Combining hypothesis- and data-driven neuroscience modeling in FAIR workflows

Olivia Eriksson1*, Upinder Singh Bhalla2, Kim T Blackwell3, Sharon M Crook4, Daniel Keller5, Andrei Kramer1,6, Marja-Leena Linne7, Ausra Saudargiene8,9, Rebecca C Wade10,11,12, Jeanette Hellgren Kotaleski1,6*

1Science for Life Laboratory, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden; 2National Center for Biological Sciences, Tata Institute of Fundamental Research, Bangalore, India; 3Department of Bioengineering, Volgenau School of Engineering, George Mason University, Fairfax, United States; 4School of Mathematical and Statistical Sciences, Arizona State University, Tempe, United States; 5Blue Brain Project, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland; 6Department of Neuroscience, Karolinska Institute, Stockholm, Sweden; 7Faculty of Medicine and Health Technology, Tampere University, Tampere, Finland; 8Neuroscience Institute, Lithuanian University of Health Sciences, Kaunas, Lithuania; 9Department of Informatics, Vytautas Magnus University, Kaunas, Lithuania; 10Molecular and Cellular Modeling Group, Heidelberg Institute for Theoretical Studies (HITS), Heidelberg, Germany; 11Center for Molecular Biology (ZMBH), ZMBH-DKFZ Alliance, University of Heidelberg, Heidelberg, Germany; 12Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Heidelberg, Germany

Abstract Modeling in neuroscience occurs at the intersection of different points of view and approaches. Typically, hypothesis-driven modeling brings a question into focus so that a model is constructed to investigate a specific hypothesis about how the system works or why certain phenomena are observed. Data-driven modeling, on the other hand, follows a more unbiased approach, with model construction informed by the computationally intensive use of data. At the same time, researchers employ models at different biological scales and at different levels of abstraction. Combining these models while validating them against experimental data increases understanding of the multiscale brain. However, a lack of interoperability, transparency, and reusability of both models and the workflows used to construct them creates barriers for the integration of models representing different biological scales and built using different modeling philosophies. We argue that the same imperatives that drive resources and policy for data – such as the FAIR (Findable, Accessible, Interoperable, Reusable) principles – also support the integration of different modeling approaches. The FAIR principles require that data be shared in formats that are Findable, Accessible, Interoperable, and Reusable. Applying these principles to models and modeling workflows, as well as the data used to constrain and validate them, would allow researchers to find, reuse, question, validate, and extend published models, regardless of whether they are implemented phenomenologically or mechanistically, as a few equations or as a multiscale, hierarchical system. To illustrate these ideas, we use a classical synaptic plasticity model, the Bienenstock–Cooper–Munro rule, as an example due to its long history, different levels of abstraction, and implementation at many scales.
Introduction

Dynamical models provide an essential counterpart to experiments in the endeavor to understand the brain, and today a large ecosystem of model types and approaches exists (Figure 1). For example, hypothesis-driven modeling typically brings a question into focus so that a model is constructed to investigate a specific hypothesis about how the brain works. Data-driven modeling, on the other hand, often follows a more unbiased approach, with model construction informed by the computationally intensive use of data. Although hypothesis- and data-driven modeling approaches are not mutually exclusive, a plethora of modeling formalisms, simulation platforms, and data formats has fragmented the neuroscience modeling community, particularly when it comes to modeling at different biological scales or levels of abstraction. This diversity is beneficial: most models and model building tools have a clear and specific role, but at the same time combining these approaches in an interoperable way would have an immense impact on our understanding of the brain. The FAIR (Findable, Accessible, Interoperable, Reusable) principles (Wilkinson et al., 2016) have been widely discussed and implemented for data management (Wittig et al., 2017), and recently also for computational workflows (Goble et al., 2020). We suggest that these principles would be beneficial for models and modeling workflows within neuroscience as well. In this review, we examine the different steps of a general modeling workflow and look at different alternatives for model building, refinement, analysis, and use (Figure 2). For each step we consider how the FAIR principles can be applied to the different aspects

**Figure 1.** Schematic illustration of different types of models and their relative relationships. A subset of neuroscience models are visualized based on four criteria: (A) biological scale, (B) level of abstraction, (C) the degree to which a hypothesis-driven, or (D) data-driven approach have been used for the model construction. Illustration includes examples from the text as well as several classical models: Albus, 1971 (hypothesis-driven phenomenological model of the cerebellar circuitry as a pattern recognition system); Bienenstock et al., 1982 (algebraic bidirectional synaptic plasticity rule); Bruce et al., 2019b (mechanistic molecular model of the regulation of adenyl cyclase 5 by G-proteins in corticostriatal synaptic plasticity induction); Dreyer et al., 2010 (data-driven mechanistic model of tonic and phasic dopamine release and activation of receptors in the dorsal striatum); Frémaux et al., 2013 (hypothesis-driven phenomenological model of a reward-modulated spike-timing-dependent learning rule for an actor-critic network); Gurney et al., 2015 (basal ganglia model with corticostriatal reinforcement learning using data-driven synaptic plasticity rules); Hayer and Bhalla, 2005 (data-driven biochemical bidirectional synaptic plasticity model involving calcium/calmodulin-dependent protein kinase II [CaMKII]); Hodgkin and Huxley, 1952 (classic biophysical model of action potentials); Knight, 1972 (hypothesis-driven phenomenological model of stimulus encoding into a neuronal population); Markram et al., 2015 (large-scale digital reconstruction of somatosensory cortex microcircuitry); Marr, 1969 (cerebellar algorithm model); Traub et al., 1994 (biophysical hippocampal CA3 neuron model showing the importance of dendritic ion channels).
of the modeling process. We focus on models at the intracellular or cellular level, where the field of computational neuroscience meets systems biology modeling, with an emphasis on models of brain plasticity and learning. However, the discussed methodology and concepts can be applied to other biological systems and scales as well (see, e.g., Einevoll et al., 2019).

As an illustration of the wide variety of model types, Figure 1 depicts a number of models arranged according to four criteria: biological scale, level of abstraction, and to what degree hypothesis- and data-driven approaches were used during model development. The axes of the figure are qualitative, additional dimensions could be considered, and the precise arrangement of the models could be interpreted differently as there are no strict metric spaces defined. Regarding biological scale, we note that experimental data can be measured and applied at different resolutions — from a molecular signaling pathway in a single synapse to an entire brain. This is reflected in the models, which exist, for example, from the level of molecular interactions important for plasticity in specific synapses (Bruce et al., 2019b) to brain structures with an aim of understanding generalized learning rules (Gurney et al., 2015). Because of different modeling goals, models are also constructed with different levels of abstraction. For example, the dopamine signal is important for reward learning, addiction, and motivation, and has been modeled phenomenologically with a reward-prediction error (Frémaux et al., 2013) or more mechanistically, taking into account dopamine diffusion and the affinity of dopamine receptors (Dreyer et al., 2010). Figure 1 also includes the data- and hypothesis-driven aspects of models. Models that have a high score on the data axis are typically based on a large amount of quantitative data and/or have been developed using data science methodology. Models that have a high score on the hypothesis axis, on the other hand, are strongly driven by a particular question or idea, and one aim could be to investigate if some assumed mechanism explains the studied phenomena. As a side note, we have used the concepts of ‘top-down’ and ‘bottom-up’ modeling restrictively within
In order to further develop modeling workflow capabilities fulfilling the FAIR criteria, a tightly integrated ecosystem of databases, software, and standardized formats is needed. Many of these pieces exist, supporting the modeling process from initial model development, to the use of models in exploration and prediction. For example, over the years, the field of computational neuroscience has developed a set of open-source simulation environments (Gewaltig and Cannon, 2014). At the same time, the community is depositing published models into online repositories, such as model databases (Birgiolas et al., 2015; Gleeson et al., 2019; Hines et al., 2004; Li et al., 2010) and GitHub (github.com), for others to inspect or use. Simulator independent, standardized model specifications such as the Systems Biology Markup Language (SBML) (Hucka et al., 2003) and NeuroML (Gleeson et al., 2010) have been developed to promote model interoperability and reproducibility. Finally, increasingly, experimental data are being used extensively in the modeling process, and frameworks for model calibration (Ashyraliyev et al., 2009; Mitra and Hlavacek, 2019; Sun et al., 2012) and for validating models against data (Omar et al., 2014) are available, or under development. What
remains is to develop standardized workflows and data formats that integrate the components of this ecosystem to facilitate reusing, extending, and comparing models, within and across scales. The aim is to promote interoperability among different software packages used within neuroscience, to increase shareability of models, and at the same time improve transparency and reproducibility with regard to the model building process and the data used for parameter calibration and validation. All this can in the end also contribute to increased automation of the entire process to set the stage for wider use of data intensive methods.

