software tools and techniques in systems biology, access to a large variety of chemical reaction network models with defined characteristics can be useful for testing and validation. For example, an existing data set enumerating small chemical reaction networks [2] has been used to test computational methods to analyze bistability [3], software to reverse engineer networks [4], and a novel framework to assess networks for multiple equilibria and other characteristics [5]. Generating these test data sets can be computationally expensive and time-consuming. The data set generated by Deckard et al. took several days to compute and contains approximately 47 million small models, but its purpose is to exhaustively enumerate small networks, not to assess and catalog their behavior [2]. Despite the usefulness of standard validation data sets, currently no public database exists containing numerous models displaying a variety of network behaviors.

In terms of network behavior, oscillatory chemical reaction networks are of particular interest due to their biological relevance, including embryogensis, DNA repair, and heart function [6, 7, 8, 9].

Lotka et al. documented the first ordinary differential equation model of an oscillatory predator-prey network [10]. The study of oscillatory mechanisms in chemical reaction networks dramatically increased with the expansion of computing power and the improved ability to solve non-linear differential equations [11]. Novak and Tyson studied non-autocatalyic small oscillators to determine design patterns and basic requirements for oscillating systems [6]. These works focus on the mechanisms and characteristics of individual oscillators or to oscillators as a broad category. Paladugu et al. used in silico evolution to create several oscillators, bistable switches, homeostatic systems and frequency filters, but the library was not made publicly available for further study [12].

We present a public database, entitled ”Cesium”, containing approximately 1800 3-species, 450 6-species, and 750 10-species oscillating networks created using a customized evolution algorithm, an optimization process based on the iterative improvement of candidate models. There are also random non-oscillating networks to serve as controls. The database can be searched by the number of reactions, the presence of autocatalytic reactions, and the number of species. Each entry contains the above information as well as the Antimony string [13] of the model which can be simulated by any software capable of reading antimony or SBML (to which Antimony models can be converted). Future work will expand this database to include a wider variety of oscillatory models as well as models that display other behaviors such as bistability.

The population of evolved oscillatory reaction networks possessed different reaction type compositions compared to non-oscillating networks that were randomly generated. Oscillating networks had significantly more uni-bi reactions and autocatalytic reactions.

The purpose of this research is two-fold: basic research into characteristics of oscillatory systems, and the development of this database.

The availability of a standardized database for use in software and algorithm testing pushed me in the direction of model credibility. This is a step beyond reproducibility, the goal of which is the easy reproduction of a model and its results by third party users. Credibility refers to the trustworthiness of a model. This topic and related research will be covered in later chapters.

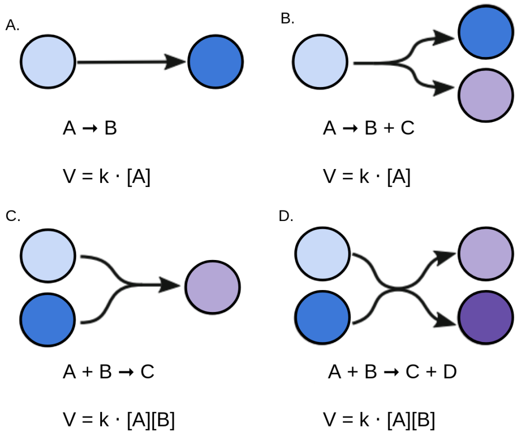


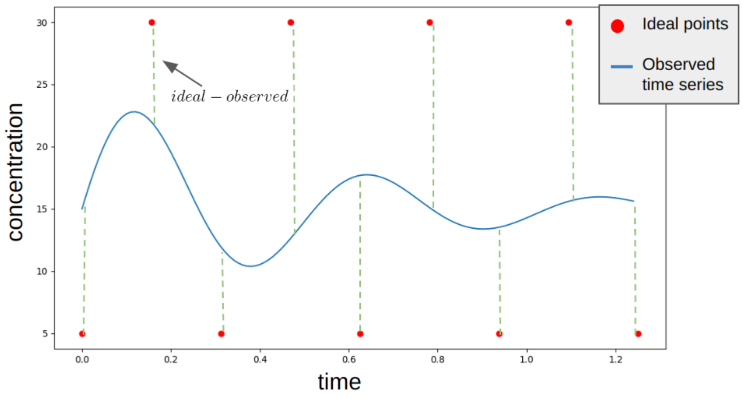
Figure 2.1: Models were composed of four reaction types and rate laws: (A) uni-uni, (B) uni-bi, (C) bi-uni, (D) bi-bi

2.2 Methods

Oscillating models were generated using evolution scripts written in python, available at https://github.com/sys-bio/evolution. The process begins by randomly generating a population of models and gradually modifying them over time to produce oscillatory behavior. Four types of reactions are used: uni-uni, uni-bi, bi-uni, and bi-bi, all with mass-action kinetics (figure 2.1). At each step, every model is evaluated against an objective function scoring the model's ability to oscillate. Models with better fits are selected and further modified and models with lower fits are gradually eliminated.

2.2.1 Model Representation

Models and reactions were represented by custom data structures. An instance of the *reaction* object represents a single reaction and consists of one or two reactants, one or two products, a rate constant, and an integer representing the reaction type (uni-uni, uni-bi, bi-uni, or bi-bi). The *model* object represents a single reaction network and consists of a list of species, a list of initial concentrations, and a list of *reaction* objects. The software converts *model* objects into systems of ordinary differential equations which are then numerically solved to produce time series data of species concentrations. After the evolution process is complete, the models are converted to antimony [13], a human readable model definition language.



$$fitness = min(\sum(ideal - observed)^2)$$

Figure 2.2: For the time series data of this species, the error is calculated by summing the square difference between the observed time series data (blue) and a set of idealized oscillator points (red). This process is repeated for each species in the model and the fitness is defined as the minimum of these sums.

2.2.2 Objective Function

The objective function minimized error between the candidate model's time course data and a series of points corresponding to the peaks and troughs of an oscillator. This "idealized" oscillator time series consisted of nine concentration points, alternative between 5 and 30 concentration units over the course of 1.25 seconds (figure 2.2). The output time series of each species was compared to these points by summing the squared difference between the observed point and the "ideal" point and the smallest error value was considered the model's fitness. This is similar to MSE but the sum is not divided by the number of observations as this number is constant across all models. This has been demonstrated to be an effective objective function for the in silico evolution of chemical oscillators [12]. In cases where candidate models could not be simulated, an arbitrarily high fitness value was assigned resulting in their subsequent removal from the population in the selection step.

2.2.3 ODE Solver

Preliminary studies suggested that the most commonly used python ODE solver from scikit was inaccurate for solving these problems. Other off-the-shelf solvers lacked the ability to deal with the custom data structure encoding the models. The Sauro Lab's simulation software, RoadRunner [14], was also unsuitable for this purpose due to the computational cost of frequently modifying and recompiling models for simulation. Many algorithms, such as RK4 are not robust enough for this work. For this reason, a custom simulator using the CVODE solver from the Sundials suite was used [15].

2.2.4 Selection

In each new generation, the top 10% of models from the previous generation were copied without modification. The remainder of the new population was chosen by tournament selection [16] from the previous generation. This is where two models are randomly chosen from the previous generation (including from the top 10% of models already carried over) and the model with the better fitness is mutated by adding or deleting a reaction, or by modifying a rate constant. The modified model is then appended to the new generation.

2.2.5 Mutation

After tournament selection, the fitter model was either mutated by modifying a rate constant or adding/deleting a reaction with equal probability. In the case of rate constant modification, a random rate constant was adjusted by a random percentage between -15% and +15% of the rate constant's current value. This mechanism ensured that rate constants could not become negative, and there was no upper limit to their value.

In the case of reaction modification, a reaction was either added or deleted with 50%-50% probability. If deleted was selected, a random reaction was removed from the model. If addition, a new reaction was added with the equal probability of each reaction type: uni-uni, uni-bi, bi-uni, bi-bi (figure 2.1).

2.2.6 Random Networks for Evolution

A population of 40 random networks were generated using the `teUtils` python package [17]. Each network was initialized with three species, and nine reactions with the probability of each reaction type being 0.1, 0.4, 0.4, 0.1 for uni-uni, uni-bi, bi-uni, and bi-bi reactions. These settings were chosen

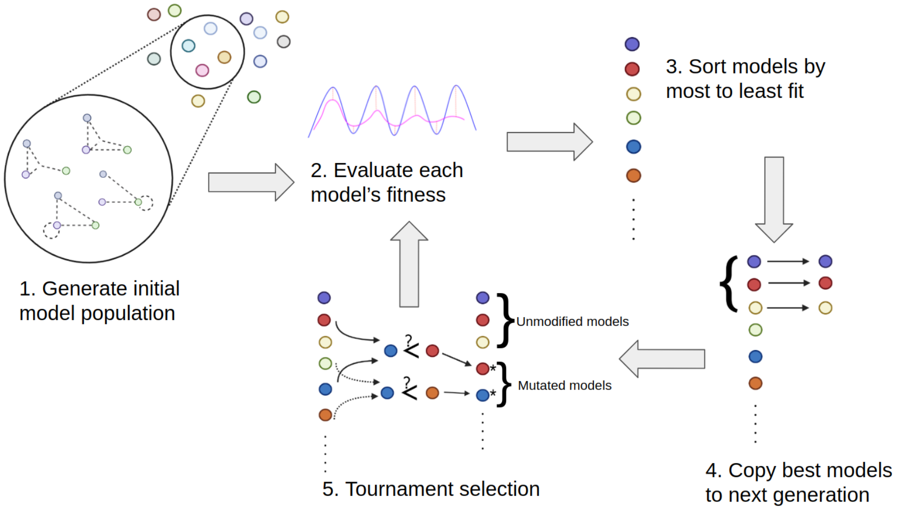

1. Generate initial model population
2. Evaluate each model's fitness
3. Sort models by most to least fit
4. Copy best models to next generation
5. Tournament selection
Unmodified models
Mutated models

Figure 2.3: The evolution algorithm. (1) An initial population of random models is created. (2) The fitness of each model is evaluated. (3) The entire population is sorted by most fit to least fit. (4) The top 10% of the models are transferred to the subsequent generation unmodified. (5) Tournament selection is used to populate the remainder of the subsequent generation. Models are randomly selected, the more fit model is chosen to be modified and carried over to the subsequent generation. Steps 2-5 are repeated.

to maximize the number of evolution trials that successfully product oscillators. All reactions had mass-action kinetics with a random rate constant between 0 and 50.

2.2.7 Custom Evolution Algorithm

The evolutionary algorithm mimics biological evolution in that populations of individuals, in this case candidate networks, are altered and forced to compete with each other. Over the course of generations, fitter individuals (models that oscillate or are likely to oscillate) out compete unfit individuals (models that do not oscillate or can not be simulated) and survive into the next generation. Although genetic and evolutionary algorithms are well characterized, their application to systems biology models remains a challenge as a key trait in these algorithms is crossover, the ability of two possible solutions to exchange information, creating a new, ideally more suitable, candidate solution [18]. Typically, genetic algorithms operate on objective functions for which solutions consist of vectors where values can easily be exchanged between two vectors. It is uncertain how crossover of mass-action kinetic models could be achieved as solutions (candidate networks) consist both of topology (how reactions are connected) as well as rate constants (vectors of values). Additionally, the number of reactions and rate constants vary and must be equal but can vary from model to model. For this reason, a custom optimization process was developed based on genetic algorithms but avoiding crossover.

This process begins with the generation of 40 random networks with pre-specified probabilities for each of the four reaction types. The fitness of each model is evaluated by comparing the model's time series data with an objective function representing the desired outcome, oscillation. Models that with time series data close to this desired outcome, those that oscillate or those with damped oscillations, are more fit than those that fail to oscillate or cannot be simulated.

These models are ordered from most to least fit and the top 10% of the models are carried over unmodified to the subsequent generation (figure 2.3). The remaining models are randomly paired and compared and the fitter of the two models is slightly modified by either adding or deleting a reaction or changing a rate constant. If the modification improves the fitness of the model, the modified model is added to the next generation. If the modification makes the model less fit, 75% of the time the unmodified model is added to the next generation and 25% of the time the less fit model will be added. This allows for the chance that small changes initially make a model worse, but subsequent changes drastically improve the model. Once the new generation is fully populated, the models are again ordered from most to least fit and the process begins again. This is repeated for 400 generations or until a threshold fitness level is reached.

2.2.8 Random Control Networks

Random models were generated as described in the previous section. To control for changes made during evolution, the random models underwent the same mutation processes described previously. Instead of populating subsequent generations based on model fitness, 10% of the previous population was randomly selected to be carried over unmodified to the new generation. The remainder of the new generation was populated with models randomly chosen and mutated from the previous generation. The purpose of this procedure was to account for model variability introduced in the evolutionary process. Of 1000 random control networks generated, 1 was a spontaneous oscillator.

2.3 Results

Evolved models are processed to remove any undesirable oscillators from the population, namely oscillators that dampen over time or oscillators where one or more species concentrations rise indefinitely towards infinity. Next, duplicated reactions were fused and reactions that are superfluous to oscillation are deleted to minimize network size resulting in a population of reaction networks that oscillate indefinitely in species concentration, examples of which are shown in figure 4.

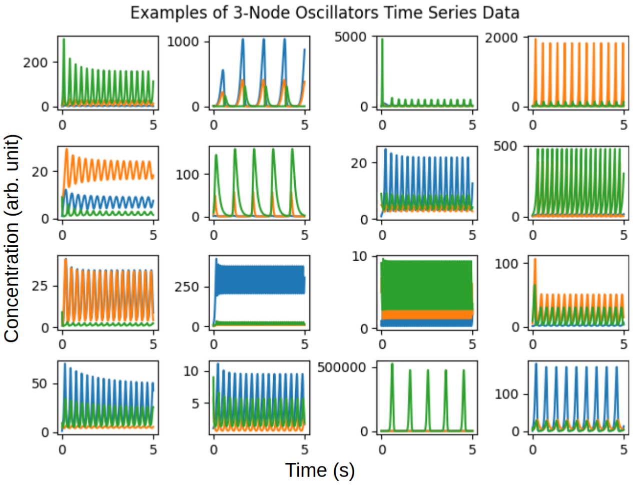
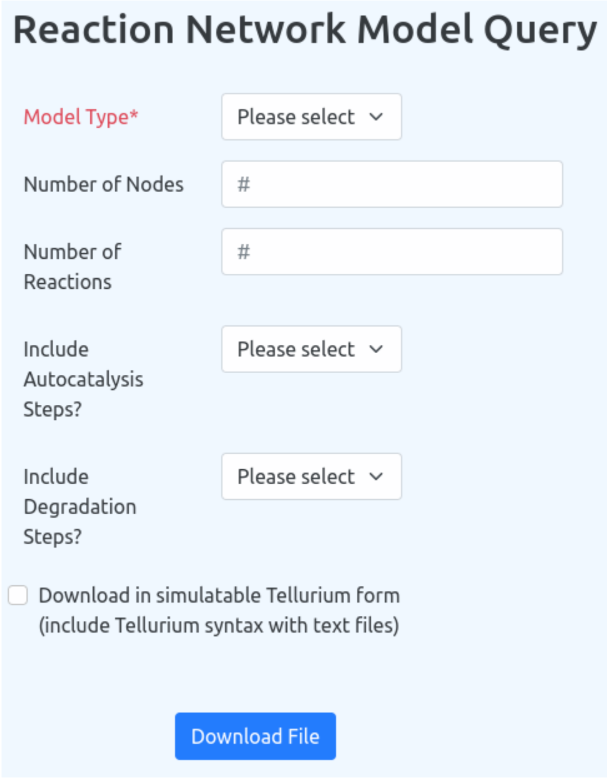


Figure 2.4: Time series data of sixteen 3-node oscillators generated using differential evolution.

2.3.1 Database Construction

A non-relational key-value database was deployed via Mongo Atlas using the PyMongo API to store all models and their information. The user enters the desired reaction network attributes into the web GUI (figure 2.5). Models are downloaded as a .zip file of text files, or if a single model a single text file, containing the antimony string of each model. If the ”Download in simulatable Tellurium form” box is checked, then each model file will contain the necessary python package imports and formatting to be immediately simulated and plotted upon running the file.

The current website queries the database by model type (current options are oscillator or random), the number of nodes (species), the number of reactions, and the presence or absence of autocatalysis or degradation reactions. Both oscillator and non-oscillating control networks are stored in the database. Each entry also includes a dictionary of tallies of each reaction type in the network, a list of redundant



Reaction Network Model Query

Model Type* Please select

Number of Nodes #

Number of Reactions #

Include Autocatalysis Steps? Please select

Include Degradation Steps? Please select

□ Download in simulatable Tellurium form (include Tellurium syntax with text files)

Figure 2.5: Landing page for the Cesium database.

reactions that were fused during post-processing, a list of reactions that were deleted as their presence did not influence oscillation.

In addition to the Antimony string comprising the model, other characteristics such as its initial reaction probabilities, fused and deleted reactions, and reaction counts are also tracked in the database. The database can be queried to select models with any number of desired traits. Both oscillator and non-oscillating control networks are stored in the database.