As we consider the modeling workflow shown in Figure 2, we discuss how FAIR principles can be introduced to this process. To illustrate these steps with a concrete example, we use a classical model of activity-dependent synaptic plasticity, the Bienenstock–Cooper–Munro (BCM) rule (Bienenstock et al., 1982), described in detail below. Here, plasticity depends on pre- and postsynaptic activity, and leads to long-term depression (LTD) or long-term potentiation (LTP). The BCM rule is chosen as an example because it was first modeled phenomenologically and also has been subsequently reproduced using mechanistic models with increasing levels of detail as more data and knowledge have accumulated (Lisman, 1989). More recent models of signaling underlying the BCM rule are substantially data driven (Castellani et al., 2005; Hayer and Bhalla, 2005).

**An example: the BCM rule**

The BCM rule is a synaptic plasticity model (Bienenstock et al., 1982) formulated in the context of the development of orientation selectivity in the visual system. It states that as postsynaptic activity increases, there are two domains of plasticity that induce either depression or potentiation. This rule has since been used as the basis for considerable theoretical and experimental work on synaptic plasticity, and here it is employed to illustrate approaches for modeling plasticity and to motivate our discussion of modeling workflows and FAIR approaches.

The original BCM rule describes the rate of change of synaptic weight $m$ as

$$\frac{dm(t)}{dt} = \phi(c(t)) \ s(t) - \epsilon \ m(t).$$

where $s$ is the synapse input current, $\epsilon$ is the time constant of synaptic decay, and $c$ is the postsynaptic activity. $\phi$ represents the postsynaptic activation function, formulated as

$$\phi(c) < 0 \text{ for } c < \Theta_M \text{ and } \phi(c) > 0 \text{ for } c > \Theta_M$$

where $\Theta_M$ is the activity threshold at which the synaptic strengths are modified. The BCM rule is phenomenological because $\phi$ does not map to any biological mechanisms, and the authors showed that this learning rule can account for the formation of orientation selectivity for a wide range of values for $\Theta_M$. A typical graph of $\phi$ is depicted in Figure 3A, and similar curves have been obtained experimentally (Kirkwood et al., 1996).

This model was published in a widely read society journal and predates today’s FAIR principles. The article included simulations of the model equations, but, lacking today’s open-source, code-sharing infrastructure, readers were left to their own skills to replicate these implementations. Like many abstract models, the BCM model is a mostly qualitative representation of concepts derived from a substantial amount of data, including many features of synaptic plasticity, network-level data for orientation selectivity, and data for the degradation of orientation selectivity in the case of monocular deprivation and similar experiments. The publication includes simulations of several such cases, showing a qualitative match to data but does not explain how parameters were derived. In summary, due to the minimal, abstract nature of the model, the BCM study is not difficult to reproduce; however, there is no reference implementation available for simulations, and the parameters for several simulations are not available. Thus, the published simulations and parameters are not consistent with today’s FAIR principles.

Moving forward a few decades, one finds mechanistic, mass-action models of synaptic signaling and the BCM rule, where the relevance of FAIR and data-driven approaches becomes more apparent. First, the quantities under consideration (input activity and synaptic strength) correspond directly to synaptic molecules. Experimental data tell us that intracellular calcium concentration is a good proxy for postsynaptic activity, $c$, and that the phosphorylation state of the AMPA receptor is one of the proxies for synaptic efficacy, $m$. Filling in the reaction diagram, one obtains a simplified reaction network of the form shown in Figure 3B. Remarkably, the properties of these molecules result in...
Figure 3. Abstract and mechanistic versions of the Bienenstock–Cooper–Munro (BCM) rule model. (A) Original version where the rate of plasticity change, $\phi$, is a function of the stimulus strength $c$. At the threshold $\Theta_M$ the sign of synaptic change flips from negative to positive. (B) Simplified mechanistic (chemical) model based on known pathways that could implement the BCM rule. The calcium stimulus activates both a kinase, CaMKII, and a

Figure 3 continued on next page
in a response that resembles the BCM curve. The phosphatase calcineurin (CaN) has a high affinity for calcium bound calmodulin, and also can partly become activated directly by calcium (Cremer, 2020). Hence, it is activated at moderate calcium and triggers an initial decrease in phosphorylated AMPAR and LTD (ϕ(c) < 0) from the BCM formulation). Calcium/calmodulin-dependent protein kinase II (CaMKII) has a weaker affinity for calcium bound calmodulin, but is present in overwhelming amounts, so its activation surpasses CaN at high calcium concentrations, thus implementing ϕ(c) > 0 for c > ΘM, and leading to increase in phosphorylated AMPAR and thus LTP. The resulting curve from this mechanistic but simplified model matches the BCM curve (Figure 3C). More detailed mechanistic versions of the BCM model have been implemented using known synaptic chemistry in considerably greater detail (Lisman, 1989; Hayer and Bhalla, 2005).

How FAIR and data driven are these more recent models? The Lisman, 2017 is not consistent with our current definition of FAIR: it provides full disclosure of equations and rate constants, but reuse requires reimplementation. The more recent model (Hayer and Bhalla, 2005) exists in at least two open-access databases, the Database of Quantitative Cellular Signaling, DOQCS (Sivakumaran et al., 2003), and BioModels (Li et al., 2010), hence is findable and accessible. It has been converted to the standardized SBML model description format, promoting interoperability and reusability. The DOQCS version has citations and calculations for the derivation of some of the parameters, and the BioModels version maintains the provenance, referencing the DOQCS entry from which it was derived.

Parameter access greatly improves as models become FAIR. For example, Lisman’s study acknowledges the original BCM model as a key motivation (Lisman, 1989), but no parameters were available, leading Lisman to use ‘plausible’ estimates for rates that form the BCM curve. In contrast, Hayer and Bhalla had access to reaction schemes and rates from databases such as DOQCS (Sivakumaran et al., 2003), and BioModels (Li et al., 2010), hence is findable and accessible. It has been converted to the standardized SBML model description format, promoting interoperability and reusability. The DOQCS version has citations and calculations for the derivation of some of the parameters, and the BioModels version maintains the provenance, referencing the DOQCS entry from which it was derived.

To summarize, the history of models that implement the BCM rule demonstrates a clear progression from abstract, qualitative, and unFAIR formulations to more mechanistic, quantitative, and FAIR formulations of the same conceptual model using data-driven parameterization. This progression is a natural consequence of the accumulation of experimental data and nascent concepts of FAIR principles. In what follows, we consider how to improve data-driven models further through FAIR workflows and data science-based approaches.