The Cesium database is publicly available on the web at `https://cesiumdb.herokuapp.com/` and is intended to serve as source of reaction networks with specified characteristics for use in research and validation. It currently contains 2000 randomly generated models and approximately 1800 3-species oscillating networks. This database can be expanded in the future to include a wider variety of oscillatory networks as well as networks with different behaviors, such as bistability.

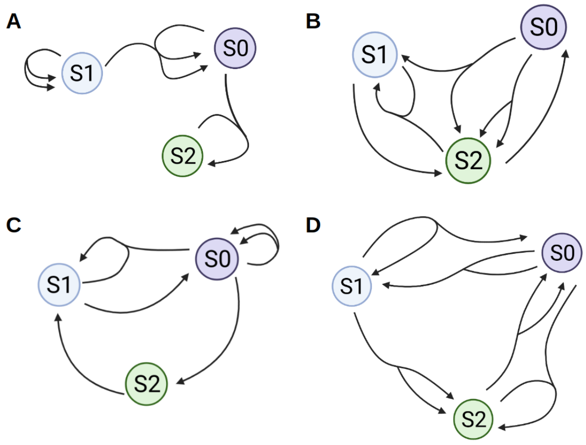


Figure 2.6: Examples of 3-species oscillating networks from the Cesium database.

2.3.2 Oscillating Network Examples

Four oscillating networks from the database are shown in figure 2.6. Arrows symbolize reaction and lead from the reactant to the product. A double headed arrow indicates two products are formed and a double tail indicates two reactants.

In network A, S1 is autocatalytic as is S0 (and catalyzed by S1). These linked positive feedback loops cause oscillation with products out flowing to S2, which essentially behaves as a boundary species, a species that is unaffected by the model and whose concentration remains fixed. Network B lacks autocatalytic reactions completely. Computing the jacobian matrix at the unstable focus results in the following matrix:

$$\begin{array}{cc} \begin{array}{ccc} S0 & S1 & S2 \end{array} & \\ \begin{pmatrix} -4.75 & 0 & 15.2 \\ 0.75 & -0.3 & 0 \\ 8.75 & -1.6 & -22.8 \end{pmatrix} & \begin{array}{c} S0 \\ S1 \\ S2 \end{array} \end{array}$$

In the bottom row center, the negative number indicates that there is an inhibitory relationship between S1 and S2. An increase in S2 will decrease the production rate of S1. These interactions for a cycle with negative feedback, suggesting that this is a feedback oscillator (figure 2.7). Similar analyses reveal that networks C and D are also likely to be feedback oscillators.

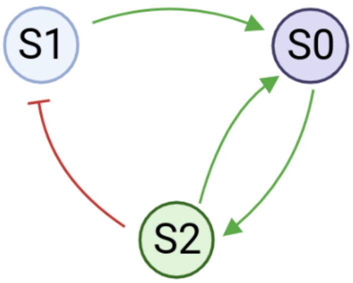


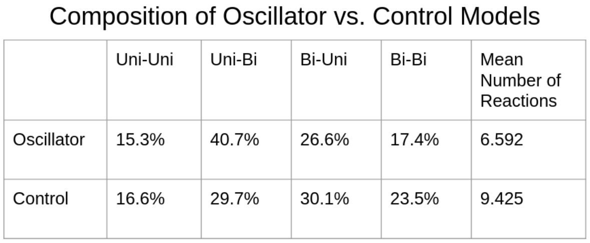
Figure 2.7: Species interactions of Network B. Green arrows indicate activation and blunt red arrows indicate inhibition. The interactions form a cycle with negative feedback suggesting that Network B is a feedback oscillator.

2.4 Discussion

Oscillating models were significantly enriched for autocatalytic reactions compared to control networks. Most oscillators are associated with unstable steady states. Destabilizing processes can be classified as 1) direct autocatalysis, 2) indirect autocatalysis (a positive feedback loop), and 3) end-product inhibition (a negative feedback loop) [19, 20]. Reactions are considered autocatalytic when the products increase the rate of reaction or when a chemical decelerates the rate of its own destruction [21]. Given the two types of oscillators, negative feedback and negative feedback coupled with positive feedback [22], it is unsurprising that autocatalytic reactions are enriched in the oscillator population as these reactions are a form of positive feedback. The portion of models containing at least one autocatalytic reaction in the oscillator and control populations were compared using a two-way chi square test. Of the 586 oscillating models, 83.1% (487 of 586) contained an autocatalytic reaction compared with 49.8% (498 of 1000) of the control models, a significant difference ($p < 0.0001$).

Given the enrichment of autocatalytic reactions, one might expect a similar enrichment of degradation reactions, reactions where one species is removed from the system (eg. $X + Y \rightarrow Y$, where X is removed from the system), to prevent species concentrations from rising to infinity. Interestingly, the presence of degradation reactions were not enriched in oscillating models. Of the oscillating model population 79.4% (465 of 586) contained at least one degradation reaction compared to 76.3% (763 of 1000) of the control models, an insignificant difference suggesting that although autocatalysis is a common characteristic in oscillating networks, degradation reactions are not necessary to prevent species concentrations from rising to infinity. Although the presence of both an autocatalytic and a degradation reaction are not necessary for oscillation, it is rare that an oscillator contains neither. Only 0.5% (3 of 586) of the oscillators analyzed had neither an autocatalytic reaction nor a degradation reaction compared to 0.2% (2 of 1000) random models, an insignificant difference.

To determine if any reaction types were enriched in the oscillator population compared to the control population, the average model compositions were compared. The average model composition was assessed as the average number of reactions of the specific type were divided by the average total number of reactions for each group. For example, there were an average of 6.592 reactions per model in the oscillator population, of which an average of 0.99 were uni-uni reactions, for an average of 15.3% uni-uni reactions in the average oscillator model (Table 2.1). Compositions were compared with permutation tests, showing that model compositions and sizes were significantly different between the oscillator and control populations (the reduced size of oscillating networks can be accounted for by the fusion of duplicate reactions). Oscillators possessed significantly more uni-bi reactions compared to random control models. This is consistent with the enrichment of autocatalytic reactions which are often uni-bi. However, autocatalytic reactions can also be bi-bi reactions, which were not enriched in oscillating models. Although bi-bi reactions were less likely to be created during evolution due to the initial settings (10%), it is somewhat surprising that bi-bi reactions were decreased in oscillatory networks given that a bi-bi reaction is slightly more likely to be autocatalytic ($\frac{4}{27}$) compared to a uni-bi reaction ($\frac{1}{9}$).

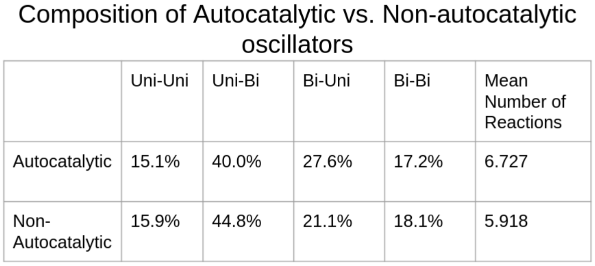


Composition of Oscillator vs. Control Models

	Uni-Uni	Uni-Bi	Bi-Uni	Bi-Bi	Mean Number of Reactions
Oscillator	15.3%	40.7%	26.6%	17.4%	6.592
Control	16.6%	29.7%	30.1%	23.5%	9.425

Table 2.1: Average reaction compositions of oscillating networks compared to non-oscillating controls.

Next, oscillators containing autocatalytic reactions (498 models) were compared to oscillators lacking them (98 models). Populations were compared by permutation test. There was a significant increase in the portion of uni-bi reactions and a significant decrease in the portion of bi-uni reactions in non-autocatalytic models as compared to autocatalytic oscillators (Table 2.2). This result is interesting given that autocatalytic reactions are either uni-bi or bi-bi and both reaction types were reduced in oscillators with autocatalytic reactions compared to oscillators without. It is possible that oscillators without a single autocatalytic reaction are achieving autocatalysis through a combination of non-autocatalytic reactions.



Composition of Autocatalytic vs. Non-autocatalytic oscillators

	Uni-Uni	Uni-Bi	Bi-Uni	Bi-Bi	Mean Number of Reactions
Autocatalytic	15.1%	40.0%	27.6%	17.2%	6.727
Non-Autocatalytic	15.9%	44.8%	21.1%	18.1%	5.918

Table 2.2: Average reaction compositions of autocatalytic oscillating networks compared to non-autocatalytic oscillating networks.

Several manually inspected oscillators had reactions with extremely high rate constants (greater than 300, whereas most rate constants were between 5 and 75). The initial range for rate constants is 0 to 50 and with each mutation, they can only increase a maximum of 15% of the current value. Although there is no upper limit for rate constants, it is nearly impossible to mutate a rate constant from the initial range to over 100 during the course of evolution. To bypass this limit and achieve high rate constants, many successful models have simply duplicated reactions during the evolutionary process. During post-processing, these duplicate reactions are fused and their rate constants summed. Fusing duplicate reactions to achieve high rate constants accounts for the observation that oscillators generally have fewer reactions than non-oscillators in this study. These high rate constant reactions can not be removed, nor can the rate constant be significantly lowered without impacting oscillation. Further investigation is needed to determine what essential function these high rate reactions seem to play in most of the oscillators included in this study.

These oscillatory models and the random control networks have been added to a database, accessible at `https://cesiumdb.herokuapp.com/`. Models can easily be selected by the number of species, the number of reactions, and the present of autocatalysis or degradation reactions. The selected models can be downloaded in a zip file containing a .txt for each model. Each .txt file contains an individual model's antimony string. If the "Download in simulatable Tellurium form" option is selected, the .txt file will also contain formatting and package imports allowing the model to be easily simulated when the file is run as a python script.

In the future, this database will be expanded to contain a variety of models with different behaviors besides oscillation. It is our hope that this database serves as a resource for the modeling community to study network behaviors and test novel software.

Funding

This project was supported by National Institute of Health grant U01 CA238475 and the National Institute of Biomedical Imaging and Bioengineering for grant P41GM109824.

Acknowledgements

We thank Lucian Smith for valuable discussions on differential evolution and model topology.

This chapter was adapted from the following publication:

Tatka, L., Luk, W., Hellerstein, Elston, T., J., Sauro, H. “Cesium: A Public Database of Evolved Oscillatory Reaction Networks.” Biosystems, vol. 224, 2023, p.104836., https://doi.org/10.1016/j.biosystems.2023.104836.

Chapter 3

This is my next paper with a clever title

Genetic algorithms (GAs), inspired by the principles of natural selection and genetics, have emerged as a powerful computational technique for solving optimization and search problems.Developed by John Holland in the 1960s, GAs mimic the process of biological evolution to solve optimization and search problems. At their core, genetic algorithms operate on a population of potential solutions, represented as individuals or ”chromosomes,” each encoding a candidate solution to the problem at hand. Through iterative generations, genetic algorithms apply mechanisms of selection, crossover, and mutation to evolve and refine the population, gradually converging towards optimal or near-optimal solutions. Selection favors individuals with higher fitness, mirroring the process of natural selection, while crossover and mutation introduce variation and diversity into the population, allowing for exploration of the solution space. By leveraging the principles of evolution, genetic algorithms offer a versatile and robust approach to solving complex problems across various domains, from engineering and optimization to biology and beyond.

Applying GAs to problems of parameter optimization is fairly straightforward as “chromosomes” can be sequences of values, easily mutated and crossed over with other strings of values. This approach becomes more complex when applied to network problems, where network topology is deeply intertwined with parameter values. In these problems, the challenges becomes innovating while preserving decent candidate solutions. Stanley and Miikkulainen addressed this problem in 2002 with NEAT (NeuroEvolution of Augmenting Topologies), a powerful algorithm used for evolving artificial neural networks (ANNs). Unlike traditional methods of neural network training, which typically involve adjusting the weights and biases of fixed network architectures, NEAT evolves both the structure and weights of neural networks simultaneously.

At its core, NEAT employs a genetic algorithm to evolve populations of neural networks. Each individual in the population represents a neural network with its own unique topology (i.e., arrangement of nodes and connections). NEAT starts with a population of simple neural networks, often with minimal structure, and evolves them over generations to solve a given task.

One of the key innovations of NEAT is its ability to handle the complex task of evolving neural network topological structures. NEAT accomplishes this by employing three main mechanisms: speciation, historical marking, and structural innovation. Speciation encourages diversity within the population by comparing individuals against similar individuals as opposed to the entire population. This shelters innovations that may be weak at first, but will develop into robust solutions over several generations. A historical record of topological innovations serves as a "chromosome" and enables meaningful crossover without computationally expensive matching procedures required in other algorithms. Lastly, NEAT allows for the introduction of new structural innovations in neural networks through mutation. New nodes and connections can be added to existing networks, providing the potential for incremental growth and improvement over successive generations.

Systems biologist are no strangers to genetic algorithms.

The similarity between ANNs and genetic regulatory networks was not lost on Dinh et al, who adapted this technique to getic

Francois and Hakim (Design of genetic networks with specified function...) were among the first to apply genetic algorithms to systems biology. They were trying to create small gene networks that functioned as either bistable switches or oscillators. Their algorithm only involved mutation and selection. For oscillators, they used the same objective function that I'm using here. Seems to have worked pretty well encourging further exploration in this space

Fujimoto et al looked at oscillaotry gene regulatory networks in the context of creating striped body pattens in developing drosophila. Well really arthorpods in general it seems. Similar approach to Francois and Hakim.

Kobayahsi et al also look at evolving oscillatory genetic networks. They fix parameter values and only allow connection rewiring to achieve oscillatory function. They wanted to look at statisticsal properties of genetic networsk as a relationship to oscillation period. THey only allow inhibitory reactions and thus all designs are essentailly extended versions of the three-gene repressilator circuit.

Deckard and Sauro expanded this concept beyond genetic regulatory networks to evolve small networks with specicfic computational capabilites such as square root and cube route calcultors. As with the previously mentioned algorithms, their alogrithm only uses mutation. They tried using crossover and it did not result in fitter offspring. They speculated that this was due to crossing over heterologous networks instead of homologous networks.

THis work was continued by Paladugu et al seeking to generate mass-action network functional modules with specific behaviors, including oscillators. The evolution algorithm used the addition or deletion of reactions or modification of rate constants as mutation operators. This algorithm avoided crossover as the authors argued that it would be disruptive.

Marchisio and Stelling argued that brute force optimization algorithms lack efficiency and that implementing some element of rational design could spped up automatic design of genetic networks implementing boolean logic gates. They separate structural and parameter optimziation by first generating several possible circuits and then ranks feasibility based on the structure. THen the best solutions undergo parameter optimziation. Separating structural design from parameters optimziation certainly saves computation time, but it relies on the existence of a rational solution, and is better suited for genetic networks, where these attributes are more easibly separated.

3.1 Existing Software

Although there are numerous studies implemting genetic algorithms in systems biology applications, few software packages exist to aid in this pursuit. In 2008, Kratz and Krishna developed GeNESis, a C based software with GUI to enable evolution of gene regulation networks. Its purpose was to help researchers undersatnd the evoltuionary behavior of populations of genetic regulatory networks. Connections were represented by binary strings, enabling the use of crossover. The URL to this tool is currently inactive.

Chandran and Sauro devloped a C library with basic functions for evolution of biological networks including genetic networks, protein networks, and mass-action networks. Although highly versatile, the library requires some C programming knowledge in order to customize the evolutionary algorithm.

—-

Oscillatory behavior is useful in a variety of biological processes including p53, circadian rhythm, and cell cycle. But also, you might want to generate oscillators as a component in a syntehtic biology ciruit. But really, this isn't about oscillators, that's just the test case. You might want a certain behavior and don't know how to build a network to get it. Well This algorithm is for you!

Other people have done similar things, but this is different. It starts with the concept of genetic algorithms, way back in 1975. These are algorithms to come up with optimization solutions. The algorithms kind of look like biologic evolution. You have a bunch of individuals, the fitter individuals share their genes and reproduce, less fit individuals die off, sometimes mutations occur and these can be either good or bad. Since then, the concept of genetic algorithms has been thoroughly explored, but here we present something slightly new. And that's how I'm going to get my PhD lol.

This all starts with the NEAt algorithm which is a way of evolving neural networks to do stuff. What's interesting about this is they claim to have meaningful crossover and they use speciation to protect innovation. Dinh et al somehwat adapted this appraoch to genetic circuits, but those are a much easier problem because they just have activation and inhibition. Here, I adapt this algorthim to mass-action networks. That's way harder because there is way more complexity. In the first case, you just have an edge that connects two nodes and a weight. In the second, you add that the edge can either be activating or inhibiting, but here we add that the edge could connect multiple nodes (in the case of everything beyond a uni-uni reaction) and the reaction weight depends on what nodes it connects. More on that later.

I need to go over a bunch of times that this was done for a similar thing. Just as a quick review

—————————————— Some review stuff

Evolutionary design of oscillatory genetic networks (Kobayashi 2010) Wanted to evolve genetic oscillatory networks. Get a large population of them and then mine them for statistics about oscillators and network motifs. They have fixed parameter values though and only change connections (?). Looks like they are also trying to evolve oscillators with specific frequencies. They have a cost function which looks at periodicity

3.2 Methods

3.3 Encoding

The main difference between ANNs and chemical reaction networks is that edges (reactions) can connect more than two nodes (chemical species) and that weights represent biochemical rate constants. Also activation is different but idk if that's relevant. A chemical reaction network is composed of a set of reactions and initial concentrations. A reaction consists of one or two reactants, one or two products, a rate constant, and a boolean indicating if the reaction is active or not.

Chapter 4

Background: *Adapting Credibility Standards to Computational Systems Biology*

4.1 Introduction

As computing power rapidly increases, computational models become more intricate and an increasingly important tool for scientific discovery. In systems biology, where the amount of available data has also expanded, computational modeling has become an important tool to study, explain, and predict behavior of biological systems. The scale of biological models ranges from subcellular components [23] to entire ecosystems [24]. Modeling paradigms include mechanistic models, rule-based systems, Boolean networks, and agent-based models [25]. This review will focus on mechanistic models of subcellular processes.

The Food and Drug Administration (FDA) defines model credibility as "the trust, established through the collection of evidence, in the predictive capability of a computational model for a context of use" [26]. Model credibility is important in systems biology as models are used to guide experiments or to optimize patient treatment. This is particularly important given the increasing scale and intricacy of models. Reproducibility, the ability to recreate a model and data *de novo* and obtain the same result [27], is directly connected to credibility, but even reproducibility remains a challenge. It was recently discovered that 49% of published models undergoing the review and curation process for the BioModels [28] database were not reproducible primarily due to missing materials necessary for simulation, the availability of the model and code in public databases, and lack of documentation [29]. With some extra effort, an additional 12% of the published models could be reproduced. A model that cannot be reproduced is not credible.

Due to the increasing importance of computational models in scientific discovery, the National Aeronautics and Space Administration (NASA), the FDA, and other regulatory bodies have developed standards to assess the credibility of models [30, 26, 31]. These standards are somewhat vague and generally qualitative to accommodate the broad scope of models in these fields. However, mechanistic models in systems biology are relatively narrow in scope and are supported by a variety of standards for model encoding, annotation, simulation, and dissemination potentially enabling the development of a credibility standard for mechanistic systems biology models.

In this chapter, we discuss current systems biology modeling standards that could aid in the development of credibility standards, examine existing credibility standards in other scientific fields, and propose that current standards in systems biology and other fields could support the development of a credibility standard for mechanistic systems biology models.

4.2 Current Standards in Systems Biology

Klipp et al. describe standards as agreed-upon formats used to enhance information exchange and mutual understanding [32]. In the field of systems biology, standards are a means to share information about experiments, models, data formats, nomenclature, and graphical representations of biochemical systems. Standardized means of information exchange improve model reuse, expandability, and integration as well as allowing communication between tools. In a survey of 125 systems biologists, most thought of standards as essential to their field, primarily for the purpose of reproducing and checking simulation results, both essential aspects of credibility [32].

A multitude of standards exist in systems biology for processes from annotation to dissemination.

Although there is currently no widely used standard for model credibility, the development of this standard is likely to depend on existing systems biology standards, just as standards for model simulation are dependent on standards for model encoding. This section will summarize current standards relevant to model credibility including standards for ontology, encoding, simulating, and disseminating models. Although standards also exist for graphical representation of systems biology models (SBGN) [33] and representation of simulation results (SBRML) [34], these will not be discussed here as they are less relevant to the future implementation of model credibility standards.

4.2.1 Model Representation

Having a commonly understood language for describing a model is essential in exchange, reproducibility, credibility. Without a common language to describe models, they cannot be simulated across different platforms or freely shared. For this reason, systems biology model representation has become standardized using XML-based languages SBML [32, 35], CellML [36], and BioPAX [37]. NeuroML [38], similar to SBML and CellML, is used to represent neuronal models, but is beyond the scope of this proposal.

SBML

The most widely used model format is SBML (Systems Biology Markup Language) [39, 40, 32, 35]. SBML is a XML-based language for encoding mathematical models that reproduce biological processes, particularly biochemical reaction networks, gene regulation, metabolism, and signaling networks [39, 41]. SBML encodes critical biological process data such as species, compartments, reactions, and other properties (such as concentrations, volumes, stoichiometry, and rate laws) in a standardized format. Annotations can also be stored in the SBML format. With its support by over 200 third party tools and its ability to easily convert to other model formats, SBML is the de facto language for systems biology models [41, 35].

SBML models are composed of entities, such as species, located in containers that can by acted upon by processes that create, destroy, or modify [42]. Other elements allow for the definition of parameters, initial conditions, variables, and mathematical relationships. The SBML language is structured as a series of upwardly compatible levels, with higher levels incorporating more powerful features. Versions describe the refinement of levels. Most recently, SBML level 3 introduced modular architecture consisting of a set of fixed features, SBML level 3 core, and a scheme for adding packages that augment the core functionality. This allows for extensive customization of the language while enabling reuse of key features. Currently, eight packages are part of the SBML 3 standard. These packages extend the capability of SBML such as enabling descriptions of uncertainties in terms of distributions [43], allowing for the encoding, exchange, and annotation of constraint-based models [44], rendering visual diagrams [45], among many others [42].

CellML

Similar to SBML but broader in scope, CellML is also an XML-based language for reproducing mathematical models of any kind, including biochemical reaction networks [46]. CellML models do not encode biological information explicitly in the model, but instead consist of mathematical formulations of biological processes [47]. This feature of the CellML language increases flexibility enabling the description of a wide variety of biological processes. CellML models are composed of several interconnected components [48] with each component containing at least one variable that is associated with physical units. This enables CellML processors to automatically check equations for dimensional consistency.

Both CellML and SBML use almost identical mathematical expressions in MathML, an international standard for encoding mathematical expression using XML [49]. CellML explicitly encodes all mathematics, such as ODEs [50]. It is more versatile than SBML, capable of describing any type of mathematical model. SBML defines reaction rates, which can be used to build rate rules and ODEs [51]. There is more third party support for SBML and it is a semantically richer language compared to CellML [35, 50].

BioPAX

BioPAX (Biological Pathway Exchange) is an ontology, a formal systems of describing knowledge that structures biological pathway data making it more easily processed by computer software [37].

It describes the biological semantics of metabolic, signaling, molecular, gene-regulatory, and genetic interaction networks [37]. Whereas SBML and CellML focus on quantitative modelling and dynamic simulation, BioPAX concentrates primarily on quantitative processes and visualization [37, 52].

BioPAX contains one superclass, Entity. Within the Entity superclass, there are two main classes: PhysicalEntity and Interaction. PhysicalEntity describes molecules, including proteins, complexes, and DNA, while the Interaction class defines reactions and relationships between instances of the PhysicalEntity class. Interactions can be either Control or Conversion, both of which are divided into several more detailed subclasses [37, 52]. Like SBML, BioPAX is released level-wise with level 1 describing interactions, level 2 supporting signaling pathways and molecular interactions, and level 3 enabling the description of gene-regulatory networks and genetic interactions.

4.2.2 Annotation

As models grow more numerous and complex, there is an increasing need for a standardized encoding format to search, compare, and integrate them. While standards such as SBML and CellML provide information on the mathematical structure of a model, there is no information as to what variables and mathematical expressions represent. Simple textual descriptions of these representations are subject to errors and ambiguity and require text-mining for computational interpretation [53]. Standardized metadata annotations address these issues by capturing the biological meaning of a model's components and describing its simulation, provenance, and layout information for visualization. The use of annotations improves model interoperability, reusability, comparability and comprehension [54]. Annotations are enabled by systems biology specific ontologies [47] which define a common vocabulary and set of rules to unambiguously represent information [55].

To avoid accounting for a variety of annotation formats and approaches, standard annotation protocols are necessary [54]. However, despite the numerous standards and tools, annotation remains a challenge. For example, the ChEBI database [56] has approximately 1,000 annotations for glucose. While more than one entry for each annotation can serve a purpose (some users may prefer to be more abstract in their annotations), this adds to the challenge of defining the purpose of a model and, therefore its credibility. Additionally, annotations can be obsolete, inappropriate or incorrect, or provide insufficient information. Evaluating the quality of annotations would be essential in any credibility assessment for systems biology models. Some tools already exist for this purpose, such as SBMate [57], a python package that automatically assesses coverage, consistency, and specificity of semantic annotations in systems biology models.

MIRIAM

MIRIAM (Minimum Information Requested in the Annotation of Biochemical Models) [58] was developed to encourage the standardized annotation of computational models by providing guidelines for annotation. The MIRIAM guidelines suggest that model metadata clearly references the relation documentation (e.g. a journal article), that the documentation and encoded model have a high degree of correspondence, and that the model be encoded in a machine-readable format (such as SBML or CellML). Annotations should also include the name of the model, the citation for its corresponding journal article, the contact information of the creators and date of creation, as well as a statement about the terms of distribution. Additionally, models should have accurate annotations that unambiguously links model components to corresponding structures in existing open access bioinformatics resources. The referenced information should be described using a triplet, data collection, collection-specific identifier, optional qualifier and expressed as a Uniform Resource Identifier (URI), a unique sequence of characters that identifies a resource used by web technologies [59]. The optional qualifier field is used to describe relationships between the model constituents and the piece of knowledge with language such as “has a”, “is a version of”, “is homologous to”, etc.

Systems Biology Ontology (SBO)

SBO (Systems Biology Ontology) describes entities used in computational modeling [60, 53]. It defines a set of interrelated concepts used to specify the types of components specified in a model and their relationships to one another. Annotation with SBO terms allows for unambiguous and explicit understanding of the meaning of model components and enables mapping between elements of different models encoded in different formats [53]. Both SBML and CellML support annotation with SBO terms. SBML elements contain an optional sboTerm attribute [53, 51, 47].

OMEX

In order to harmonize the metadata annotations across models encoded in various formats, Gennari *et al.*, with the consensus of the COMBINE (Computational Modeling in Biology Network) community, developed a specification for encoding annotations in Open Modeling Exchange (OMEX)-formatted archives. The specification describes standards for model component annotations as well as for annotation at the model-level, and archive-level [54]. The specification describes annotation best practices and addresses annotation issues such as composite annotations, annotating tabular data and physical units, as well as provides a list of ontologies relevant to systems biology. Implementation of these specifications is aided by LibOMexMeta, a software library supporting reading, writing, and editing of model annotations. It uses Resource Description Framework [61] (RDF), an XML-based standard format for data exchange on the web, for representing annotations. It also makes use of several standard knowledge resources describing biology and biological processes such as ChEBI [56], a dictionary of small chemical compounds, and UniProt [62], a database of protein sequence and functional information.

Annotation in CellML and SBML

Both CellML and SBML have their own annotation protocols based on RDF [47]. The CellML language uses its own ontology for model annotation, a necessity due to the flexibility of the language [46]. The CellML Metadata Specification was developed parallel to the CellML language [47]. CellMLBiophysical/OWL ontology is composed of two categories: physical and biological [47]. The physical ontology describes physical quantitative information and concepts captured in the model's mathematical expressions. It is subdivided into processes, such as enzyme kinetics, ionic current, and rate constants, and physical entities, such as area, concentration, volume, and stoichiometry. The biological ontology provides description for processes, entities, the role of an entity in relation to a process, and the specific location of the entity in a biological system. Bioprocesses are divided into three subclasses: biochemical reactions, transport, and complex assembly. Biological entities include proteins, small molecules, and complexes. The biological roles subclass is composed of modifiers, reactants, and products.

SBML also facilitates MIRIAM compliant annotation using RDF) [63, 61]. Annotations use BioModels.net [64] qualifier elements embedded in XML form of RDF [65]. Each annotation is a single RDF triple consisting of the model component to annotate (subject), the relationship between the model component and the annotation term (predicate), and a term which describes the meaning of the component (object). These terms come from defined ontologies, such as SBO [60]. RDF annotation is supported by the software libraries libSBML [66] and JSBML [67].

4.2.3 Simulation and Parameter Estimation

Information about a model alone is insufficient to enable efficient reuse. A variety of advanced numerical algorithms and complex modeling workflows make the reproduction of simulations challenging. Many modelers reproduce simulations by reading the simulation description in the corresponding publication [68]. This is time consuming and error prone and the published description of a simulation is often incomplete or incorrect. For these reasons, it is essential to define and include information necessary to perform all simulations.

MIASE

Guidelines for the Minimum Information About a Simulation Experiment (MIASE) were introduced to specify what information should be provided in order to correctly reproduce and interpret a simulation [68]. MIASE is a set of rules that fall into three categories: information about the model used in the simulation experiment must be listed in a way that enables reproduction of the experiment; all information necessary to run any step of the experiment must be provided; all information needed to post-process data and compare results must be included. Along with MIRIAM [58] guidelines, MIASE compliance guarantees that the simulation experiment is true to the intention of the original authors and is reproducible.

KiSAO

KiSAO (Kinetic Simulation Algorithm Ontology) is an ontology used to describe and structure existing simulation algorithms [69, 53]. It consists of three main branches, each with several subbranches. The first branch is Kinetic simulation algorithm characteristics, such as the type of system behavior or type of solution. The second in the kinetic simulation algorithm such as Gillespie or accelerated stochastic simulation. The third branch is kinetic simulation algorithm parameters which describe error and granularity, among other characteristics.

SED-ML

Simulation Experiment Description Markup Language (SED-ML) is software independent, XML-based format for encoding descriptions of simulation experiments and results [70, 71]. To help modelers comply with MIASE rules, SED-ML describes the details of simulation procedures, including what datasets and models to use, which modifications to apply to models, which simulations to run on each model, how to post-process data, report, and present results can all be encoded [68]. Each algorithm mentioned in a SED-ML file must be identified by a KiSAO term [53]. PhraSED-ML was developed to enable modelers to encode human readable SED-ML elements without the use of specialized software [72].

PEtab

Parameter estimation is common in modeling and simulation, which often requires running multiple simulations to scan the suitability of several parameter sets. Although many parameter estimation toolboxes exist, they each use their own input formats. The lack of a standardized format makes it difficult to switch between tools, hindering reproducibility [73]. PEtab is a parameter estimation problem definition format consisting of several files containing information necessary for parameter estimation, including the model (in SBML format), experimental conditions, observables, measurements, parameters, and optional visualization files [73]. A final PEtab problem file links all other files to form a single, reusable, parameter estimation problem. Following the success of PEtab, parameter estimation functionality was added to SED-ML [71].

4.2.4 Dissemination

Model reproducibility best practices describe dissemination as an essential part of reproducibility [74]. Sharing all model artifacts and documentation on open-source repositories allows independent researchers to reproduce, reuse, and understand the model. Several guidelines and archive formats have been developed to ensure that all relevant information necessary to reproduce a modeling result is easily accessible to the public.

MIRIAM Curation Guidelines

In addition to annotation guidelines, MIRIAM also provides guidelines for model curation, the process of collecting and verifying models. The aim of MIRIAM guidelines is the ensure that model is properly associated with a reference description (e.g. a journal article) and that it is consistent with that reference description, meaning that it reflects the biological process listed in the reference description. The model must be encoded in a public, machine-readable format such as SBML or CellML and comply with the associated encoding standard. The encoded model must be simulatable, including quantitative values for initial conditions, parameters, and kinetic expressions, and must reproduce relevant results when simulated [58].

FAIR

More recently, the FAIR guidelines were published to improve the ability of computers to **F**ind, **A**ccess **I**nteroperate, and **R**euse models [75] with minimal human interaction. FAIR defines characteristics that data resources should possess to assist with discovery and reuse by third-parties. Unlike most data management and archival guidelines, FAIR is a set of high-level, domain-independent guidelines that can be applied to a variety of digital assets. Each element of the FAIR principle is independent.

For a model to be "findable," it should be easy to find for both humans and computers. This requires describing and annotating data and metadata with unique identifiers that are registered or indexed in a searchable resource. Once the user finds the relevant model, it should be accessible: data and metadata should be retrievable by their identifiers using standard communications protocol. Metadata should remain accessible even when data are no longer available. Interoperability refers to the integration with other data and the ability to operate with various applications and workflows. This is enabled by the use of broadly applicable languages for model representation and annotation. As the ultimate goal of FAIR is to enable the reuse of data, the guidelines dictate that data and metadata should be associated with detailed provenance, meet domain-specific community standards (such as COMBINE archive format described below), and released with clear and accessible data usage license.

COMBINE Archives

COMBINE (COmputational Modelling in BIology NEtwork) is a formal entity that coordinates standards in systems biology. To assist in this coordination, a MIRIAM compliant system for sharing groups of documents regarding a model was developed called the COMBINE Archive [76]. The archive is encoded in OMEX (Open Modeling Exchange format) and the archive itself is a "ZIP" file. A COMBINE archive could contain files in several different standard formats including SBML, SBOL, and SED-ML among others. Additionally, every COMBINE Archive contains at least one file titled *manifest.xml* that contains a list of all the files comprising the archive and describing their locations. An archive also may contain a metadata file, ideally conforming to MIRIAM and MIASE guidelines. The inclusion of all necessary protocols and data needed to implement a model enables distribution of models via a single file encouraging reuse and improving reproducibility [77].