Framework for FAIR modeling workflows

A typical model development process can be divided into different stages or modules, as illustrated in Figure 2. The model foundation (Figure 2, box 1) stage corresponds to the process of collecting the experimental data and prior knowledge that will be used for the rest of the modeling process and combine these into an initial plan for the model structure (or topology), that is, a description of the different model entities and how they interact. In the next stage, the specification of the model and data (Figure 2, box 2a), the information from step 1 is formalized into a standardized, machine-readable format. During the model refinement stage (Figure 2, box 2b), the model is transferred to a mathematical formalism, simulated, calibrated, and refined. After model refinement is complete, the specification files are to be updated with this new information. Finally, during the model usage...
Review article

Computational and Systems Biology | Neuroscience

Eriksson et al. eLife 2022;11:e69013. DOI: https://doi.org/10.7554/eLife.69013  8 of 31

Table 1. Databases in cellular neuroscience and systems biology. These are some of the commonly used databases for creating and constraining models at the intracellular and cellular scale.

| Database, alphabetically | Purpose/focus | Reference | Homepage |
|-------------------------|--------------|-----------|----------|
| **Computational models** |  |  |  |
| 1 BioModels Database    | Physiologically and pharmaceutically relevant mechanistic models in standard formats | Li et al., 2010 | http://www.ebi.ac.uk/biomodels/ |
| 2 MoDEL Central Nervous System | Atomistic-MD trajectories for relevant signal transduction proteins |  | http://mmb.irbbarcelona.org/MoDEL-CNS/ |
| 3 NeuroML-DB            | Models of channels, cells, circuits, and their properties and behavior | Birgiolas et al., 2015 | https://neuroml-db.org/ |
| 4 NeuroElectro          | Extract and compile from literature electrophysiological properties of diverse neuron types | Tripathy et al., 2014 | https://neuroelectro.org |
| 5 ModelDB               | Computational neuroscience model | McDougal et al., 2017 | https://senselab.med.yale.edu/modeldb/ |
| 6 Ion Channel Genealogy | Ion channel models | Podlaski et al., 2017 | https://icg.neurotheory.ox.ac.uk/ |
| **Experimental data**   |  |  |  |
| 7 Allen Brain Atlas     | Human and mouse brain data | Lein et al., 2007 | http://www.brain-map.org/ |
| 8 BRENDA                | Enzyme kinetic data | Chang et al., 2021 | https://www.brenda-enzymes.org/ |
| 9 CRCNS - Collaborative Research in Computational Neuroscience | Forum for sharing tools and data for testing computational models and new analysis methods | Teeters et al., 2008 | https://CRCNS.org |
| 10 NeuroMorpho          | Neuronal cell 3D reconstructions | Ascoli et al., 2007 | http://neuromorpho.org/ |
| 11 Protein Data Bank (PDB) | 3D structures of proteins, nucleic acids, and complex assemblies | wwPDB consortium, 2019 | http://www.wwpdb.org/ |
| 12 Sabio-RK             | Curated database on biochemical reactions, kinetic rate equations with parameters and experimental conditions | Wittig et al., 2012 | https://sabio.h-its.org/ |
| 13 Yale Protein Expression Database (YPED) | Proteomic and small molecules | Colangelo et al., 2019 | https://medicine.yale.edu/keck/nida/yped/ |
| **Experimental data and models** |  |  |  |
| 14 Channelpedia         | Ion channel data and channel models | Ranjan et al., 2011 | https://channelpedia.epfl.ch/ |
| 15 DOQCS The Database of Quantitative Cellular Signaling | Kinetic data for signaling molecules and interactions | Sivakumaran et al., 2003 | http://doqcs.ncbs.res.in/ |
| 16 EBRAINS (including EBRAINS Knowledge Graph) | Digital research infrastructure that gathers data, models and tools for brain-related research |  | https://ebrains.eu (https://search.kg.ebrains.eu) |
| 17 FAIRDOMHub           | The FAIRDOMHub is a repository for publishing FAIR Data, Operating procedures and Models for the Systems Biology community | Wolstencroft et al., 2017 | https://fairdomhub.org/ |
| 18 Open Source Brain    | A resource for sharing and collaboratively developing computational models of neural systems | Gleeson et al., 2019 | https://www.opensourcebrain.org/ |

stage (Figure 2, box 3), the resulting model is validated, analyzed, and used to predict or investigate additional phenomena. Some examples of tools and formats that can be used in this process are given in Table 1–4. Note that these tables do not provide an exhaustive list, but rather a starting point for discussion.

**Model foundation: experimental data and prior knowledge**

The Model foundation (Figure 2, box 1) may include information from published literature, novel unpublished experimental data, experimental data and models retrieved from databases, or, for multi-scale modeling, information from models at other biological scales (Table 1). These different data
sources may include qualitative as well as quantitative data. Qualitative data give information about qualitative traits such as the possibility that two proteins interact or knowledge of positive or negative correlations, which are useful for defining the structure of the model, that is the modeling entities and how they interact. As an example, in the case of intracellular pathway models, such model structure defines which molecular species and reactions to consider, as exemplified in the chemical signaling network in Figure 3B. Quantitative data, on the other hand, are needed to specify quantitative entities in models, such as rate constants, equilibrium constants, or initial concentrations of molecules. This can be done either through direct measurements or indirectly during the model refinement stage (Figure 2, box 2b), where model outputs are compared to experimental readouts.

Findable data, data provenance, and data reusability are key components of the FAIR principles (Wittig et al., 2017) and are important for this first step of the modeling process. FAIR modeling workflows are facilitated by the many databases that can be used to constrain neuroscience models, which also includes databases for models from earlier studies. As mentioned above, the focus of this review is on cellular- and intracellular-level models, and data and model sharing resources for this level can be found in Table 1. For example, summary statistics (e.g. [Tripathy et al., 2014]) and morphological data for different neuron types (e.g. [Ascoli et al., 2007]), and integrated data for gene expression, connectivity, and neuroanatomy (e.g. Allen Brain Atlas [Lein et al., 2007; Sunkin et al., 2013]) are useful for constructing models at the cellular level. Often, published computational neuroscience models are shared via ModelDB (McDougal et al., 2017), which includes over 1700 models and is tightly coupled with NeuronDB, a database of neuronal properties that are used to constrain models based on experimental observations. Other repositories of neuroscience models include NeuroML-DB with over 1500 models in NeuroML format at multiple scales (Birgiolas et al., 2015), and Open Source Brain, a platform for collaborating, simulating, and sharing neuroscience models (Gleeson et al., 2019). Systems biology databases, such as BRENDA (Chang et al., 2021) and SABIO-RK (Wittig et al., 2018), contain kinetic data on enzyme kinetics and protein–protein interactions, which are critical for modeling subcellular signaling pathways. SABIO-RK is a curated database that was specifically designed to facilitate systems biology modeling, which provides reaction kinetics data along with information on experimental conditions, units, and kinetic rate equations. Systems biology models are also available through model databases such as BioModels, which includes many models that are relevant to systems biology level models in neuroscience (Malik-Sheriff et al., 2020).

One important feature of databases for experimental data and models is the accompanying metadata, which are critical for finding and using these data. For example, associating ontology references to biological terms makes it possible to use the associated domain knowledge in formulating search strategies (Birgiolas et al., 2015) and in the model building process.

For the synaptic plasticity model example described here, the models that employ the BCM rule have progressively used more information and data from different databases as new models have been developed. As mentioned above, mechanistically detailed versions of the synaptic plasticity model use model reaction schemes and reaction rates from existing databases: DOQCS (accessions 59–64) and the BioModels Database. Additionally, several subsequent cellular and network models available in ModelDB rely on various versions of the synaptic plasticity model (Jedlicka et al., 2015; Wilmes and Clopath, 2019; Mäki-Marttunen et al., 2020; Manninen et al., 2020).

In spite of the large amounts of data available at different biological scales, data are still sparse if one wants to build mechanistic, ‘bottom-up’ models to better understand multiscale, causal chains of events such as how properties of proteins affect cellular-level phenomena or how cellular and synaptic properties affect network dynamics and function (Klipp et al., 2010). However, it is sometimes possible to use predictions from a model at finer biological resolution to provide constraints to model parameters at the next level of abstraction (Boras et al., 2015; Stein et al., 2007; Wang et al., 2018; Xie et al., 2014). For example, molecular dynamics simulations which use biomolecular structural data can provide important quantitative or qualitative constraints on kinetic parameters, binding affinities, and their modulation by allosteric interactions in intracellular signaling pathway models (Bruce et al., 2019b; Bruce et al., 2019a; Gabdoulline et al., 2003; van Keulen et al., 2022). Tools exist to facilitate the use of biomolecular structural data in model building (Stein et al., 2007). For example, SYCAMORE (Weidemann et al., 2008) can use protein structural data along with published kinetic measurements for parameter assignment in the construction of signaling pathway models. Additional modeling tools and use cases for reuse of model components in multiscale models
Specifications of model and data

An important part of a FAIR modeling workflow is the specification of the model and experimental data (Figure 2, box 2a). Standardized formats for models have been developed with a goal of making it possible to efficiently reproduce the modeling results of another laboratory. Specifically, standardized model formats are critical for both interoperability and reuse. If we want to reproduce the entire modeling process, including model refinement, this requires information not only about the model, but also the quantitative data used for calibration and validation, a description of the different simulation experiments performed (corresponding to the biological experiments to be reproduced or predicted), as well as the prior assumptions made on parameters, all in machine readable format. A minimal requirement on such a specification is (but not limited to):

- the model
  - the model constituents, for example molecular species in a biochemical signaling model;
  - the interactions, for example reactions in a biochemical signaling model or synaptic connectivity in neuronal network model;
  - the parameters, for example synaptic conductances, reaction rates;
- the simulation experiments
  - the changes that are made to the model to replicate each of the biological experiments in for example initial conditions, parameters, or input functions;
- the quantitative experimental data
  - used to constrain model parameters;
  - used for validation – if model validation was performed;
  - the mapping between model output and experimental data readouts;
- the prior (i.e., before the model refinement) assumptions on parameter values;
- the posterior (i.e., after the model refinement) estimates for parameter values.