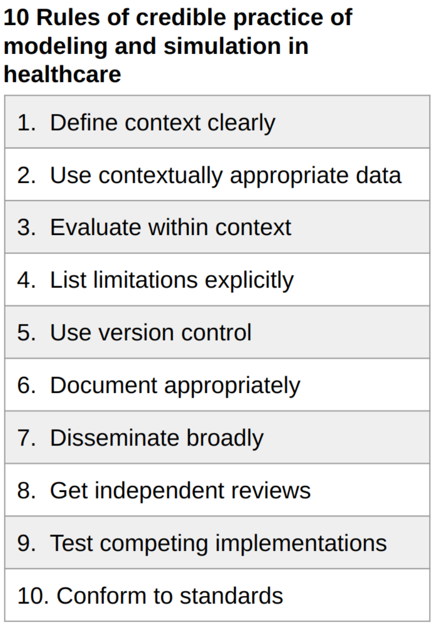
4.3 Credibility Guidelines in Systems Biology

Although no standard for model credibility in systems biology exists, there are general guidelines aimed at improving the trustworthiness of models developed by the Committee on Credible Practice of Modeling and Simulation in Healthcare, a group formed by the U.S. National Institutes of Health [78]. The purpose of these guidelines is to encourage the credible use of modeling and simulation in healthcare and translational research. These guidelines are qualitative and share many components with best practices for reproducibility. The term "credible" was defined as "dependable, with a desired certainty level to guide research or support decision-making within a prescribed application domain and intended use; establishing reproducibility and accountability.” These guidelines are qualitative and intended to cover a variety of modeling approaches and applications within the biomedical context (figure 4.1).

The credibility of a model should be evaluated within the model’s context of use [78]. To this end, the guidelines recommend using contextually appropriate data and evaluating the model (performing verification, validation, uncertainty quantification and sensitivity analysis) with respect to the context in which the model will be used. Any limitations should be listed explicitly.

Borrowing from software engineering best practices, the guidelines also recommend the use of version control to track model and simulation development as well as extensive documentation of simulation code, model mark-up, scope and intended use. Models should also include guides for developers and users and conform to domain-specific standards [78].

Different simulation strategies should tested to ensure that the results and conclusions are similar across various tools and methods. All modeling components such as software, models, and results should be reviewed by third party users and developers and disseminated widely.



10 Rules of credible practice of modeling and simulation in healthcare

1. Define context clearly
2. Use contextually appropriate data
3. Evaluate within context
4. List limitations explicitly
5. Use version control
6. Document appropriately
7. Disseminate broadly
8. Get independent reviews
9. Test competing implementations
10. Conform to standards

Figure 4.1: The Committee on Credible Practice of Modeling and Simulation in Healthcare 10 rules of model credibility.

4.4 Qualitative Credibility Assessment in Other Modeling Fields

NASA and the FDA also have a keen interest in producing well-documented and credible models for the purpose of making critical decisions. However, modeling and simulation tasks in these institutions are far broader compared to systems biology. NASA models range from the analysis of individual parts to orbits and spacecraft while models submitted to the FDA include medical devices and pharmacokinetics. Due to the wide variety of modeling tasks relevant to NASA and the FDA, credibility guidelines in these institutions are general, largely qualitative, and do not prescribe specific tests.

4.4.1 NASA Credibility Assessment Scale

After the loss of the Columbia Space Shuttle and its seven crew members in 2003, NASA significantly increased its focus on quantitative and credible models. The misuse of an existing model and the reliance on engineers' judgment led to the false conclusion that shuttle reentry would not be affected by a small hole in the heat shield caused by a debris strike during takeoff [79, 80]. The lack of quantifiable uncertainty and risk analysis in the report to management ultimately led to the shuttle's disintegration [81]. Since then, NASA has developed extensive modeling and simulation standards including the Credibility Assessment Scale [30] (CAS) (figure 4.2).

Each model credibility standard described here emphasizes assessments be made within a specific context of use, the specific role and scope of the model and the specific question of interest that the model is intended to help answer [26]. The judgment error that ultimately led to the Columbia Space Shuttle disaster was partially due to the use of a modeling software far outside the intended context of use, leading to incorrect predictions and over-reliance on engineer's judgment [79]. In addition to specifying the scope and question of interest, the context of us should also describe how model outputs will be used to answer the question of interest and whether other information, such as bench-testing, will be used in conjunction with the model to answer the question of interest [26]. The standards described here, from various institutions such as NASA and the FDA, all specify that credibility is to be evaluated within a specific context of use.

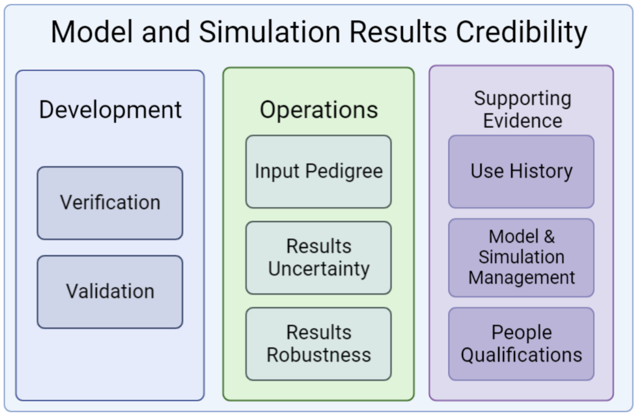


Figure 4.2: Categories of the NASA Credibility Assessment Scale (CAS)

NASA's CAS is intended to help a decision-maker evaluate the credibility of specific modeling and simulation results and to identify aspects of the results that most influence credibility [82, 79]. The credibility assessment process can be viewed as a two part process: first the modeler conveys an assessment of the results, then a decision maker infers the credibility of these results. The CAS standard consists of eight factors grouped into three categories [30]: development, operations, and management. Each of the eight factors is scored on a scale of 0-4 with guidelines for each numeric score. These factors were selected as they were considered to be the most essential, sufficiently independent of one another, and could be objectively assessed. While the primary concern is the score for each individual factor, the secondary concern is the score of the overall model, which is the minimum score of the eight subfactors.

The model and simulation (M&S) development category consists of subsections verification and validation [30]. Scoring in these subcategories assess the correctness of the model implementation, the numerical error and uncertainty, and the extent to which the M&S result matches reference data. If numerical errors for important features are "small" and if results agree with real-world data, the highest score of 4 is awarded for these factors.

The second category, M&S operations, consists of three factors: input pedigree, results uncertainty, and result robustness [30]. Input pedigree describes the level of trust in the input data, where input data that accurately reflects real-world data receiving the highest score. The results uncertainty category earns the highest score if non-deterministic numerical analysis is performed. Result robustness high scores are achieved by including sensitivity analysis for most parameters and identifying key sensitivities.

Model and simulation management, the third category, is less technical, containing the factors use history, M&S management, and people qualification [30]. Use history scores the highest score if the model has previously been used successfully and meets de facto standards. For example, a model used for finite element analysis (FEA) would be required to meet FEA standards and codes for the type of object being modeled. M&S management refers to the maintenance and improvement of the model with continual process improvement receiving the highest score of 4. The people qualification category assesses the experience and qualifications of those constructing, maintaining, and using the model where personnel with extensive experience with the model and best practices scoring the highest.

Although these categories were chosen, in part, due to their ability to be objectively assessed, there is still a significant subjective component of the scoring process. It is acknowledged that different decision makers may assign different degrees of credibility to the same model and different decisions may require different levels of credibility. The CAS serves as a template to assess and clearly communicate risks to decision-makers. Additionally, it can be useful in measuring model development progress or in identifying areas where improvement is most needed [79].

4.4.2 Credibility Standards for Medical Models

In addition to systems biology, computational models are also becoming essential tools in biomedical applications such as drug discovery [83], pharmacokinetics [84], and medical devices [85]. Credibility is essential in biomedical modeling, particularly in cases where models influence patient treatment or regulatory approval of a device or drug. Both the FDA and the European Medicines Agency have developed standards guiding model credibility for the purposes of regulatory approval. As with NASA, these guidelines are broad and qualitative due to the broad scope of biomedical modeling. Before the FDA began formalizing guidelines for model credibility, the American Society of Mechanical Engineers (ASME) issued Verification and Validation (V&V) 40 for assessing credibility of computational models in medical device applications [86]. However, this standard assumes the ability to perform traditional validation activities such as comparing model predictions to well-controlled validation experiments [26], a task which can be unfeasible with some biomedical models. Many models used in regulatory submissions are supported by many sources of evidence beyond traditional validation experiments including clinical trials and population-level validation. Recognizing this fact, FDA modeling credibility guidelines expand on ASME V&V 40 concepts to provide a more general framework for assessing a wider variety of models.

ASME V&V 40

The ASME developed the V&V 40 standard in 2012 as a means of describing verification and validation activities in the modeling and simulation of medical devices [86]. Like the NASA CAS, V&V 40 focuses on context of use, model risk, and the establishment of credibility goals prior to any credibility assessment. The context of use addresses the specific role of the model in addressing the question of interest. Model risk is then assessed based on the possibility that the model may lead to incorrect conclusions resulting in adverse outcomes. After the establishment of credibility goals, verification and validation take place.

Of particular relevance to modeling in systems biology are the descriptions of code and calculation verification found in V&V 40. Verification seeks to determine if the model is built correctly. More specifically, code verification aims to identify any errors in the source code and numerical algorithms. This can be done by comparing output from a model to benchmark problems with known solutions [86]. Calculation verification estimates the error in the output of a model due to numerical methods. Output errors can include discretization errors, rounding errors, numerical solver errors, or user errors. Calculation verification is complete when it is demonstrated that errors in the numerical solution are minimized to the point that they are not corrupting the numerical results [86].

Validation assesses how well the computational model represents reality. Validation activities might include comparing the model's behavior to the biological features of the real phenomenon by comparing results to *in vitro/in vivo* benchmark experiments. Validation also includes uncertainty quantification and sensitivity analysis. Uncertainty quantification refers to the estimation of how stochastic error in the input propagates into the model's output. Sensitivity analysis is a post-hoc examination of the results of the uncertainty quantification to evaluate which elements most influence output variability [86].

Unlike the NASA CAS, V&V 40 does not describe the quality of evidence needed to prove a model credible and lacks an objective scoring system necessary for implementing “cut-offs” of credible versus non-credible models, or for comparing the credibility of multiple models.

FDA Guidance on Computational Model Credibility in Medical Devices

Based on the V&V 40 standard, the FDA released guidance on assessing credibility for models of medical devices [26]. This guidance expands V&V 40 to include other forms of credibility evidence beyond traditional verification and validation exercises. Applicable to physics-based, mechanistic, or other first-principles-based models of medical devices, these guidelines consist of ten categories broadly divided into code verification, calculation verification, and validation. The code verification category is taken directly from V&V 40 described previously.

The calculation verification guideline extends the V&V 40 by detailing several methods to verify that the model produces the intended output. For example, the model results can be compared with the same data used to calibrate the model parameters. Broader evidence in support of the model, but perhaps without a specific context of use are also acceptable. A model can also be verified using *in vitro* or *in vivo* experiments either within the context of use, or within conditions supporting a different context of use. These techniques can also be used for validation evidence.

Validation assesses the model’s ability to reproduce real-world behavior. In addition to the methods described for calculation verification, validation can also include population-based evidence, statistical comparisons of model predictions to population-level data such as the results of a clinical trial. Credibility is also supported by emergent model behavior, the ability of a model to reproduce real-world phenomena that were not pre-specified or explicitly modeled, as well as general model plausibility, that model assumptions, input parameters, and other characteristics are deemed reasonable based on scientific knowledge of the system modeled.

Unlike the NASA Credibility Assessment Scale, these FDA guidelines are sets of nonbinding recommendations. Additionally, no scoring or suggested quality measures of FDA credibility factors are included making quantitative analysis of credibility impossible.

EMA Guidelines for PBPK Models

Of particular relevance for the field of systems biology is The European Medicines Agency's (EMA) Guideline on the Reporting of Physiologically Based Pharmacokinetic (PBPK) Modeling and Simulation issued in 2018 [87, 31] . PBPK models are mathematical models that simulate the concentration of a drug over time in tissues and blood. With the rise in regulatory submissions that include PBPK models that rely on specialized software programs, this guideline provides detailed advice on what to include in a PBPK modeling report.

The standard dictates the necessary information to describe and justify model parameters. Like the FDA standard, modelers are required to submit any assumptions made when assigning parameters and to document the sources of any literature-based parameters. Additionally, modelers must perform a sensitivity analysis for parameters that are key to the model (those that significantly influence the outcome) and list any parameters that are uncertain.

The submission must include the simulation results as well as the files used to generate the final simulations in both tabular and executable format. This requirement is shared with reproducibility standards already in place for systems biology in the COMBINE Archive standard as well as described in systems biology modeling reproducibility best practices [74].

The predictive performance of the model must also be evaluated. That is, its ability to recapitulate observed pharmacokinetics. This requirement is also mentioned in the FDA guidelines.

Lastly, the guideline requires a discussion of uncertainty and confidence in the model. Although described more qualitatively in the EMA standard, this requirement is shared by NASA's CAS, V&V 40, FDA credibility guidelines, and best practices for reproducible modeling in systems biology [74].

4.4.3 Current Tools for Systems Biology Model Testing

Although there is no credibility standard in systems biology modeling, some tools provide automated model testing. Although these tools were not developed explicitly to assess credibility, many of the factors they test for could be considered aspects of credibility. Future model credibility assessments could aspire to the level of quantification and automation these tools offer.

MEMOTE

MEMOTE (MEtabolic MOdel TEsts) is an open-source Python software that automatically tests and scores genome-scale metabolic models [88]. MEMOTE offers a web interface and command line interface where SBML files can be uploaded and analyzed, and ultimately scored. The tests check that a model is annotated according to the MIRIAM standard, that components are described using SBO terms, and that the model is properly constructed using the relevant SBML package, SBML-FBC [88, 89]. Basic tests check for the presence of relevant components, charge information, and metabolite formulas. Biomass tests check that biomass precursors are produced and that the growth rate is non-zero. Stoichiometry tests test for inconsistency, erroneously produced energy metabolites, and reactions that are permanently blocked. A numeric score is output after testing indicating the extent to which a model conforms to these standards.

MEMOTE is designed to assess genome-scale metabolic models and largely includes tests that are specific to this model subset. Although a high MEMOTE score is likely to be indicative of model quality and reproducibility, it is not an assessment of credibility. A credible model will likely have a good MEMOTE score, but a good MEMOTE score does not necessarily indicate a credible model. However, the quantitative and automated nature of MEMOTE allows for quickly gauging model quality, comparing models, and the iterative improvement of metabolic models.

FROG Analysis

Similar to MEMOTE, the COMBINE community has recently developed FROG analysis, an ensemble of analyses for constraint-based models to generate standardized numerically reproducible reference datasets [90]. Results from constraint-based models are often communicated as flux values and there are often multiple solutions for a single model. As such, results cannot be used to gauge reproducibility. The COMBINE community outlined a list of outputs and results of flux balance analysis that are numerically reproducible and can be used for curation, known as FROG reports. FROG reports can be used in the BioModels [91, 28] curation process to assess reproducibility.

FROG analysis consists of **F**lux variability analysis, **R**eaction deletion, **O**bjective function values, and **G**ene deletion fluxes. Flux variability analysis tests that the maximum and minimum fluxes are reproducible using different software tools. The objective function value for a defined set of bounds should be reproducible. The systematic deletion of all reactions or all genes, one at a time, should provide comparable reference results. Currently four tools support the generation of FROG reports. Web-based tools include fbc_curation [90], CBMPy MOdel Curator [92] (both of which are also available as command line tools), and FLUXER [93]. fbc_curation_matlab is a command-line tool and exports results in COMBINE archive format [94].

Unlike MEMOTE, FROG analysis produces a report in lieu of a single numerical score.

4.5 Discussion

When standards are established, tests can be established to assess the extent to which a model conforms to that standard. With the development of standardized quantitative metrics (as opposed to qualitative guidelines such as those discussed previously), models can be constructed to meet minimum quality requirements lending credibility to those models and allowing for easy comparison across models [95].

The difficulty in developing these quantitative metrics is that the characteristics of an ideal biomodel must be known and expressed concisely. Existing standards in systems biology seek to address the first point by outlining what information is necessary to completely define and reproduce a model as well as the format in which that information is to be presented. However, a model could meet all existing standards and not be credible. For example, a model could be fully defined in SBML with extensive annotations, be reproducible, properly formatted for dissemination with SED-ML files describing all simulations. Despite meeting these standards, this hypothetical model could produce negative concentrations when simulating, clearly indicating that the model is not credible. Additional metrics and standards must be established to adequately assess credibility. These metrics might include the relative concentration of floating species or the shape of response curves.

Hellerstein *et al.* note that several issues in biomedical modeling are analogous to problems faced in software development and propose that software development best practices might be translated to improve modeling in systems biology [96]. Of particular interest is software testing, which can be considered a form of credibility assessment. These tests aim to ensure the correctness, reliability, and availability of software, all characteristics that are also essential in systems biology model credibility.

Software tests can be divided into two categories, which may also be applicable in systems biology modeling: (i) black-box testing and (ii) white-box testing. Black-box software testing assesses the behavior of the code and does not deal with implementation. For systems biology model credibility assessment, black-box credibility indicators might be that the model accurately predicts observed data. White-box testing evaluates the internal workings of a software project or model. The absence of errors, such as undefined parameters, typos, or unused species, might serve as white-box credibility indicators.

4.6 Conclusion

Although many reproducibility standards are in use to simplify assessing reproducibility, there are no standards and scoring systems for model credibility in systems biology. Unlike institutions such as NASA and the FDA, which deal with models spanning a broad scope of applications and scales, systems biology is focused on the modeling of cellular processes. This narrow scope, combined with the variety of standards already in use, makes systems biology models well-suited for a credibility standard.

A quantitative credibility scoring system would be particularly useful and enable comparing the credibility of different models and guide the development of more credible models. Credibility metrics could be published alongside models to indicate the trustworthiness of results and allow users to make informed decisions about reusing models.

Systems such as MEMOTE demonstrate that model standards and model quality indicators can be automatically quantitatively scored enabling iterative improvement during the development phase. More challenging is further developing standards to express characteristics, both quantitative and qualitative, that make a model credible. Current modeling standards in other scientific fields emphasize assessing credibility in the model's context of use. This poses a challenge for automating credibility assessment in systems biology modeling as more or less rigor may be required to achieve a sufficiently credible model depending on the intended use of the model. It may prove useful to develop a manual, semi-quantitative scoring systems, such as NASA's Credibility Assessment Scale prior to attempting to implement a fully quantitative and perhaps automated credibility scoring system for systems biology models.

Funding

This work was supported by NIH Imaging and Bioengineering (NIBIB) award P41GM109824, and the National Science Foundation award 1933453.

This chapter was adapted from the following publication:

Tatka, L., Smith, L., Hellerstein, J., Sauro, H. "Adapting Modeling and Simulation Credibility Standards to Computational Systems Biology." Journal of Translational Medicine, submitted.

Chapter 5

Overview of Proposed Research: *Software for Automated Credibility Assessment in Systems Biology Models*

5.1 Introduction

As computational models grow in number and complexity, they become an increasingly important part of decision making processes. For this reason, it is essential that computational models be high quality, reproducible, and credible. Although other scientific fields are building qualitative guidelines for model credibility, no such standards exist for systems biology models.

The purpose of this research is to create a user-friendly tool to automatically test systems biology models and assess their credibility. Users could access this software either online or locally via GUI or command line. The user would input their model in a standardized format, such as SBML [39], and the software would assess the model's structure and behavior, verify that the model's output is consistent across platforms, and validate the model against validation data provided by the user. The software would produce and overall score for model credibility as well as a report detailing areas where model credibility could be improved. This tool would be targeted at (1) modelers who may use this tool during the development process to iteratively improve their model (2) by journals or reviewers who may wish to verify a model is sufficiently credible for publication and (3) third parties who want to assess a previously published model's credibility prior to using it in their own studies or decisions.

5.2 Types of Software Testing

The proposed credibility assessment tool will borrow from software testing best practices common in computer science. In particular, this software will make use of static, dynamic, and unit testing. The proposed research will focus on dynamic testing. Other members of the Sauro lab are currently developing static tests.

Static tests are tests that can be performed without running any code. In this context, static tests might include testing for errors in naming, ratelaws, or kinetics, nonsensical processes, uninitialized variables/parameters, relationships between species, and more. Model annotation may also be included in static testing. These test are beyond the scope of the proposed research, but will be included in the final software.

Dynamic tests, the primary focus of this research proposal, assess a model's behavior and require that a model be simulated. These tests might examine the dynamics of model time courses, test for runtime exceptions, and ensure the species concentrations are within certain bounds.

Unit testing refers to the practice of testing small pieces of isolated code to ensure the correct output. Translating this concept to systems biology is less obvious as it is unclear what could be considered an isolatable unit. Perhaps a single reaction could be a unit. Tests might involve checking for errors in the rate law (i.e. that the rate law is dependent on at least one of the substrates), checking for mass balance errors, verifying correct annotate, and ensuring that there are no nonsensical reactions.

5.3 Sensitivity Analysis

This proposal will also address parameter sensitivity as a metric for credibility. Parameter sensitivity refers to the extent to which a change in a parameter value influences change in the model output. A small change in a highly sensitive parameter value would lead to a large change in model output.

Due to genetic variability, biological system parameters are slightly varied in nature. Nonetheless, biological systems remain functional across organisms despite these small variations. In general, a the presence of a highly sensitive parameter would indicate that a model is less credible.

5.4 Research Aims

This proposal is broken down into three primary aims discussed in detail in the following chapters.

Aim 1: Develop a python package to perform dynamic tests on systems biology models in order to assess their credibility.

Initial dynamic tests will include testing that species concentrations do not become negative or increase to infinity, testing that specific changes in model variables affect the model output as intended, and checking if species concentrations are within pre-specified bounds.

Aim 2: Create tools to automatically verify and validate models.

Verification ensures that models perform consistently when simulated on different platforms. BioSimulations, a recently developed tool to encapsulate various simulators and operating systems, will be used to perform verification tests. Validation ensures that a model captures the experimental data it was intended to represent. Developing validation protocols is a greater challenge and will likely require that the user submit validation data alongside their model.

Aim 3: Build software to automate parameter sensitivity analysis.

High parameter sensitivity increases uncertainty in a model's output and reduces model credibility.

Chapter 6

Dynamic Testing

The field of software engineering has relied on static and dynamic testing for decades to ensure proper functionality [97]. In broad terms, static tests can assess code without executing the code whereas dynamic test require code execution[1]. Translated to computational systems biology, static tests assess a models structure and dynamic tests assess a model's behavior when simulated. Both are essential tools to ensure the functionality and credibility of a model, but **the scope of this proposal covers only dynamic testing.** Static tests are being addressed by other members of the Sauro Lab. In the long term, the proposed credibility test suite may be combined with static test suites currently in development.

6.1 Model dynamics

Unlike static tests, dynamic tests require simulating the model in order to check for things like enzyme behavior. Some essential dynamic checks are for species concentrations. Species concentrations shouldn't be less than zero as negative concentration is meaningless. Similarly, concentrations should not approach infinity over time as this is biologically impossible and clearly indicates either an error in the model or lack of model credibility. These checks might seem trivial, but they are simple ways of testing model credibility. One might think it is easy to simulate a model and see if there are negative

[1] It should be noted that the phrase "dynamic testing" as used in this proposal refers to tests that require model simulation. It *does not* refer specifically to tests of a model's dynamics, although confusingly, model dynamics are one subject tested with dynamic tests.

concentrations or concentrations that approach infinity, but with an automated approach, a reviewer could easily assess the credibility of a model, a journal editor could run several models through the software to quickly check for these issues.

6.2 Negative Concentrations and Concentrations Approaching Infinity

In terms of implementation, checking for negative concentrations is relatively simple. The model would be simulated over a relevant timecourse, perhaps determined by units included in the model or drawn from figures, and then the timecourse data would be scanned to check for negative values. Very small negative values that are possibly results of machine precision errors would be flagged as a warning, while significant negative values would fail the test and alert the user. The presence of negative concentration values could indicate a simple typographical error in stoichiomety or rate constants, or could be indicative or a larger problem with the model. In either case, they signal reduced model credibility.

Concentrations that approach infinity are also indicative of poor model credibility. Like negative concentrations, concentrations approaching infinity could be due simple typos, or signal a larger problem with the model. Unlike negative concentration testing, tests for infinitely rising concentrations are less simple. As depicted in figure 6.1, a model simulated for a short time period might show a species rising in concentration throughout the simulation, but when simulated for longer, the concentration may level out to a steady state or oscillate. Additionally, checking that a concentration never decreases during simulation is insufficient, as a concentration can oscillate while approaching infinity. Similarly, checking for "unreasonably large" concentrations might yield false positives. For example, a concentration of 10^30 g/mL would clearly be unreasonable, but if the concentration is given in molecules it becomes believable. Most likely a test to determine if concentrations approach infinity would need to be multifaceted.

The most straight forward testing technique would be to simulate the model for increasing amounts of time while testing for RuntimeErrors in the solver. When a model can be successfully simulated in smaller timescales, but causes the solver to fail at longer time periods, it is often indicative of a concentration that approaches infinity. Solver errors typically arises when values are so large that the solver can't handle them. Unfortunately, "small" timescales where the model simulates and "large" timescales where it doesn't due to infinite concentrations are poorly defined and different across models. One model may rapidly approach infinity and fail when simulated for longer than 10 seconds while another might only fail after 500.

Neither checking for simulation errors nor checking for species that fail to decrease or reach steady state are sufficient to test for concentrations that approach infinity. It may be possible to check for this mathematically, but implementation would be a challenge. While insufficient, perhaps the combination of multiple tests would be able to flag suspect concentrations for manual inspection.

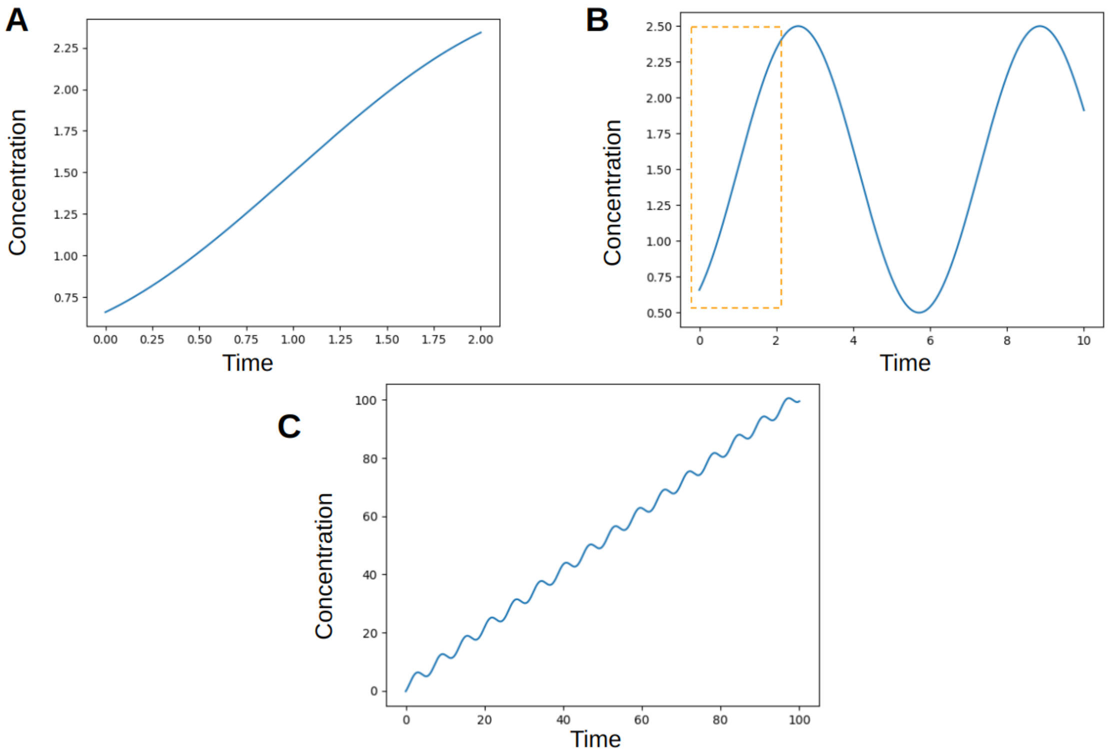


Table 6.1: Challenges in testing for concentrations that approach infinity with time series data. (A) Appears to approach infinity, but is in fact an oscillator, as depicted in (B). The orange box shows the time period plotted in (A). (C) A species may decrease in concentration at some points, but still approach infinite concentration.

6.3 Specific Changes in Model Variables

Due to range variety of systems biology models, some credibility tests may be specific to the model in question. These tests would be customized by the model developer and would relate to the behavior expected in the model of interest. One such test might verify that key species in the model behave as expected when other model variables are changed. For example, if the concentration of ATP were to increase, the concentration of ADP might be expected to decrease. If such a decrease were not present after an increase in ATP, it might indicate an error in the model or a peculiarity in the model's function, either of which would imply that the model is less credible.

These tests are not limited to species concentrations. Changes in enzyme concentration or substrate might be expected to increase reaction rates or flux. For example, and increase in enzyme concentration should produce an increase in the associated reaction's rate. Increasing glucose concentration should increase glycolytic rates.

These tests would be defined by the user, but the proposed software would automatically create these tests upon definition of the parameters in question and the expected result. The result of these customized tests and the tests themselves would be recorded in a credibility report file and a test file respectively to be included with the model upon dissemination.

Although these test have yet to be developed, some testing techniques are already under consideration. In the case of expected concentration changes with respect to another variable (e.g. increases in ATP resulting in decreases of ADP), it may be useful to compute the derivatives of existing time series data. The user could specify the species in question (e.g. ATP and ADP) and indicate their expected correlation (negative in this case). The test software would then numerically compute the derivatives across the time series data and report the percent of data that demonstrated the expected correlation.

These tests may also make use of the "events" feature in the antimony model description language [13]. An instantaneous change in species concentration or rate can be programmed into the model to take place at a specific time in the simulation. For example, an event could be added that would cause glucose to instantaneously double in value. The test could check if the rate of conversion from glucose to glucose 6-phosphate increases as well as other downstream reactions at that moment.

6.4 Bounds Checking

Checking that certain species concentrations stay within a specified bounds could also be a model specific credibility test. The user would input an upper and lower bounds for a specific species concentration. The simulation would be run and the time series data examined to ensure that the species concentration did not deviate from the specified range.

These tests would likely be based on known biological properties. For example, normal blood glucose concentrations typically fall between 3.5 and 5.5 mM [98]. "Normal" blood glucose concentrations vary among individuals as well as the labs measuring these concentrations. To be generous, a user might specify that the glucose concentration in a model of blood glucose dynamics be no less than 2 mM and no greater than 8 mM. A model producing a glucose concentration beyond this range might have an error in structure or parameter value that renders it less credible. This is of course a very simple example and one might expect a modeler building a blood glucose model would surely examine the output of the primary species of interest, but even in this simple case, such a test could serve as a tool to ensure that the model was disseminated without error or that the numerical algorithms are functioning as they should. An addition parameter sensitivity test (described in chapter 7), might point a user to the parameters that may be responsible for any errors in concentration bounds.

6.5 Infrastructure

Implementing these test will require some additional standardization in file structure. As discussed in chapter 3, COMBINE archives are standardized sip files that contain all necessary files and data required to completely reproduce a systems biology model [76]. A credibility testing software could make use of these standardized formats to access models and any associated files. The test system could output a credibility report to be stored in the COMBINE archives for ease of access upon dissemination.

The proposed credibility test suite could also make use of SED-ML (Simulation Experiment Description Markup Language) [70, 71, 77], a language developed to describe model simulations. Model developers may already have already created some tests for their models, which could be included in the COMBINE archive as SED-ML files. These tests could be incorporated into the proposed test suite as an additional measure of credibility. A model that can not pass customized tests that were used in development could be considered less credible.

The test suite could also produce a SED-ML file of the simulations performed to test the model. This would provide transparency as to how tests were performed for that specific model and enable a user to independently reproduce the tests or modify them for their own purposes.

Chapter 7

Verification and Validation

7.1 Validation

In the context of computational modeling, validation is defined as "the process of determining the degree to which a simulation model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model" [99]. Validation is a component of all existing credibility guidelines reviewed in chapter 3 and can be considered a principal component of any credibility testing suite. Unfortunately validation data or evidence is often omitted from published models, including those curated in databases. In cases where validation was performed, it is often only mentioned in passing in the model's associated research article. Given its importance model credibility, validation data and evidence should be readily available to third parties in a format that easily allows for confirmation and reproduction. Examining and potentially reproducing model validation may be of interest to users, reviewers, and publishers alike.

The proposed credibility test suite could support validation testing by allowing users to include validation data sets in the COMBINE Archive. In the shot term, The format of this data might be standardized to allow the test suite to simulate the model and assess the extent to which the model conforms to the validation set. The user could potentially customize an acceptable level of"non-conformity" or the software could simply report a metric describing the the model's ability to match the validation data. The user could optionally include any additional validation tests they performed to be factored into the credibility assessment. Validation tests might be included as python scripts and expected results, python unit tests, or SED-ML files with expected outputs. The results of these validation tests would be factored in to a credibility score or assessment and included in a credibility report.

However, it is often the case that systems biology modelers do not wish to write validation tests. The longer term vision for this feature would be a software system with a user friendly interface that would allow modelers to decide how a given model is validated and automatically produce tests to validate the model as specified. Options might include what variable are validated, at what values, and with what data. Validation data might come in the form of time course data, steady-state data, or perturbation data. In addition to validation results, the custom validation tests would be outputted as SED-ML files and python scripts so that users and third parties could easily see exactly what validation tests were performed and reproduce them if desired. For model developers, these test may guide their development process or serve as templates for additional customized validation tests.