The prior assumptions on parameter values inform the model refinement process and may contain parameter ranges or distributions, or previously estimated values based on data, including error bounds. To be complete, the model and data specification may also include metadata on the calibration method used and identifiers for all model entities. This recommendation is similar to MIASE (Minimum Information About a Simulation Experiment) compliance; in Waltemath et al., 2011 the authors list more specific measures that authors can take to make their models easier to simulate by others. These simulation experiments can range from a time series simulation, to a parameter scan, to a sensitivity or bifurcation analysis, methods that are further described below. COMBINE (co.mbine.org, Hucka et al., 2015) is an initiative to coordinate the development of the various community standards and formats for computational models.

There are many ways to archive and share this information, and we discuss several possibilities. One approach is to maintain all information in one location, but often, a more distributed approach is required. In either case, the information should be clearly described and linked through metadata, including unique persistent identifiers to the different components like models or experimental data. In Table 2, we list various model description standards and file formats that can be used, and in later sections we describe ways to retrieve permanent unique identifiers.

At the systems biology level, examples of existing storage formats for the information above include SBtab (Lubitz et al., 2016) and the JSON format used in FindSim (Viswan et al., 2018). SBtab is well structured, so it is machine readable as well as human readable, and it is able to capture information about biochemical network models and associated calibration data in a single location. SBtab has defined fields for sbo (systems biology ontology) terms (Courtot et al., 2011), as well as database identifiers (e.g. to UniprotKB [UniProt Consortium, 2021] or KEGG [Kanehisa and Goto, 2000]), and additional information can be included to map between experimental readouts and model simulation outputs. In the FindSim framework, an experiment is codified as a file with information about the experimental stimuli and the readouts, as well as how they map to the model output.

SBML provides a standardized machine readable markup language for sharing only the model description that is supported by many simulation platforms. For example, the MOOSE and STEPS
simulators accept models in the SBML format. One advantage of SBML is that it describes all aspects of a model using a simulator-independent approach, but a disadvantage is that an SBML file can be hard to write by hand so that SBML files typically are created using an applications programming interface (API) such as libSBML. In some cases, a simulation platform may provide an API as the primary method for model specification. This API approach is taken by packages such as rxdd (Newton et al., 2018). Another approach to model handling, taken by VFGEN (vector field generator, Weckesser, 2008), is independent of biology but specific to differential equations. VFGEN converts an input file

Table 2. Model standards and file formats in cellular neuroscience and systems biology.
The formats described in this table allow standardized representation of models and their porting across simulation platforms.

| Name          | Purpose                                         | Webpage                                      | Reference          |
|---------------|-------------------------------------------------|----------------------------------------------|--------------------|
| **Formats for intracellular models in systems biology** |                                                 |                               |                    |
| 1 SBML        | Systems biology markup language, for storing and sharing models | https://sbml.org                       | Hucka et al., 2003 |
| 2 STab        | Systems Biology tables, for storing models (and data for parameter estimation) in spreadsheet form | https://shtab.net/                      | Lubitz et al., 2016|
| 3 CellML      | Store and exchange mathematical models, primarily in Biology | https://www.cellml.org/                  | Kohl et al., 2001  |
| **Formats for cellular and network-level models in Neuroscience** |                                                 |                               |                    |
| 4 NeuroML     | A XML-based description language that provides a common data format for defining and exchanging descriptions of neuronal cell and network models | http://www.neuroml.org/                | Gleeson et al., 2010|
| 5 NineML      | Unambiguous description of neuronal network models | https://github.com/INCF/nineml-spec        | Raikov et al., 2011|
| 6 NestML      | Domain-specific language for the specification of neuron models (python) | https://github.com/nest/nestml            | Plotnikov et al., 2016|
| **Custom formats for specific simulators** |                                                 |                               |                    |
| 7 sbproj      | Simbiology Project file                          | https://se.mathworks.com/products/simbiology/html | Schmidt and Jirstrand, 2006|
| 8 COPASI project file | COPASI native format for models and simulations | http://copasi.org/                  | Hoops et al., 2006 |
| 9 SONATA      | Efficient descriptions of large-scale neural networks | https://github.com/AllenInstitute/sonata | Dai et al., 2020 |
| 10 JSON (HillTau) | JSON files for FindSim and HillTau model reduction method | https://github.com/BhallaLab/HillTau | Bhalla, 2020 |
| 11 MOD         | Expanding NEURON’s repertoire of mechanisms with NMODL | https://www.neuron.yale.edu/neuron/      | Hines and Carnevale, 2000|
| **Formats for specification of parameter estimation problems** |                                                 |                               |                    |
| 12 STab        | Systems Biology tables, for storing both models and data for parameter estimation in spreadsheet form | https://shtab.net/                      | Lubitz et al., 2016|
| 13 PEtab       | Interoperable specification of parameter estimation problems in systems biology | https://github.com/PEtab-dev/PEtab      | Schmiester et al., 2021|
| **Formats for specification of experiments and data** |                                                 |                               |                    |
| 14 SED-ML      | Simulation Experiment Description Markup Language | https://sed-ml.org                    | Waltemath et al., 2011|
that describes a differential equation to source code for many different solvers in various languages (e.g., cvode in C, javascript, gsl in C, RADAU5 in Fortran).

For neuron and circuit models at other scales, several standards have been developed and adopted to varying degrees for use by simulation platforms. Examples of these include NeuroML, CellML, SONATA, and others (see Table 2). In particular, NeuroML is an International Neuroinformatics Coordinating Facility (INCF) (Abrams et al., 2021) endorsed standard supported by dozens of downstream software packages. Well-tested software libraries can be used to convert NeuroML model description files to code for specific simulators such as NEURON or MOOSE or to programming languages such as Python or Matlab (Gleseson et al., 2010). NeuroML also includes defined fields for ontological metadata, such as the terminology provided by the Neuroscience Information Framework (NIF) ontologies (Bug et al., 2008). There are standards for data as well, which aid the comparison of model simulation outputs to experimental data. Neurodata Without Borders (NWB) (Teeters et al., 2015; Rübel et al., 2019) and neuroscience interchange format (NIX) (Stoewer et al., 2014) are both standards for describing electrophysiology data that are recognized by the INCF.

How should the abstract and the mechanistic BCM models be specified using a standardized approach? The abstract version of the model should be defined using an existing simulator independent standard, such as SBtab or SBML, and should include the published data for the shape of the curve (Kirkwood et al., 1996). Similarly, the mechanistic version of the BCM model should be specified using SBML or SBtab, where the chemical experiments are defined using, for example, SBtab or JSON for FindSim. An important validation step for the mechanistic model would be to recreate the same final curve used to constrain the abstract model, demonstrating that the two formulations converge. To be able to reproduce the parameter estimation process, information on prior assumptions made on possible parameter ranges should be added to the files, as well as metadata concerning the optimization procedure.

**Model simulation**

Over the past 25 years, many general purpose and domain-specific simulation tools have been developed for modeling and simulation of biochemical and biophysical events in cells and circuits and even whole brain regions (Figure 2, boxes 2b and 3). Representative examples of these tools are provided in Table 3.