7.2 Verification

Model verification is also a recurring theme in model credibility across disciplines. Verification is defined as "the process of determining that a model implementation and its associated data accurately represent the developer's conceptual description and specifications" [99]. In the context of computational modeling, a significant portion of verification consists of static tests, such as checking for typos in rate laws and species name, syntax errors in code, that all parameters are initialized. As static tests are beyond the scope of this proposal, the credibility test suite will perform dynamic verification tests. In the future the proposed dynamic verification tests may be combined with static verification tests in a single model testing software package.

In practice, dynamic verification tests entail testing if the underlying numerical algorithms have been implemented correctly. One way to test this is to check if the model produces the same results when identical simulations are run on different platforms (such as Tellurium [100] and COPASI [101]) and operating systems (such as MacOS, Windows, and various Linux distributions). The proposed verification tests will make use of BioSimulations, a user-friendly web applications provides access to a number of different simulation tools and containerized interfaces to each version of each tool [102]. The credibility assessment software will automatically link to the BioSimulations wed page to simulate the model using several different tools and operating systems. A later iteration of this project could allow users to customize which versions, tools, operating systems, they wish to use for verification. The model would be considered more credible if its outputs are consistent across a variety of platforms. BioSimulations can also automatically generate reports which could be directly included in the proposed software's credibility assessment report or simplified for brevity.

Chapter 8

Sensitivity and Uncertainty Analysis

8.1 Introduction

A major aspect in model credibility is uncertainty, which appears in many of the credibility guidelines described in chapter 3. The more uncertain a result is, the less credible it is. Uncertainty analysis is performed to investigate the uncertainty in a model's output resulting from uncertainty in parameter inputs [103]. Model outputs that depend strongly on parameters with high uncertainty yield less credible predictions. Sensitivity analysis assesses how variations in a model's outputs can be apportioned to different input sources [104]. This is illustrated in figure 8.1. Input uncertainty (in this case parameter uncertainty) lies on the X-axis as an inverted probably distribution function. The Y-axis depicts model output as a probability distribution functions. The less uncertain model (blue) has a lower standard deviation compared to the more uncertain result in green. The green line illustrates the relationship between an input parameter with high sensitivity, where a scan across the parameters probability distribution function yields a wider range of model outputs compared to a scan across a less sensitive parameter.

Clearly, highly sensitive parameters that are not known with certainty (as is usually the case) lead to more uncertainty in model results and therefore reduced credibility. There are two reasons that parameter sensitivity should be considered from a credibility perspective. First, in nature, it is unlikely that there is a single "true" value of a parameter. It is more likely that parameters vary

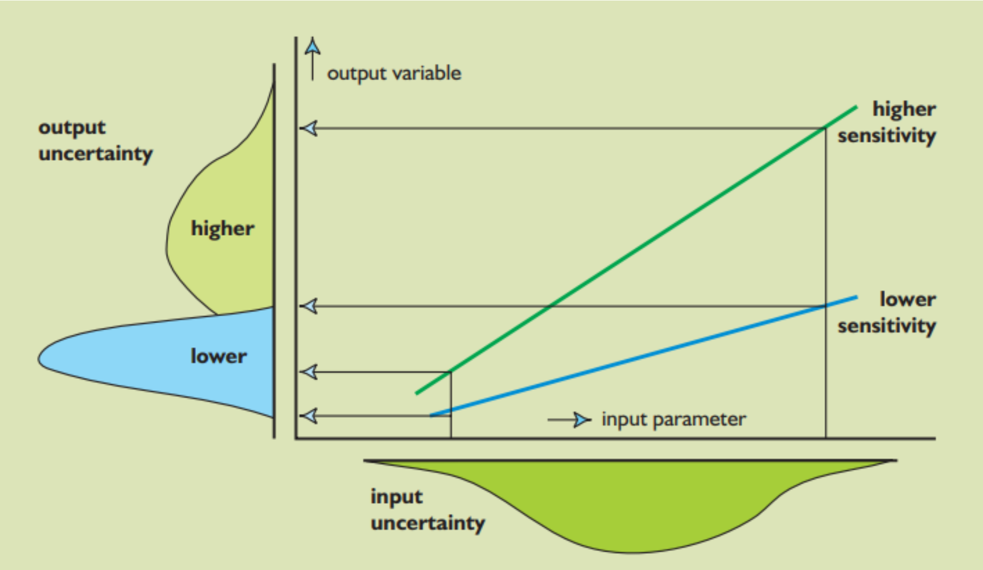


Figure 8.1: The relationship between parameter sensitivity and output uncertainty. In highly sensitive parameters, parameter uncertainty leads to larger output uncertainty and reduced credibility. (Image from Loucks *et al.* [105], originally published by W. Lal, 1995 [106].

over ranges in different cell types, cellular environments, and genetic variants [107]. Second, many biological model display "sloppy" sensitivity spectra, meaning that most parameters can be varied with minimal effect on the result. In other words, a model output can be predictive and relatively certain despite high uncertainty in parameters [108].

Although model output uncertainty is more directly indicative of credibility compared to parameter sensitivity, uncertainty analysis is significantly more complicated in terms of implementation, variety of techniques, and computational expense. So while high parameter sensitivity does not necessarily indicate low credibility, it may be useful to highlight highly sensitive parameters so that (1) they can be corrected in the case of a simple mistakes, (2) further and clarified explored experimentally, and (3) model output uncertainty can be apportioned to specific parameters. The third aim of this proposal is to build software to automate simple parameter sensitivity analyses in order to highlight highly sensitive parameters that most contribute to output uncertainty.

8.2 Sources of Parameter Uncertainty

Computational modeling has proved useful across a variety of scientific and engineering fields, but modeling applications in biology face unique challenges with parameterization. For example, when modeling systems in classical physics, parameters are often based on physical laws with well established physical constraints. Similarly, in engineering, individual model components are typically well characterized and their interactions known. Computational systems biology faces additional challenges in parameterization due to the reliance on *in vitro* measurements. These measurements can vary due to differences in the biochemical environment and regulatory mechanisms that were not accounted for [109]. Additionally, these measurements can be expensive and time consuming, resulting in a paucity of data. Combined with the complexity of biological models, this lack of data leads to high uncertainty in inferred parameter values and consequently, poorly constrained and uncertain model predictions. Although many techniques exist for analyzing parameter sensitivity [109, 110, 111, 112], the proposed research will initially explore relatively simple and computationally inexpensive techniques in an effort to make a broadly applicable and user-friendly tool to identify potentially problematic parameter sensitivities.

8.3 One-at-a-Time (OAT) Sensitivity Analysis

There are many techniques to assess the sensitivity of parameters in computational models. A complete exposition of sensitivity analysis and techniques is beyond the scope of this proposal, so discussion will be limited to the relatively simple one-factor-at-a-time analysis. Changing one-factor-at-a-time (OAT) to see how it affects the output is one of the simplest and most common approaches to sensitivity analysis. OAT involves changing an input variable slightly while keeping all others unchanged, measuring the model's output, returning the variable to its baseline value and then repeating for all other input variables. Sensitivty can then be measured by examining changes in the output, usually by partial derivatives, $\partial Y_j / \partial X_i$, where the change in output Y_i is compared to a change of input X_i. This is referred to the sensitivity of Y_j versus X_i [104].

This is the technique that the simulation software RoadRunner uses to quickly asses sensitivity in systems biology models [14]. RoadRunner defines two types of sensitivity, systems and global, based on metabolic control analysis (MCA) [113]. Although MCA was originally developed to study metabolic networks, it can also be applied to other biological networks such as cell signaling, genetic networks, and others [110]. Local sensitivity is defined as $\frac{\partial v}{\partial S}\frac{S}{v}$ where v is a reaction rate and S is an effector of the reaction (typically a species that influences the reaction as a substrate, inhibitor, catalyst etc). This metric describes how a change in a reaction's effector species concentration effects the reaction's rate. This metric is scaled, representing approximately $\frac{\%S}{\%v}$

System sensitivities are described by control coefficients, each of which has two forms, flux and concentration [113]. The flux control coefficient measures how sensitive a reaction's flux is to changes in the reaction rate, usually performed by changing and enzyme concentration. This is defined as where E is an enzyme concentration and J is a reaction's flux. Similarly, the concentration control coefficient measures how a change in reaction rate (i.e. enzyme concentration) effects a given species concentration, $\frac{\partial S}{\partial E}\frac{E}{S} \approx \frac{\%S}{\%E}$.

As a simple preliminary step, the proposed software will make use of these built in functions to measure local and systems sensitivity, where appropriate. Due to the variety of models, it is likely that not all high sensitivities indicate decreased credibility. For example, a signaling cascade might be particularly sensitive to a species concentration. Similarly, the amount of "acceptable" sensitivity may vary from model to model. For these reasons, the preliminary sensitivity assessment will likely be excluded from any quantitative credibility scoring system, but will be included in a credibility report to alert the user to particularly high sensitivities so they can be verified, corrected, or further assessed.

8.4 Weaknesses of OAT Sensitivity Analysis

A possible weakness of this approach, as the name implies, is that sensitivity is measured by perturbing a single variable. It is possible that the combination of two or more parameters perturbed slightly at the same time would dramatically affect the model's output, however testing these combinations is not feasible for the software proposed here, designed to do a coarse grain assessment of a model's general credibility. If this avenue were to be pursued in more detail, it may be possible to implement these features or create a separate software package designed for sensitivity testing. If this were the case, the user might choose the experimental design and ranges over which parameters are tested.

Additionally, the utility of sensitivity analyses may depend on the identifiability of model parameters. If models are calibrated and well-constrained with large experimental data sets, sensitivity analysis can be highly useful in pinpointing the most influential parameters which should be investigated further. If models are "sloppy," as is often the case, it is difficult to determine the parameters by fitting to experimental data. In these cases, sensitivity analysis may be a useful tool to gauge model output uncertainty, but the information may not be sufficient to identify parameters that should be studied further [110]. In any case, the purpose of including sensitivity analysis in this tool is to serve as a preliminary assessment of result uncertainty. Extensive sensitivity testing and result uncertainty quantification is beyond the scope of this research, although a python package implementing more complex sensitivity and uncertainty analyses for systems biology models may prove useful to the modeling community.

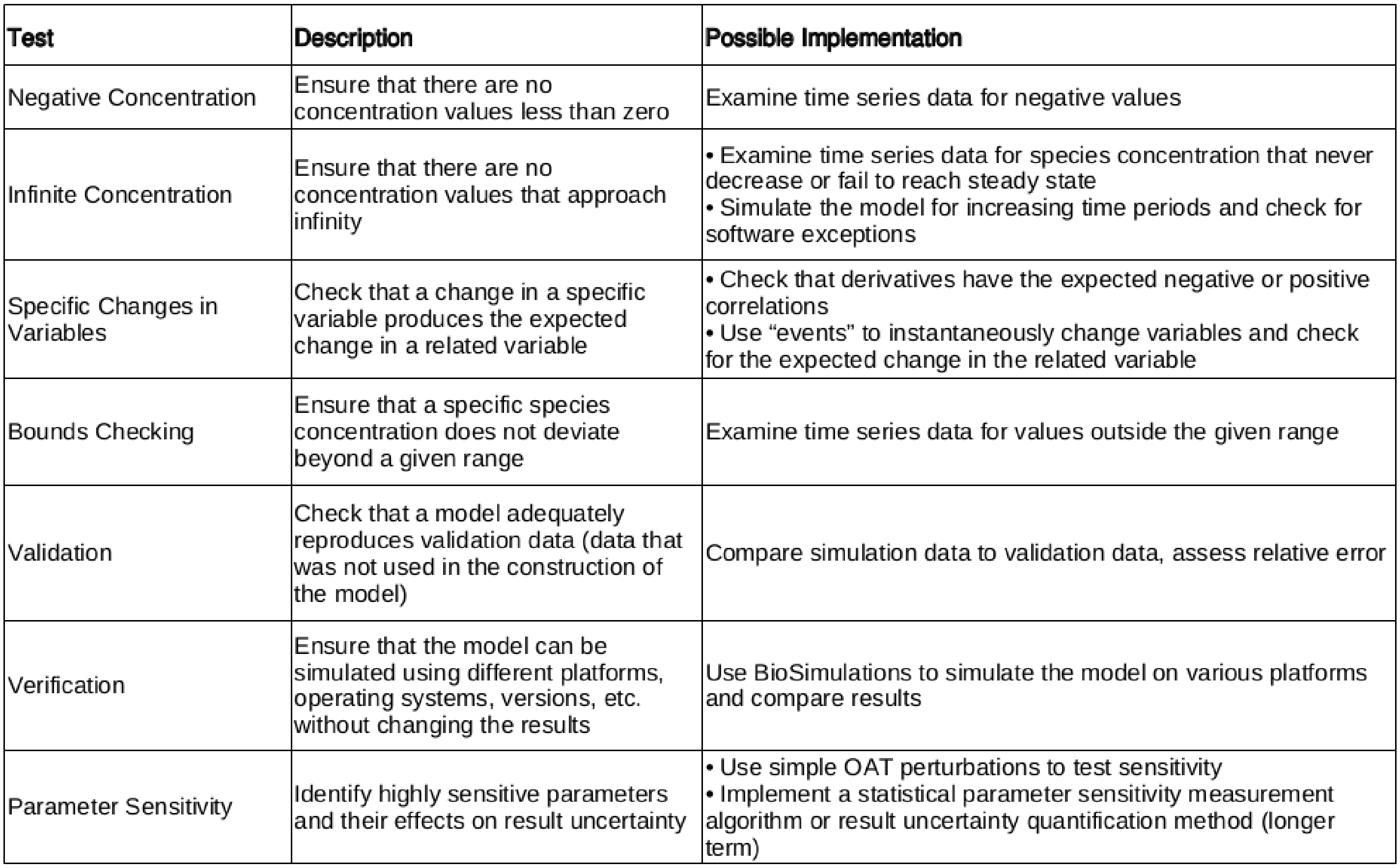
Chapter 9

Summary

As computational models become increasingly important in decision-making, so too does model credibility. Although many efforts have been made in systems biology to create standards that ensure model reproducibility, to date, no standards exist to promote credibility. In other industries, such as aerospace, medical devices, and pharmaceuticals, preliminary credibility guidelines have been developed to steer model creators towards making credible models. Owing to the range of applications and models in these fields, these credibility guidelines are largely qualitative and highly subjective. Fortunately, computational systems biology models are of relatively small scope and already rely on several standards describing model description, annotation, simulation, and dissemination. The purpose of the proposed research is to standardize systems biology model credibility with a software tool that automatically assess and scores the credibility of models.

This research is broken down into three aims: (1) to create dynamic tests to assess systems biology model credibility, (2) to create tools to verify and validate models, and (3) to automate simple parameter sensitivity tests as a proxy for result uncertainty. The proposed tests and possible implementations are summarized in table 9.1. It is likely that as the research progresses, more tests and implementations will be added to this list.

The proposed research is expected to result in two first author publications, to be submitted by June 2024. The first will highlight the need and uses of the software, introduce the tests to be performed, and provide a simple demonstration of an early implementation of the software on



Test	Description	Possible Implementation
Negative Concentration	Ensure that there are no concentration values less than zero	Examine time series data for negative values
Infinite Concentration	Ensure that there are no concentration values that approach infinity	• Examine time series data for species concentration that never decrease or fail to reach steady state • Simulate the model for increasing time periods and check for software exceptions
Specific Changes in Variables	Check that a change in a specific variable produces the expected change in a related variable	• Check that derivatives have the expected negative or positive correlations • Use “events” to instantaneously change variables and check for the expected change in the related variable
Bounds Checking	Ensure that a specific species concentration does not deviate beyond a given range	Examine time series data for values outside the given range
Validation	Check that a model adequately reproduces validation data (data that was not used in the construction of the model)	Compare simulation data to validation data, assess relative error
Verification	Ensure that the model can be simulated using different platforms, operating systems, versions, etc. without changing the results	Use BioSimulations to simulate the model on various platforms and compare results
Parameter Sensitivity	Identify highly sensitive parameters and their effects on result uncertainty	• Use simple OAT perturbations to test sensitivity • Implement a statistical parameter sensitivity measurement algorithm or result uncertainty quantification method (longer term)

Table 9.1: Summary of proposed tests and possible implementations.

simple models. The second publication will introduce the fully developed software and demonstrate its functionality with several published models.

Bibliography

[1] Janyce Eunice Gnanvi, Kolawolé Valère Salako, Gaëtan Brezesky Kotanmi, and Romain Glèlè Kakaï. On the reliability of predictions on covid-19 dynamics: A systematic and critical review of modelling techniques. *Infectious Disease Modelling*, 6:258–272, 2021.

[2] Anastasia C Deckard, Frank T Bergmann, and Herbert M Sauro. Enumeration and online library of mass-action reaction networks. *arXiv preprint arXiv:0901.3067*, 2009.

[3] Casian Pantea and Gheorghe Craciun. Computational methods for analyzing bistability in biochemical reaction networks. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 549–552, 2010.

[4] Marco S. Nobile, Daniela Besozzi, Paolo Cazzaniga, Dario Pescini, and Giancarlo Mauri. Reverse engineering of kinetic reaction networks by means of cartesian genetic programming and particle swarm optimization. pages 1594–1601, 2013.

[5] Pete Donnell, Murad Banaji, Anca Marginean, and Casian Pantea. CoNtRol: an open source framework for the analysis of chemical reaction networks. *Bioinformatics*, 30(11):1633–1634, 01 2014.

[6] Béla Novák and John J. Tyson. Design principles of biochemical oscillators. *Nature Reviews Molecular Cell Biology*, 9(12):981–991, oct 2008.

[7] Alexander Aulehla and Olivier Pourquié. Oscillating signaling pathways during embryonic development. *Current opinion in cell biology*, 20(6):632–637, dec 2008.

[8] Naama Geva-Zatorsky, Nitzan Rosenfeld, Shalev Itzkovitz, Ron Milo, Alex Sigal, Erez Dekel, Talia Yarnitzky, Yuvalal Liron, Paz Polak, Galit Lahav, and Uri Alon. Oscillations and variability in the p53 system. *Molecular systems biology*, 2(1):2006–0033, jan 2006.

[9] Balth. van der Pol and J. van der Mark. LXXII.the heartbeat considered as a relaxation oscillation, and an electrical model of the heart. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(38):763–775, nov 1928.

[10] Alfred J. Lotka. Contribution to the theory of periodic reactions. *The Journal of Physical Chemistry*, 14(3):271–274, March 1910.

[11] Joseph Higgins. The theory of oscillating reactions - kinetics symposium. *Industrial &amp; Engineering Chemistry*, 59(5):18–62, May 1967.

[12] SR Paladugu, V Chickarmane, A Deckard, JP Frumkin, M McCormack, and HM Sauro. In silico evolution of functional modules in biochemical networks. *IEE Proceedings-Systems Biology*, 2006.

[13] Lucian Smith, Frank Bergmann, Deepak Chandran, and Herbert Sauro. Antimony: A modular model definition language. *Bioinformatics (Oxford, England)*, 25:2452–4, 08 2009.

[14] Endre T Somogyi, Jean-Marie Bouteiller, James A Glazier, Matthias König, J Kyle Medley, Maciej H Swat, and Herbert M Sauro. libroadrunner: a high performance sbml simulation and analysis library. *Bioinformatics*, 31(20):3315–3321, jun 2015.

[15] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[16] Brad L. Miller and David E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst.*, 9, 1995.

[17] Herbert Sauro. Network dynamics. *Methods in Molecular Biology*, 2009.

[18] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126, October 2020.

[19] John J. Tyson. Classification of instabilities in chemical reaction systems. *The Journal of Chemical Physics*, 62(3):1010–1015, February 1975.

[20] J.J Tyson. Biochemical oscillators. In *Computational Cell Biology*. Springer New York, 2007.

[21] John J. Tyson. Biochemical oscillations. In *Interdisciplinary Applied Mathematics*, pages 230–260. Springer New York.

[22] Herbert Sauro. teUtils [software]. *retrieved from https://github.com/sys-bio/teUtils*, 2020.

[23] Chengyuan Wang, Si Li, Adesola S. Ademiloye, and Perumal Nithiarasu. Biomechanics of cells and subcellular components: A comprehensive review of computational models and applications. *International Journal for Numerical Methods in Biomedical Engineering*, 37(12), August 2021.

[24] James M. Hassell, Tim Newbold, Andrew P. Dobson, Yvonne-Marie Linton, Lydia H. V. Franklinos, Dawn Zimmerman, and Katrina M. Pagenkopp Lohan. Towards an ecosystem model of infectious disease. *Nature Ecology &amp; Evolution*, 5(7):907–918, May 2021.

[25] Ezio Bartocci and Pietro Lió. Computational modeling, formal analysis, and tools for systems biology. *PLOS Computational Biology*, 12(1):e1004591, January 2016.

[26] Assessing the credibility of computational modeling and simulation in medical device submissions: Draft guidance for industry and food and drug administration staff. *U.S. Food and Drug Administration*, 2021.

[27] Janis Shin, Veronica Porubsky, James Carothers, and Herbert M Sauro. Standards, Dissemination, and Best Practices in Systems Biology.

[28] Rahuman S Malik-Sheriff, Mihai Glont, Tung V N Nguyen, Krishna Tiwari, Matthew G Roberts, Ashley Xavier, Manh T Vu, Jinghao Men, Matthieu Maire, Sarubini Kananathan, Emma L Fairbanks, Johannes P Meyer, Chinmay Arankalle, Thawfeek M Varusai, Vincent Knight-Schrijver,

Lu Li, Corina Dueñas-Roca, Gaurhari Dass, Sarah M Keating, Young M Park, Nicola Buso, Nicolas Rodriguez, Michael Hucka, and Henning Hermjakob. BioModels — 15 years of sharing computational models in life science. *Nucleic Acids Research*, 48(D1):D407–D415, 1 2020. gkz1055.

[29] Krishna Tiwari, Sarubini Kananathan, Matthew G. Roberts, Johannes P. Meyer, Mohammad Umer Sharif Shohan, Ashley Xavier, Matthieu Maire, Ahmad Zyoud, Jinghao Men, Szeyi Ng, Tung V. N. Nguyen, Mihai Glont, Henning Hermjakob, and Rahuman S. Malik-Sheriff. Reproducibility in systems biology modelling. *Molecular Systems Biology*, 17(2):e9982, February 2021. Publisher: John Wiley & Sons, Ltd.

[30] Maria Babula, William Bertch, Lawrence Green, Joseph Hale, Gary Mosier, Martin Steele, and Jody Woods. NASA Standard for Models and Simulations: Credibility Assessment Scale. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, January 2009. American Institute of Aeronautics and Astronautics.

[31] T Shepard, G Scott, S Cole, A Nordmark, and F Bouzom. Physiologically based models in regulatory submissions: Output from the ABPI/MHRA forum on physiologically based modeling and simulation. *CPT Pharmacometrics Syst. Pharmacol.*, 4(4):221–225, April 2015.

[32] Edda Klipp, Wolfram Liebermeister, Anselm Helbig, Axel Kowald, and Jörg Schaber. Standards in computational systems biology. 2007.

[33] N. Novère, M. Hucka, and H Mi. The systems biology graphical notation. *Nature Biotechnology*, 27, 2009.

[34] Dada, Spasić, Paton, and Mendes. Sbrml: a markup language for associating systems biology data with models. *Bioinformatics*, 26, 2010.

[35] Daniel Machado, Rafael S Costa, Miguel Rocha, Eugénio C Ferreira, Bruce Tidor, and Isabel Rocha. Modeling formalisms in Systems Biology. *AMB Express*, 1:45, December 2011.

[36] Autumn A. Cuellar, Catherine M. Lloyd, Poul F. Nielsen, David P. Bullivant, David P. Nickerson, and Peter J. Hunter. An overview of CellML 1.1, a biological model description language. *SIMULATION*, 79(12):740–747, December 2003.

[37] Emek Demir, Michael P. Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'Eustachio, Carl Schaefer, Joanne Luciano, Frank Schacherer, Irma Martinez-Flores, Zhenjun Hu, Veronica Jimenez-Jacinto, Geeta Joshi-Tope, Kumaran Kandasamy, Alejandra C. Lopez-Fuentes, Huaiyu Mi, Elgar Pichler, Igor Rodchenkov, Andrea Splendiani, Sasha Tkachev, Jeremy Zucker, Gopal Gopinath, Harsha Rajasimha, Ranjani Ramakrishnan, Imran Shah, Mustafa Syed, Nadia Anwar, Ozgun Babur, Michael Blinov, Erik Brauner, Dan Corwin, Sylva Donaldson, Frank Gibbons, Robert Goldberg, Peter Hornbeck, Augustin Luna, Peter Murray-Rust, Eric Neumann, Oliver Reubenacker, Matthias Samwald, Martijn van Iersel, Sarala Wimalaratne, Keith Allen, Burk Braun, Michelle Whirl-Carrillo, Kam Dahlquist, Andrew Finney, Marc Gillespie, Elizabeth Glass, Li Gong, Robin Haw, Michael Honig, Olivier Hubaut, David Kane, Shiva Krupa, Martina Kutmon, Julie Leonard, Debbie Marks, David Merberg, Victoria Petri, Alex Pico, Dean Ravenscroft, Liya Ren, Nigam Shah, Margot Sunshine, Rebecca Tang, Ryan Whaley, Stan Letovksy, Kenneth H. Buetow, Andrey Rzhetsky, Vincent Schachter, Bruno S. Sobral, Ugur Dogrusoz, Shannon McWeeney, Mirit Aladjem, Ewan Birney, Julio Collado-Vides, Susumu Goto, Michael Hucka, Nicolas Le Novère, Natalia Maltsev, Akhilesh Pandey, Paul Thomas, Edgar Wingender, Peter D. Karp, Chris Sander, and Gary D. Bader. BioPAX – A community standard for pathway data sharing. *Nat Biotechnol*, 28(9):935–942, September 2010.

[38] Padraig Gleeson, Sharon Crook, Robert C. Cannon, Michael L. Hines, Guy O. Billings, Matteo Farinella, Thomas M. Morse, Andrew P. Davison, Subhasis Ray, Upinder S. Bhalla, Simon R. Barnes, Yoana D. Dimitrova, and R. Angus Silver. NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Computational Biology*, 6(6):e1000815, June 2010.

[39] Hucka, Bolouri H., Finney A., Sauro H.M., Doyle J.C., Kitano H., Arkin A.P., and Born-

stein B.J. et al. The systems biology markup language (sbml): A medium for representation and exchange of biochemical network models. *bioinformatics*, 19, 2003.

[40] A. Finney and M. Hucka. Systems biology markup language: Level 2 and beyond. *Biochemical Society Transactions*, 31(6):1472–1473, December 2003.

[41] Michael Kohl. Standards, Databases, and Modeling Tools in Systems Biology. In Michael Hamacher, Martin Eisenacher, and Christian Stephan, editors, *Data Mining in Proteomics*, volume 696, pages 413–427. Humana Press, Totowa, NJ, 2011. Series Title: Methods in Molecular Biology.

[42] Sarah M Keating, Dagmar Waltemath, Matthias König, Fengkai Zhang, Andreas Dräger, Claudine Chaouiya, Frank T Bergmann, Andrew Finney, Colin S Gillespie, Tomáš Helikar, Stefan Hoops, Rahuman S Malik-Sheriff, Stuart L Moodie, Ion I Moraru, Chris J Myers, Aurélien Naldi, Brett G Olivier, Sven Sahle, James C Schaff, Lucian P Smith, Maciej J Swat, Denis Thieffry, Leandro Watanabe, Darren J Wilkinson, Michael L Blinov, Kimberly Begley, James R Faeder, Harold F Gómez, Thomas M Hamm, Yuichiro Inagaki, Wolfram Liebermeister, Allyson L Lister, Daniel Lucio, Eric Mjolsness, Carole J Proctor, Karthik Raman, Nicolas Rodriguez, Clifford A Shaffer, Bruce E Shapiro, Joerg Stelling, Neil Swainston, Naoki Tanimura, John Wagner, Martin Meier-Schellersheim, Herbert M Sauro, Bernhard Palsson, Hamid Bolouri, Hiroaki Kitano, Akira Funahashi, Henning Hermjakob, John C Doyle, Michael Hucka, Richard R Adams, Nicholas A Allen, Bastian R Angermann, Marco Antoniotti, Gary D Bader, Jan Červený, Mélanie Courtot, Chris D Cox, Piero Dalle Pezze, Emek Demir, William S Denney, Harish Dharuri, Julien Dorier, Dirk Drasdo, Ali Ebrahim, Johannes Eichner, Johan Elf, Lukas Endler, Chris T Evelo, Christoph Flamm, Ronan MT Fleming, Martina Fröhlich, Mihai Glont, Emanuel Gonçalves, Martin Golebiewski, Hovakim Grabski, Alex Gutteridge, Damon Hachmeister, Leonard A Harris, Benjamin D Heavner, Ron Henkel, William S Hlavacek, Bin Hu, Daniel R Hyduke, Hidde Jong, Nick Juty, Peter D Karp, Jonathan R Karr, Douglas B Kell, Roland Keller, Ilya Kiselev, Steffen Klamt, Edda Klipp, Christian Knüpfer, Fedor Kolpakov, Falko Krause, Martina Kutmon, Camille Laibe, Conor Lawless, Lu Li, Leslie M Loew, Rainer Machne, Yukiko Matsuoka,

Pedro Mendes, Huaiyu Mi, Florian Mittag, Pedro T Monteiro, Kedar Nath Natarajan, Poul MF Nielsen, Tramy Nguyen, Alida Palmisano, Jean-Baptiste Pettit, Thomas Pfau, Robert D Phair, Tomas Radivoyevitch, Johann M Rohwer, Oliver A Ruebenacker, Julio Saez-Rodriguez, Martin Scharm, Henning Schmidt, Falk Schreiber, Michael Schubert, Roman Schulte, Stuart C Sealfon, Kieran Smallbone, Sylvain Soliman, Melanie I Stefan, Devin P Sullivan, Koichi Takahashi, Bas Teusink, David Tolnay, Ibrahim Vazirabad, Axel Kamp, Ulrike Wittig, Clemens Wrzodek, Finja Wrzodek, Ioannis Xenarios, Anna Zhukova, and Jeremy Zucker and. Sbml level 3: an extensible format for the exchange and reuse of biological models. *Molecular Systems Biology*, 16(8), August 2020.

[43] Lucian P. Smith, Stuart L. Moodie, Frank T. Bergmann, Colin Gillespie, Sarah M. Keating, Matthias König, Chris J. Myers, Maciek J. Swat, Darren J. Wilkinson, and Michael Hucka. Systems biology markup language (SBML) level 3 package: Distributions, version 1, release 1. *Journal of Integrative Bioinformatics*, 17(2-3), June 2020.

[44] Brett G. Olivier and Frank T. Bergmann. SBML level 3 package: Flux balance constraints version 2. *Journal of Integrative Bioinformatics*, 15(1), March 2018.

[45] Ralph Gauges, Ursula Rost, Sven Sahle, Katja Wengler, and Frank T. Bergmann. The systems biology markup language (SBML) level 3 package: Layout, version 1 core. *Journal of Integrative Bioinformatics*, 12(2):550–602, June 2015.

[46] Daniel A. Beard, Randall Britten, Mike T. Cooling, Alan Garny, Matt D.B. Halstead, Peter J. Hunter, James Lawson, Catherine M. Lloyd, Justin Marsh, Andrew Miller, David P. Nickerson, Poul M.F. Nielsen, Taishin Nomura, Shankar Subramanium, Sarala M. Wimalaratne, and Tommy Yu. CellML metadata standards, associated tools and repositories. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1895):1845–1867, May 2009.

[47] S. M. Wimalaratne, M. D. B. Halstead, C. M. Lloyd, E. J. Crampin, and P. F. Nielsen. Biophysical annotation and representation of CellML models. *Bioinformatics*, 25(17):2263–2270,

June 2009.

[48] Marco Mesiti, Ernesto Jiménez Ruiz, Ismael Sanz, Rafael Berlanga Llavori, Giorgio Valentini, Paolo Perlasca, and David Manset. Data integration issues and opportunities in biological XML data management. In *Open and Novel Issues in XML Database Applications*, pages 263–286. IGI Global, 2009.

[49] O. Caprotti and D. Carlisle. OpenMath and MathML: semantic markup for mathematics. *XRDS: Crossroads, The ACM Magazine for Students*, 6(2):11–14, November 1999.

[50] L. P. Smith, E. Butterworth, J. B. Bassingthwaighte, and H. M. Sauro. SBML and CellML translation in antimony and JSim. *Bioinformatics*, 30(7):903–907, November 2013.

[51] Michael Hucka, Frank T. Bergmann, Andreas Dräger, Stefan Hoops, Sarah M. Keating, Nicolas Le Novère, Chris J. Myers, Brett G. Olivier, Sven Sahle, James C. Schaff, Lucian P. Smith, Dagmar Waltemath, and Darren J. Wilkinson. Systems biology markup language (SBML) level 2 version 5: Structures and facilities for model definitions. *Journal of Integrative Bioinformatics*, 12(2):731–901, June 2015.

[52] Finja Büchel, Clemens Wrzodek, Florian Mittag, Andreas Dräger, Johannes Eichner, Nicolas Rodriguez, Nicolas Le Novère, and Andreas Zell. Qualitative translation of relations from BioPAX to SBML qual. *Bioinformatics*, 28(20):2648–2653, October 2012.

[53] Courtot, Juty N, Knupfer C, Waltemath D, Zhukova A, Drager A, Dumontier M, Finney A, and Golebiewski M. Controlled vocabularies and semantics in systems biology, 2011.

[54] John H. Gennari, Matthias König, Goksel Misirli, Maxwell L. Neal, David P. Nickerson, and Dagmar Waltemath. OMEX metadata specification (version 1.2). *Journal of Integrative Bioinformatics*, 18(3), September 2021.

[55] Natasha Noy. Ontology development 101: A guide to creating your first ontology. 2001.

[56] Janna Hastings, Gareth Owen, Adriano Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, and Christoph Steinbeck. ChEBI

in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Research*, 44(D1):D1214–D1219, October 2015.

[57] Woosub Shin, Joseph L. Hellerstein, Yuda Munarko, Maxwell L. Neal, David P. Nickerson, Anand K. Rampadarath, Herbert M. Sauro, and John H. Gennari. SBMate: A framework for evaluating quality of annotations in systems biology models. *BioRxivs*, October 2021.