In simulations, molecular reactions and cellular- or network-level behavior and interactions are often described by a set of ordinary differential equations (Hodgkin and Huxley, 1952) or stochastic differential equations (Manninen et al., 2006). Standardized model specifications like those described above are interpreted by simulators as terms and parameters in the differential equations, and given appropriate initial values, the mathematical equations are solved using numerical solvers. In systems biology, Boolean networks have also been applied successfully, for example, SBtab or JSON for FindSim. An important validation step for the mechanistic model would be to recreate the same final curve used to constrain the abstract model, demonstrating that the two formulations converge. To be able to reproduce the parameter estimation process, information on prior assumptions made on possible parameter ranges should be added to the files, as well as metadata concerning the optimization procedure.

For biochemical models, a common approach is to model them deterministically using the law of mass action and Michaelis–Menten kinetics; these models are continuous in state space and macroscopic. This approach is accurate for large numbers of molecules and is more computationally efficient than stochastic approaches, when applicable. In contrast, stochastic approaches are more accurate but numerically costly; however, stochastic methods must be favored at low molecule numbers to give accurate results. Reaction–diffusion simulations, which take into account that particular molecules can diffuse, require additional geometrical information. The geometry can be relatively simple such as the morphology of a reconstructed neuron in a multicompartment model or more intricately detailed such as a tetrahedral mesh. Some reaction–diffusion simulators, such as MOOSE (Ray et al., 2008), NeuroRD (Jedrzejewski-Szmek and Blackwell, 2016), and STEPS (Hepburn et al., 2012), allow both stochastic and deterministic modes of simulation.

Table 3 provides examples of widely used model simulation frameworks for different spatial scales – from the molecular level to networks of spiking neurons and whole brain regions (Brette et al., 2007; Sanz-Leon et al., 2015; Einevoll et al., 2019) It is, of course, valuable when such simulators and other software are interoperable. This is facilitated by standardized, machine readable formats for model descriptions. Interoperability can have another meaning in addition to file format compatibility among simulators and other software: the possibility to run two simulation frameworks
Table 3. Software for model simulation.

These tools span a wide range of scales and levels of abstraction.

| Name          | Purpose                                                                 | Interchange file formats supported | Homepage                                      | RRID            | Reference                          |
|---------------|-------------------------------------------------------------------------|------------------------------------|-----------------------------------------------|-----------------|------------------------------------|
| **Molecular level** |                                                                 |                                     |                                               |                 |                                    |
| 1 BioNetGen   | Rule-based modeling framework (NFsim)                                   | BNGL, SBML                         | http://bionetgen.org/                         |                 | Harris et al., 2016                |
| 2 COPASI      | Biochemical system simulator                                            | SBML                              | http://copasi.org/                           | SCR_014260      | Hoops et al., 2006                 |
| 3 IQM Tools   | Systems Biology modeling toolbox in MATLAB, successor to SBPOP          | SBML                              | https://iqmtools.intiquan.com/                |                 |                                    |
| 4 MCell       | Stochastic diffusion simulator to model intracellular signaling pathways| SBML                              | https://mcell.org/                           | SCR_007307      |                                    |
| 5 NeuroRD     | MATLAB’s systems biology toolbox (Mathworks)                            | XML                                | http://krasnow1.gmu.edu/CENlab/software.html  | SCR_014769      | Oliveira et al., 2010              |
| 6 Simbiology  | Simulation tool for cellular signaling and biochemical pathways to build systems that describe reaction–diffusion of molecules and membrane potential | SBML                              | https://www.mathworks.com/products/simbiology.html |                 | Schmidt and Jirstrand, 2006        |
| 7 STEPS       | Simulation tool for deterministic, stochastic, and hybrid deterministic–stochastic models of molecular reactions, diffusion and electrophysiology | SBML                              | http://steps.sourceforge.net/STEPS/default.php | SCR_008742      | Hepburn et al., 2012               |
| 8 VCell       | Simulation environment to build and use computational models of neurons and networks of neurons; also subcellular simulations with the reaction–diffusion module | SBML, CellML                       | https://vcell.org/                           | SCR_007421      | Schaff et al., 1997               |
| **Cellular level** |                                                                 |                                     |                                               |                 |                                    |
| 9 NEURON      | Simulation tool for spiking neural networks                              | SBML, SONATA (after conversion) for networks, but can also use NeuroML and SBML | https://neuron.yale.edu/neuron/                | SCR_005393      | Carnevale and Hines, 2009; Hines and Carnevale, 1997 |
| **Network level** |                                                                 |                                     |                                               |                 |                                    |
| 10 BRIAN      | Simulation tool for spiking neural networks                              | SBML                              | http://briansimulator.org/                    | SCR_002998      | Goodman and Brette, 2008; Stimberg et al., 2019 |

Table 3 continued on next page
in parallel (e.g., at different biological scales), denoted cosimulation, and establish communication between them (Cannon et al., 2007; Djurfeldt et al., 2010; Newton et al., 2018).

Model refinement

Once a model can be simulated, the next critical step in model development is model refinement: the adjustment of the model structure and parameter values (Figure 2, box 2b). Here, we emphasize automation and the use of new data science methodologies to improve this process. This is important as the amount of data rapidly increases, and models become more complex. Other important aspects of model refinement are the inclusion of measures of uncertainty into the parameter estimates and predictions, as well as making the refinement process reproducible.

Model structure constraints

There are often physical or mathematical conditions that relate to the structure of the model, which must be respected for the model to be valid, in most cases to make sure that physical laws of some kind are not violated. Automatic methods for such structure refinement require machine readable standardized model specifications that can be updated easily in an automatic fashion, like those described above. Such standardized formats also enable interoperability so that different software can be used for different types of refinement on the same model.

For models at the biological scale of neurons or circuits, model structure refinement might include checking morphological structures of dendrites or axons for closed paths (Ascoli et al., 2007) or performing unit checking. The use of model description standards can aid these efforts by allowing easy access to model components in an object-oriented way, and at least one library for interacting with description standards provides support for unit checking (Cannon et al., 2014).
Models at the subcellular level often require structure refinements that include modifications due to mass conservation laws and thermodynamic parameter relationships (Wegscheider, 1902; Klipp et al., 2016, chapter 4). Mass conservation means that the total quantity of a substance should not change, for example the total of all forms of protein kinase A (holoenzyme or bound to cAMP) should remain constant. The exception is when molecules are deliberately introduced or removed from the simulation. Thermodynamic constraints require that the combined reaction rates from a set of substrates to a set of products need to be the same regardless of the route taken. Checking for conservation laws and thermodynamic constraints can be done in a semi-automatic fashion through stoichiometric analysis, as is done for conserved substances by the Copasi simulator (Hoops et al., 2006). Software is also available to introduce such relationships semiautomatically into the SBtab format (Santos et al., 2021).

Parameter estimation and uncertainty quantification
The refinement of model parameter values, called parameter estimation or calibration, is typically an iterative process where the model is simulated, compared to experimental data, and then updated — repeatedly (Figure 2, box 2b). This requires model and experimental data in a machine readable (preferably standardized) form and software that efficiently searches the parameter space. Parameter estimation is a field under development, and here we describe classical optimization methods as well as some data science methods for uncertainty quantification more novel to this field. More information is given in, for example, Ashyraliyev et al., 2009; Mitra and Hlavacek, 2019; Sun et al., 2012.

In an ideal world, parameters such as rate constants and species concentrations would be directly reported in publications. The reality is that there are many experiments that potentially provide good constraints for parameters, but do not directly report the parameters in the form needed for a model. For example, the rate of appearance of the phospho-form of a protein depends on the concentration of the phosphorylating enzyme, its kinetics, the concentration of the substrate, and possibly other factors. What is most useful for a model is an estimate of the kinetic rate because enzyme and substrate concentration are determined by other reactions. The kinetic rate, and other parameters, could potentially be inferred (indirectly) from experimental data for quantities like the relative activation of the enzyme or turnover of the protein. The overall approach, denoted parameter estimation, is as follows.

- Select the experimental data to be used for the parameter estimation, and formalize a mapping between model output and experimental readouts.
- Describe the prior assumptions made on possible parameter ranges. These assumptions guide your parameter estimation process and could be based on experimental information or physical constraints, for example all parameters are often assumed to be nonnegative.
- Calibrate the model, that is do parameter estimation, either through optimization techniques that provide one good parameter set, or through an approach resulting in a distribution of good parameter values.
- Quantify the uncertainty in your parameter estimates. Ideally by describing the whole parameter space that would correspond to a good fit with the data used for parameter estimation through for example a Bayesian approach, or at least by performing a sensitivity analysis.
- Update the model specification with the new parameter values, and if possible, uncertainty estimates.

In the section ‘Model specification and data’, we describe different formats and practices that can be used to contain all this information, keeping FAIR aspects in mind.

Numerous approaches have been developed for finding parameters that allow a model to match experimental data through optimization (Ashyraliyev et al., 2009; Villaverde et al., 2019). For most models, the optimization method must be able to deal with multiple local minima. A key aspect of optimization is to specify the function that compares simulations with experimental data to find the best set of parameters. This function is often called the objective function, cost function, or energy function and includes a decision on how the different experiments should be weighed against each other. The most common function to use is the sum of the squared differences between simulations and data, weighted by the measurement error (Ashyraliyev et al., 2009). Some existing tools that can be used for parameter estimation are listed in Table 4.
Criticisms are often heard about the large numbers of parameters in modeling studies implying that ‘anything can be fitted’. Implicitly, this means that the model lacks explanatory and predictive power. This is a more common criticism for models that include many details, like mechanistic models built at a fine granularity in a bottom-up manner. However, it is important to note that the mechanistic model structure itself puts a lot of restrictions on the possible model behaviors. Also, since the parameters and variables that are associated with mechanistic models can be mapped to biological entities, these are also restricted through known physiological constraints, increasing the specificity of the model. Thus, it can actually be very hard (or impossible) to fit a specific dataset. If solutions exist, however, it is often so that there are many parameter sets that produce simulation results with a good fit to data, thus explaining the data equally well (Gutenkunst et al., 2007). Different parameter sets may however give different model predictions (Eriksson et al., 2019). This calls for methods that go beyond the use of a single parameter set, accounting for uncertainty in the modeling of neuroscience systems. Such uncertainty can come from sparse experimental data or structural unidentifiability (Raue et al., 2009) or it may reflect actual biological variability (Marder et al., 2015). In key studies,

Table 4. Tools for model refinement and analysis.
This table exemplifies some common and new tools for parameter estimation and different types of model analyses.

| Name       | Purpose                                                                 | Interchange file formats supported | Homepage                                    | RRID                  | Reference                  |
|------------|-------------------------------------------------------------------------|------------------------------------|--------------------------------------------|-----------------------|-----------------------------|
| Ajustador  | Data-driven parameter estimation for Moose and NeuroRD models. Provides parameter distributions. | CSV, MOOSE, NeuroRD               | https://neurord.github.io/ajustador/       |                      | Jedrzejewski-Szmek and Blackwell, 2016 |
| AMICI      | High-level language bindings to CVODE and SBML support.                 | SBML                               | https://amici.readthedocs.io/en/latest/index.html |                      | Fröhlich et al., 2021      |
| BluePyOpt  | Data-driven model parameter optimization.                                | SBML                               | https://github.com/BlueBrain/BluePyOpt     | SCR_014753            | Van Geit et al., 2016      |
| PottersWheel| Parameter estimation, profile likelihood: determination of identifiability and confidence intervals for parameters. | SBML                               | https://potterswheel.de/                   | SCR_021118            | Maiwald and Timmer, 2008, Raue et al., 2009 |
| pyABC      | Parameter estimation through Approximate Bayesian Computation (likelihood free Bayesian approach). | PEtab, SBML via AMICI             | https://pyabc.readthedocs.io/en/latest/    |                      | Klinger et al., 2018       |
| Simbiology | MATLAB’s systems biology toolbox (Mathworks), performs, for example, parameter estimation, local and global sensitivity analysis, and more. | SBML                               | https://se.mathworks.com/products/simbiology.html |                      | Schmidt and Jirstrand, 2006 |
| Uncertainpy| Global Sensitivity Analysis                                             | NEURON and NEST models            | https://github.com/simetenn/uncertainpy    |                      | Tennøe et al., 2018        |
| XPPAUT/AUTO| Model analysis including phase plane analyses, stability analysis, vector fields, null clines, and more. XPPAUT contains a frontend to AUTO for bifurcation analysis. | SBML                               | http://www.math.pitt.edu/~bard/xpp/xpp.html | SCR_001996            | Ermentrout, 2002 (XPPAUT) |
| pyPESTO    | Toolbox for parameter estimation                                         | SBML, PEtab                        | https://github.com/ICB-DCM/pyPESTO/        | SCR_016891            | Stapor et al., 2018        |
| COPASI     | Simulation and analysis of biochemical network models                    | SBML                               | https://copasi.org                         | SCR_014260            | Hoops et al., 2006         |
| PyBioNetFit| Parameterizing biological models                                         | BNGL, SBML, BPSL                    | https://bionetfit.nau.edu/                 |                      | Mitra et al., 2019         |
| Data2Dynamics| Establishing ODE models based on experimental data                      | SBML                               | https://github.com/Data2Dynamics/d2d       |                      | Raue et al., 2015          |
| HippoUnit  | Testing scientific models                                               | HOC language                        | https://github.com/KalLab/hippounit        |                      | Sáray et al., 2021         |
Prinz et al., 2004 investigated the possibility of variability in small neural invertebrate circuits using computational models. With an ensemble model approach, they showed in models of the pyloric rhythm of the crustacean stochastic ganglion that similar oscillatory network activity could arise from widely disparate circuit parameters (Prinz et al., 2004), suggesting that there could be considerable animal-to-animal variability in many of the parameters that control network activity. This has also been confirmed in experiments (Schulz et al., 2006).

Although computationally expensive, Bayesian approaches provide the perfect tools for describing such uncertainty in parameter estimates (Vanlier et al., 2013). These approaches often use Markov Chain Monte Carlo (MCMC) methods to explore the possible parameter space, resulting in a joint distribution describing good parameter values (e.g. Girolami and Calderhead, 2011). However, currently, MCMC methods are only applicable for medium sized models, on the order of dozens of parameters. A less costly approach with a similar goal is profile likelihood (Raue et al., 2009), which provides a possible range for each of the uncertain parameters, but not the joint parameter distribution. Global sensitivity analysis can also be used to investigate uncertainty, assuming that the parameters have specific predefined distributions (Tennae et al., 2018).

Parameter space exploration is a process that could benefit from additional automation. For example, many conventional plasticity models are 'manually tuned'. Nonetheless, parameter optimization has been combined into different workflows, often using custom scripts to interface with an optimization tool. FindSim (Viswan et al., 2018) is a framework for integrating many individual electrophysiological and biochemical experiments with large, multiscale models so that model output can be compared with experiments in an optimization workflow. Santos et al., 2021 describe an automatic workflow for parameter estimation using an SBtab file as input, rendering itself to automatic parameter estimation through many of the optimization tools that exist within Matlab, and with a final step of global sensitivity analysis. Ajustador (Jedrzejewski-Szmek et al., 2018) is an optimization algorithm for models specified in declarative format for either MOOSE or NeuroRD that allows specification of the weights of various data features. SciUnit (Omar et al., 2014) provides a Python-based framework for creating standardized tests that compare model outputs to experimental data. NeuroUnit provides a domain-specific repository of tests for use within the SciUnit framework that neuron models must pass to demonstrate validity when modeling at the detailed cellular level. These tests can also be used as the basis for model optimization (Gerkin et al., 2018). More parameter estimation tools are listed in Table 4.

To be able to reproduce the model refinement process, it is important that the model, experimental data, and prior parameter assumptions are well described and explicitly specified, as described above. If this is the case, then ideally different parameter estimation tools should come to similar conclusions, at least if a methodology including a distribution of possible parameter values is used.