[58] Nicolas Le Novère, Andrew Finney, Michael Hucka, Upinder S Bhalla, Fabien Campagne, Julio Collado-Vides, Edmund J Crampin, Matt Halstead, Edda Klipp, Pedro Mendes, Poul Nielsen, Herbert Sauro, Bruce Shapiro, Jacky L Snoep, Hugh D Spence, and Barry L Wanner. Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat Biotechnol*, 23(12):1509–1515, December 2005.

[59] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform resource identifier (uri): Generic syntax. Technical report, 2005.

[60] N Juty. Systems biology ontology: Update. *Nature Precedings*, 2010.

[61] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The semantic web: the roles of XML and RDF. *IEEE Internet Computing*, 4(5):63–73, 2000.

[62] R. Apweiler. UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 32(90001):115D–119, January 2004.

[63] N. Swainston and P. Mendes. libAnnotationSBML: a library for exploiting SBML annotations. *Bioinformatics*, 25(17):2292–2293, June 2009.

[64] N. Le Novere. BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Research*, 34(90001):D689–D691, January 2006.

[65] Michael Hucka, Frank T. Bergmann, Claudine Chaouiya, Andreas Dräger, Stefan Hoops, Sarah M. Keating, Matthias König, Nicolas Le Novère, Chris J. Myers, Brett G. Olivier, Sven

Sahle, James C. Schaff, Rahuman Sheriff, Lucian P. Smith, Dagmar Waltemath, Darren J. Wilkinson, and Fengkai Zhang. The systems biology markup language (SBML): Language specification for level 3 version 2 core release 2. *Journal of Integrative Bioinformatics*, 16(2), June 2019.

[66] BJ Bornstein, SM Keating, A Jouraku, and M Hucka. Libsbml: an api library for sbml. *Bioinformatics*, 24, 2008.

[67] Nicolas Rodriguez, Alex Thomas, Leandro Watanabe, Ibrahim Y. Vazirabad, Victor Kofia, Harold F. Gómez, Florian Mittag, Jakob Matthes, Jan Rudolph, Finja Wrzodek, Eugen Netz, Alexander Diamantikos, Johannes Eichner, Roland Keller, Clemens Wrzodek, Sebastian Fröhlich, Nathan E. Lewis, Chris J. Myers, Nicolas Le Novère, Bernhard Ø. Palsson, Michael Hucka, and Andreas Dräger. JSBML 1.0: providing a smorgasbord of options to encode systems biology models: Table 1. *Bioinformatics*, 31(20):3383–3386, June 2015.

[68] Dagmar Waltemath, Richard Adams, Daniel A. Beard, Frank T. Bergmann, Upinder S. Bhalla, Randall Britten, Vijayalakshmi Chelliah, Michael T. Cooling, Jonathan Cooper, Edmund J. Crampin, Alan Garny, Stefan Hoops, Michael Hucka, Peter Hunter, Edda Klipp, Camille Laibe, Andrew K. Miller, Ion Moraru, David Nickerson, Poul Nielsen, Macha Nikolski, Sven Sahle, Herbert M. Sauro, Henning Schmidt, Jacky L. Snoep, Dominic Tolle, Olaf Wolkenhauer, and Nicolas Le Novère. Minimum Information About a Simulation Experiment (MIASE). *PLoS Comput Biol*, 7(4):e1001122, April 2011.

[69] Anna Zhukova, Anna Zhukova, Dagmar Waltemath, Nick Juty, Camille Laibe, and Nicolas Le Novère. Kinetic simulation algorithm ontology. *Nature Precedings*, September 2011.

[70] Frank T. Bergmann, Jonathan Cooper, Matthias König, Ion Moraru, David Nickerson, Nicolas Le Novère, Brett G. Olivier, Sven Sahle, Lucian Smith, and Dagmar Waltemath. Simulation Experiment Description Markup Language (SED-ML) Level 1 Version 3 (L1V3). *J Integr Bioinform*, 15(1):20170086, March 2018.

[71] Lucian P Smith, Frank T Bergmann, Alan Garny, Tomáš Helikar, Jonathan Karr, David Nickerson, Herbert Sauro, Dagmar Waltemath, and Matthias König. The simulation experiment description markup language (sed-ml): language specification for level 1 version 4. *Journal of integrative bioinformatics*, 18(3), 2021.

[72] Kiri Choi, Lucian P. Smith, J. Kyle Medley, and Herbert M. Sauro. phraSED-ML: A paraphrased, human-readable adaptation of SED-ML. *J Bioinform Comput Biol*, 14(6):1650035, December 2016.

[73] Leonard Schmiester, Yannik Schälte, Frank T. Bergmann, Tacio Camba, Erika Dudkin, Janine Egert, Fabian Fröhlich, Lara Fuhrmann, Adrian L. Hauber, Svenja Kemmer, Polina Lakrisenko, Carolin Loos, Simon Merkt, Wolfgang Müller, Dilan Pathirana, Elba Raimúndez, Lukas Refisch, Marcus Rosenblatt, Paul L. Stapor, Philipp Städter, Dantong Wang, Franz-Georg Wieland, Julio R. Banga, Jens Timmer, Alejandro F. Villaverde, Sven Sahle, Clemens Kreutz, Jan Hasenauer, and Daniel Weindl. PEtab—interoperable specification of parameter estimation problems in systems biology. *PLOS Computational Biology*, 17(1):e1008646, January 2021.

[74] Veronica L. Porubsky, Arthur P. Goldberg, Anand K. Rampadarath, David P. Nickerson, Jonathan R. Karr, and Herbert M. Sauro. Best Practices for Making Reproducible Biochemical Models. *Cell Syst*, 11(2):109–120, August 2020.

[75] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The

FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1), March 2016.

[76] Falk Schreiber, Björn Sommer, Tobias Czauderna, Martin Golebiewski, Thomas E. Gorochowski, Michael Hucka, Sarah M. Keating, Matthias König, Chris Myers, David Nickerson, and Dagmar Waltemath. Specifications of standards in systems and synthetic biology: status and developments in 2020. *J Integr Bioinform*, 17(2-3):20200022, June 2020.

[77] Frank T Bergmann, Richard Adams, Stuart Moodie, Jonathan Cooper, Mihai Glont, Martin Golebiewski, Michael Hucka, Camille Laibe, Andrew K Miller, David P Nickerson, Brett G Olivier, Nicolas Rodriguez, Herbert M Sauro, Martin Scharm, Stian Soiland-Reyes, Dagmar Waltemath, Florent Yvon, and Nicolas Le Novère. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15(1):369, December 2014.

[78] Ahmet Erdemir, Lealem Mulugeta, Joy P. Ku, Andrew Drach, Marc Horner, Tina M. Morrison, Grace C. Y. Peng, Rajanikanth Vadigepalli, William W. Lytton, and Jerry G. Myers. Credible practice of modeling and simulation in healthcare: ten rules from a multidisciplinary perspective. *J Transl Med*, 18(1):369, December 2020.

[79] Steve R Blattnig, James M Luckring, Joseph H Morrison, Andre J Sylvester, Ram K Tripathi, and Thomas A Zang. NASA standard for models and simulations: Philosophy and requirements overview. *J. Aircr.*, 50(1):20–28, 2013.

[80] Elizabeth Howell and Dobrijevic Daisy. Columbia disaster: What happened and what nasa learned, 2021.

[81] Robert Niewoehner, Craig Steidle, and Eric Johnson. The loss of the space shuttle columbia: Portaging the leadership lessons with a critical thinking model. *American Society for Engineering Education*, 2008.

[82] Steve Blattnig, Lawrence Green, James Luckring, Joseph Morrison, Ram Tripathi, and Thomas Zang. Towards a Credibility Assessment of Models and Simulations. In *49th*

AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Schaumburg, IL, April 2008. American Institute of Aeronautics and Astronautics.

[83] Xing Chen, Chenggang Clarence Yan, Xiaotian Zhang, Xu Zhang, Feng Dai, Jian Yin, and Yongdong Zhang. Drug–target interaction prediction: databases, web servers and computational models. *Briefings in Bioinformatics*, 17(4):696–712, August 2015.

[84] Hyo-Jeong Kuh, Seong Hoon Jang, M. Guillaume Wientjes, and Jess L.S. Au. Computational model of intracellular pharmacokinetics of paclitaxel. *Journal of Pharmacology and Experimental Therapeutics*, 293(3):761–740, June 2000.

[85] Ethan Kung, Masoud Farahmand, and Akash Gupta. A hybrid experimental-computational modeling framework for cardiovascular device testing. *Journal of Biomechanical Engineering*, 141(5), March 2019.

[86] Marco Viceconti, Miguel A. Juárez, Cristina Curreli, Marzio Pennisi, Giulia Russo, and Francesco Pappalardo. Credibility of In Silico Trial Technologies—A Theoretical Framing. *IEEE Journal of Biomedical and Health Informatics*, 24(1):4–13, January 2020. Conference Name: IEEE Journal of Biomedical and Health Informatics.

[87] Marco Viceconti, Francesco Pappalardo, Blanca Rodriguez, Marc Horner, Jeff Bischoff, and Flora Musuamba-Tshinanu. In silico trials: Verification, validation and uncertainty quantification of predictive models used in the regulatory evaluation of biomedical products. *Methods*, 185, 2021.

[88] Christian Lieven, Moritz E Beber, Brett G Olivier, Frank T Bergmann, Meric Ataman, Parizad Babaei, Jennifer A Bartell, Lars M Blank, Siddharth Chauhan, Kevin Correia, Christian Diener, Andreas Dräger, Birgitta E Ebert, Janaka N Edirisinghe, José P Faria, Adam M Feist, Georgios Fengos, Ronan M T Fleming, Beatriz García-Jiménez, Vassily Hatzimanikatis, Wout van Helvoirt, Christopher S Henry, Henning Hermjakob, Markus J Herrgård, Ali Kaafarani, Hyun Uk Kim, Zachary King, Steffen Klamt, Edda Klipp, Jasper J Koehorst, Matthias König, Meiyappan Lakshmanan, Dong-Yup Lee, Sang Yup Lee, Sunjae Lee, Nathan E Lewis, Filipe

Liu, Hongwu Ma, Daniel Machado, Radhakrishnan Mahadevan, Paulo Maia, Adil Mardinoglu, Gregory L Medlock, Jonathan M Monk, Jens Nielsen, Lars Keld Nielsen, Juan Nogales, Intawat Nookaew, Bernhard O Palsson, Jason A Papin, Kiran R Patil, Mark Poolman, Nathan D Price, Osbaldo Resendis-Antonio, Anne Richelle, Isabel Rocha, Benjamín J Sánchez, Peter J Schaap, Rahuman S Malik Sheriff, Saeed Shoaie, Nikolaus Sonnenschein, Bas Teusink, Paulo Vilaça, Jon Olav Vik, Judith A H Wodke, Joana C Xavier, Qianqian Yuan, Maksim Zakhartsev, and Cheng Zhang. MEMOTE for standardized genome-scale metabolic model testing. *Nat. Biotechnol.*, 38(3):272–276, March 2020.

[89] Brett G Olivier and Frank T Bergmann. SBML level 3 package: Flux balance constraints version 2. *J. Integr. Bioinform.*, 15(1), March 2018.

[90] Matthias König. sbmlsim: Sbml simulation made easy, 2021.

[91] Mihai Glont, Tung V N Nguyen, Martin Graesslin, Robert Hälke, Raza Ali, Jochen Schramm, Sarala M Wimalaratne, Varun B Kothamachu, Nicolas Rodriguez, Maciej J Swat, Jurgen Eils, Roland Eils, Camille Laibe, Rahuman S Malik-Sheriff, Vijayalakshmi Chelliah, Nicolas Le Novère, and Henning Hermjakob. Biomodels: expanding horizons to include more modelling approaches and formats. *Nucleic Acids Research*, 46(D1):D1248–D1253, 2018.

[92] Brett Olivier, Willi Gottstein, Douwe Molenaar, and Bas Teusink. Cbmpy release 0.8.2, 2021.

[93] Archana Hari and Daniel Lobo. Fluxer: a web application to compute, analyze and visualize genome-scale metabolic flux networks. *Nucleic Acids Research*, 48(W1):W427–W435, May 2020.

[94] Karthik Raman. Matlab/cobra helper for frog analysis of fbc models, 2022.

[95] Chanchala Kaddi, Erica D Oden, Chang F Quo, and May D Wang. Exploration of quantitative scoring metrics to compare systems biology modeling approaches. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, August 2007.

[96] Joseph L. Hellerstein, Stanley Gu, Kiri Choi, and Herbert M. Sauro. Recent advances in biomedical simulations: a manifesto for model engineering. *F1000Research*, 8:261, March 2019.

[97] R.E. Fairley. Tutorial: Static analysis and dynamic testing of computer software. *Computer*, 11(4):14–23, 1978.

[98] Maria Güemes, Sofia A Rahman, and Khalid Hussain. What is a normal blood glucose? *Archives of Disease in Childhood*, 101(6):569–574, 2016.

[99] B.H.Thacker, S.W.Doebling, F.M.Hemez, M.C. Anderson, J.E. Pepin, and E.A. Rodriguez. Concepts of model verification and validation. Technical report, October 2004.

[100] Kiri Choi, J. Kyle Medley, Matthias König, Kaylene Stocking, Lucian Smith, Stanley Gu, and Herbert M. Sauro. Tellurium: An extensible python-based modeling environment for systems and synthetic biology. *Biosystems*, 171:74–79, September 2018.

[101] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. COPASI—a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, October 2006.

[102] Bilal Shaikh, Lucian P Smith, Dan Vasilescu, Gnaneswara Marupilla, Michael Wilson, Eran Agmon, Henry Agnew, Steven S Andrews, Azraf Anwar, Moritz E Beber, Frank T Bergmann, David Brooks, Lutz Brusch, Laurence Calzone, Kiri Choi, Joshua Cooper, John Detloff, Brian Drawert, Michel Dumontier, G Bard Ermentrout, James R Faeder, Andrew P Freiburger, Fabian Fröhlich, Akira Funahashi, Alan Garny, John H Gennari, Padraig Gleeson, Anne Goelzer, Zachary Haiman, Jan Hasenauer, Joseph L Hellerstein, Henning Hermjakob, Stefan Hoops, Jon C Ison, Diego Jahn, Henry V Jakubowski, Ryann Jordan, Matúš Kalaš, Matthias König, Wolfram Liebermeister, Rahuman S Malik Sheriff, Synchon Mandal, Robert McDougal, J Kyle Medley, Pedro Mendes, Robert Müller, Chris J Myers, Aurelien Naldi, Tung V N Nguyen, David P Nickerson, Brett G Olivier, Drashti Patoliya, Loïc Paulevé, Linda R Petzold, Ankita Priya, Anand K Rampadarath, Johann M Rohwer, Ali S Saglam, Dilawar Singh, Ankur Sinha, Jacky Snoep, Hugh Sorby, Ryan Spangler, Jörn Starruß, Payton J Thomas, David van Niekerk,

Daniel Weindl, Fengkai Zhang, Anna Zhukova, Arthur P Goldberg, James C Schaff, Michael L Blinov, Herbert M Sauro, Ion I Moraru, and Jonathan R Karr. BioSimulators: a central registry of simulation engines and services for recommending specific tools. *Nucleic Acids Research*, 50(W1):W108–W114, May 2022.

[103] Simeone Marino, Ian B. Hogue, Christian J. Ray, and Denise E. Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of Theoretical Biology*, 254(1):178–196, September 2008.

[104] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: the Primer*. Wiley, 2008.

[105] D. Loucks, E. van Beek, J. Stedinger, J. Dijkman, and M. Villars. *Water Resource Systems Planning and Management: An Introduction to Methods, Models, and Applications*. United Nations Educational, Scientific and Cultural Organization, 2005.

[106] W. Lal. Sensitivity and uncertainty analysis of a regional model for the natural system of south florida. Technical report, South Florida Water Management District, 1995.

[107] Zhike Zi, Yanan Zheng, Ann E Rundell, and Edda Klipp. SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool. *BMC Bioinformatics*, 9(1), August 2008.

[108] Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, 3(10):e189, October 2007.

[109] J. Vanlier, C.A. Tiemann, P.A.J. Hilbers, and N.A.W. van Riel. Parameter uncertainty in biochemical models described by ordinary differential equations. *Mathematical Biosciences*, 246(2):305–314, December 2013.

[110] Z. Zi. Sensitivity analysis approaches applied to systems biology models. *IET Systems Biology*, 5(6):336–346, November 2011.

[111] Simon van Mourik, Cajo ter Braak, Hans Stigter, and Jaap Molenaar. Prediction uncertainty assessment of a systems biology model requires a sample of the full probability distribution of its parameters. *PeerJ*, 2:e433, June 2014.

[112] Victoria Volodina and Peter Challenor. The importance of uncertainty quantification in model reproducibility. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197), March 2021.

[113] A. Somogyi, J.K. Medley, K. Choi, C. Welsh, and L. Smith. libroadrunner, release 1.1.16. `https://libroadrunner.readthedocs.io/_/downloads/en/latest/pdf/`.