### BCM example

Parameter estimation is not confined to data-driven, mechanistic models. Even abstract models reflect the observables that they set out to explain. Thus, one can ‘tune’ parameters in abstract models to data using the same techniques discussed above. The authors of the original BCM model (Bienenstock et al., 1982) did this to obtain ocular dominance properties in their simulations. Another early model of synaptic plasticity (Lisman, 1989) explicitly proposed a set of pathways that could result in bidirectional plasticity equivalent to the BCM curve and implemented chemical reactions with rates that were tuned to obtain properties consistent with the BCM model. Today more efficient parameter estimation methods are used, and a next step could be to quantify the uncertainty in the parameter estimates and predictions through a Bayesian methodology, if possible, or perform a global sensitivity analysis, to understand the effects of different parameters on the model behavior.

### Model validation and usage

Following the model refinement process, we obtain the final version of the model that can be validated, analyzed, and used in simulation studies to make predictions and test hypotheses (Figure 2, Box 3). In multiscale modeling the output from the model can also be used as the input for another scale. Ideally, before the model is used for predictions, it is validated against data that were not used in the model construction and refinement process. However, data may be sparse requiring that validation occur through experimental tests of model predictions. After the model is finalized, the
predictions from the model can be extended and better informed through model analysis. This could be done by sensitivity analysis, with tools from dynamical systems theory or through experimental design. Some examples of tools for model analysis are listed in Table 4.

Sensitivity analysis is a methodology that investigates how different input factors, such as model parameters or model input, affect the model output (Saltelli et al., 2007; Zi, 2011). The output could be the concentration of the active form of a protein as an example. Sensitivity analysis can be performed through local methods, which considers the partial derivatives in the close neighborhood of a point in parameter space, or global methods, which investigate a much larger region of the parameter space, often with statistical methods. Local methods are computationally efficient, but can be problematic due to limited range. Global sensitivity analysis on the other hand investigates a larger region, but is computationally costly. One example where a global sensitivity analysis was used is the study by Halnes et al., 2009. There are, however, also coarse-grained global sensitivity analysis methods as well as novel methods that interface coarse-grained global with local sensitivity analyses methods. Several studies have also investigated the sensitivity by a so-called one-at-a-time approach, where parameters are perturbed one at a time, for example Gutierrez-Arenas et al., 2014. Finally, a Bayesian approach for uncertainty quantification can also be combined with global sensitivity analysis (Eriksson et al., 2019), as illustrated in Figure 4. Tennøe et al. have recently developed a software for global sensitivity analysis (Tennøe et al., 2018).

In an experimental design process, models can be used to simulate a large number of different scenarios corresponding to different potential experimental setups and in this way be used to design experiments that provide the most amount of new information (Kreutz and Timmer, 2009). Statistical
methods have been developed for this, often using Bayesian methodology, see for example Liepe et al., 2013.

Another way of analyzing dynamical models is to use tools from dynamical systems theory. Such methods typically examine asymptotic stability, provide phase diagrams, and identify bifurcations in parameter space that identify and explain how major changes in model behavior rely on parameters or input. This approach has, as an example, been used to better understand systems level mechanisms (e.g., how a cell goes from silent to spiking, or how oscillatory phenomena in networks might be controlled [Ermentrout et al., 2019; Keeley et al., 2017]). With regard to subcellular-level phenomena important for plasticity, intracellular calcium release (Li and Rinzel, 1994) as well as switch-like behaviors have been investigated, for example CaMKII and protein kinase M – zeta (PKMζ) (Graupner and Brunel, 2007; Helfer and Shultz, 2018; Pi and Lisman, 2008). For instance, stability analysis shows that CaMKII can be at two stable states at resting intracellular calcium concentration (Graupner and Brunel, 2007), and the system of coupled CaMKII and protein phosphatase A (PP2A) could behave as a tristable system that describes LTP and LTD (Pi and Lisman, 2008). The original BCM study also included analysis with tools from dynamical systems theory to test whether the plasticity rule was stable.

Toward FAIR modeling workflows

There are many options for assembling the resources and approaches described to create specific modeling workflows that follow the steps outlined in Figure 2. A flexible workflow going through all the steps requires:

1. Much effort toward identifying the scientific questions to be addressed and the model prerequisites (see ‘Model foundation’).
2. A well-defined format for the model, the quantitative experimental data, and the assumptions on prior parameter ranges or distributions (see ‘Specifications of model and data’).
3. A simulator that supports a machine readable, standardized specification of the model (see ‘Mathematical formulation and simulation’).
4. A tool for searching the parameter space, that compares the simulations with the experimental data guided by the parameter prior information (see ‘Model refinement’).
5. Tools for model analysis (see ‘Model usage and validation’).
6. Parsing tools to go between the specification of model and data format and internal formats for the model refinement and model analysis tools.

In Figure 5, we provide three specific example workflows for subcellular- and cellular-level modeling that traverse these steps in different ways. These examples illustrate recent steps toward FAIR workflows and provide a starting point for further discussion within the community.

Aside from single steps and components, we will now consider some questions that are important for all parts of the workflow, and in relation to this we also recommend the article by Goble et al., 2020, which describes FAIR computational workflows from a more general perspective. The first question concerns findability and here we emphasize the importance of metadata and providing unique and stable identifiers to the workflow components. Concerning models and data, some databases provide a unique Digital Object Identifier (DOI) for its constituents. This can be retrieved when the model or data are deposited in the database and included in the metadata of the workflow. Each resource and software used in the workflow should also be indicated using the associated, unique SciCrunch Research Resource Identifier (RRID) (Vasilevsky et al., 2013; Bandrowski and Martone, 2016), if it exists. It can be noted that in order to get an identifier to a more complex research object a general database like Zenodo (European Organization For Nuclear Research, 2013) could be used. As an example it is possible to use Zenodo to get a DOI for a GitHub repository release.

The second question concerns approaches for the packaging, description, and dissemination of the entire model building workflow, which is important for transparency and reproducibility. FAIRDOMhub (Wolstencroft et al., 2017) is a repository and collaboration environment for publishing FAIR data, operating procedures, and models for the systems biology community. EBRAINS has a similar ambition to provide a collaboration hub for brain research. Virtual machines or containers (Merkel, 2014) can play a role in reproducing all steps and results of a workflow. Easily accessible technologies such as Jupyter notebooks (Kluyver et al., 2016) can aid in interoperability, dissemination, and collaboration.
Figure 5. Toward FAIR (Findable, Accessible, Interoperable, Reusable) workflows in neuroscience. Three examples of workflows at different biological scales. Several of the workflow components can be found in the tables of the article as indicated by T (Table) and R (Row) in the figure, for example T1,R4. (a) To build a single neuron multicompartmental electrical model, morphologies from databases such as NeuroMorpho or Allen Brain Atlas (http://celltypes.brain-map.org/) can be used (Ascoli et al., 2007; Gouwens et al., 2020). Electrical recordings from Allen Brain Atlas are also available. Figure 5 continued on next page.
Features from experimental traces can be extracted using eFEL (https://github.com/BlueBrain/eFEL), and kinetic parameters for ion channel models obtained from Channelpedia. During the neuron model reconstruction step, the model is represented in NeuroML and model parameters such as conductance density are optimized with BluePyOpt. In silico experiments can be performed using NEURON. An example workflow for building a chemical kinetic model to implement the Bienenstock–Cooper–Munro (BCM) curve. The expected shape of the curve is obtained from the classic Bienenstock et al., 1982 study, and a first pass of likely chemical pathways from Lisman, 1989. Detailed chemistry and parameters are from databases that cover models (DOQCS, BioModels) and from BRENDA, which hosts enzyme kinetics. Both model databases support SBML, which can be used to define the model and parameters. Several simulators including MOOSE and COPASI can run the SBML model during the optimization step and subsequently for model predictions. For optimization, the FindSim framework (Viswan et al., 2018) compares model outcome to experiments. The score from these comparisons is used by HOSS (Hierarchical Optimization of Systems Simulations https://github.com/upibhalla/HOSS) to carry out parameter fitting. A prototype of the workflow was used in Church et al., 2021, where the experimental data is described. The model structure was adopted from Buxbaum and Daudai, 1989. A smaller demo version can be found at https://github.com/cpm-kth/uqsa (copy archived at https://doi.org/10.5281/zenodo.6625529). Experimental data and model structure are taken from literature and saved in the SBtab format. Scripts written in R convert the SBtab file to R code and the Bayesian parameter estimation is performed with the UQSA software written in R (https://github.com/cpm-kth/uqsa, copy archived at https://doi.org/10.5281/zenodo.6625529). The SBtab files can subsequently be updated with the refined model with new parameter estimates including uncertainties.

The Common Workflow Language allows users to fully specify inputs, outputs, and other execution details needed for tools in a workflow and instructions for executing them within containers (Piccolo et al., 2021). The workflow around a model can also be made public analogous to publishing the protocol of a lab experiment, for example via protocols.io (Teitelman and Stoliartchouk, 2015). The recently published Research Object Crate (RO-Crate) provides a way to package and aggregate research ‘artifacts’ with their metadata and relationships (Soiland-Reyes et al., 2022), like spreadsheets, code, examples, and figures.

Finally, we illustrate a FAIR workflow by considering how one would today develop simulations for the BCM curve in Figure 3. Even though the original BCM model and many subsequent experimental studies predate the advent of FAIR principles, it is interesting to think about what would need to be done in order to make them FAIR compliant. Ideally, these studies would be Findable through PubMed and other resources. This implies that both metadata and data are easy to locate for both humans and computer programs. We would probably rely more heavily on later models, for example, Hayer and Bhalla, 2005, that are easily Accessible and in an Interoperable format such as SBML or SBtab. The Hayer model is available via the DOQCS database (Sivakumaran et al., 2003). In order to be Reusable, models should be replicable and compatible in different contexts. Quantitative experimental data could today be specified using FindSim or SBtab formats. Both formats support the comparison of model behavior with the experimental data, which is necessary for model refinement. For the model refinement to be reproducible it is important that the prior assumptions on parameters are described. It should be noted that the provenance for the Hayer model is not machine readable, though ideally it would be. Furthermore, more comprehensive metadata and appropriate license information should be included. Finally, since the refined model is defined in SBML, it could be used on any of several dozen simulator platforms (Table 3), and uploaded back to the Accessible databases to become part of the FAIR ecosystem.

Conclusions and outlook
We have investigated how data- and hypothesis-driven modeling approaches can be combined, through a FAIR infrastructure, in order to improve modeling capabilities within the neuroscience community. We suggest a minimal format for the ‘specification of model and data’ (Figure 2, box 2a) in order for the modeling process to be reproducible. We also describe different databases, formats, and software (Tables 1–4) and methods for combining them to achieve FAIR modeling workflows. We believe that such workflows would increase the capacity for combining different types of models, extending models as new data accumulates, and validating existing models.

When it comes to model refinement (Figure 2, box 2b), we have emphasized the importance of uncertainty quantification with for example Bayesian methods (Figure 4) in order to describe the uncertainty in the parameter estimates and model predictions. However, currently such methods are only possible (in practice) for models of the order of some dozens of parameters, pointing to a need for further method development. Concerning approaches for model analysis (Figure 2, boxes 2b...
and 3) an important aspect is to characterize models more objectively, thus avoiding the situation that the model has been ‘tuned’ to give a certain outcome. This is especially important as the goal of a model often is to reproduce specific observed behaviors in order to validate a hypothesis, rather than being predictive. This is particularly true for abstract, phenomenological models; however, data-driven mechanistic models that are portrayed as predictive often play the same, useful postdictive role. Many studies of such models now conduct parameter sensitivity analysis and parameter explorations to discover regimes in which new properties emerge, thereby providing an unbiased overview of the possible properties of the model.

In the future, it is likely that machine learning tools increasingly will be integrated with traditional modeling approaches as a complement but also a necessity for dealing with high throughput data of varying fidelity (Alber et al., 2019). While machine learning approaches are useful for finding correlations in big data, traditional mechanistic models are important for revealing causal relationships and mechanistic explanations. Mechanistic models also have the added possibility to generalize to new situations not considered during the model construction process.

If we want to understand the brain — from molecules to behavior — we need to model the brain at all biological scales to integrate and interpret various heterogeneous experimental data across scales. Here we have examined the opportunities associated with models arising at the scale where subcellular signaling processes meet the cellular level, corresponding to the field where the systems biology community meets the computational neuroscience community. Although the development of resources that integrate the FAIR principles into the computational neuroscience field has progressed over the last decade (e.g., compare De Schutter, 2008), standards and best practices still need to be further developed and aligned, an important endeavor coordinated by INCF (Abrams et al., 2021). The adoption of FAIR principles is timely due to the recent debut of brain initiatives around the world (Grillner et al., 2016), including the establishment of the International Brain Initiative (International Brain Initiative, 2020). Such efforts will help promote data and model exchange and reuse through international research and infrastructure collaborations.

Acknowledgements
We would like to thank Federica Milinanni, Alexandra Jauhiainen and Matthew Gold for help with the workflow of Figure 5C and Noora Pihlajarinne for help with the illustrations.

Additional information

Competing interests
Upinder Singh Bhalla: Reviewing Editor, eLife. The other authors declare that no competing interests exist.

Funding

| Funder                          | Grant reference number                              | Author                                           |
|--------------------------------|-----------------------------------------------------|--------------------------------------------------|
| Horizon 2020 Framework Programme | 945539 (The Human Brain Project)                    | Rebecca C Wade                                   |
|                                 |                                                     | Jeanette Hellgren Kotalski                       |
|                                 |                                                     | Marja-Leena Linne                                |
|                                 |                                                     | Daniel Keller                                    |
| Swedish Research Council        | VR-M-2017-02806                                      | Jeanette Hellgren Kotalski                       |
| Swedish Research Council        | VR-M-2020-01652                                      | Jeanette Hellgren Kotalski                       |
| Swedish e-Science Research Centre| Governmental grant for strategic research areas     | Olivia Eriksson                                  |
|                                 |                                                     | Andrei Kramer                                    |
|                                 |                                                     | Jeanette Hellgren Kotalski                       |
**Author contributions**

Olivia Eriksson, Jeanette Hellgren Kotaleski, Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Writing – original draft, Writing – review and editing; Upinder Singh Bhalla, Kim T Blackwell, Sharon M Crook, Daniel Keller, Andrei Kramer, Marja-Leena Linne, Ausra Saudargienë, Rebecca C Wade, Conceptualization, Funding acquisition, Investigation, Methodology, Writing – original draft, Writing – review and editing

**Author ORCIDs**

Olivia Eriksson [http://orcid.org/0000-0003-0740-4318](http://orcid.org/0000-0003-0740-4318)

Upinder Singh Bhalla [http://orcid.org/0000-0003-1722-5188](http://orcid.org/0000-0003-1722-5188)

Kim T Blackwell [http://orcid.org/0000-0003-4711-2344](http://orcid.org/0000-0003-4711-2344)

Sharon M Crook [http://orcid.org/0000-0003-1659-0749](http://orcid.org/0000-0003-1659-0749)

Daniel Keller [http://orcid.org/0000-0003-3280-6255](http://orcid.org/0000-0003-3280-6255)

Andrei Kramer [http://orcid.org/0000-0002-3828-6978](http://orcid.org/0000-0002-3828-6978)

Marja-Leena Linne [http://orcid.org/0000-0003-2577-7329](http://orcid.org/0000-0003-2577-7329)

Ausra Saudargienë [http://orcid.org/0000-0003-2768-3334](http://orcid.org/0000-0003-2768-3334)

Rebecca C Wade [http://orcid.org/0000-0001-5951-8670](http://orcid.org/0000-0001-5951-8670)

Jeanette Hellgren Kotaleski [http://orcid.org/0000-0002-0550-0739](http://orcid.org/0000-0002-0550-0739)

**References**

Abrams MB, Bjalie JG, Das S, Egan GF, Ghosh SS, Gosciniski WJ, Grethe JS, Kotaleski JH, Ho ETW, Kennedy DN, Lanyon LJ, Leergaard TB, Mayberg HS, Milanesi L, Mouček R, Poline JB, Roy PK, Strother SC, Tang TB, Tiesinga P, et al. 2021. A Standards Organization for Open and FAIR Neuroscience: the International Neuroinformatics Coordinating Facility. *Neuroinformatics* 1:e09509-0. DOI: https://doi.org/10.1007/s12021-020-09509-0, PMID: 33506383

Alber M, Buganza Tepole A, Cannon WR, De S, Dura-Bernal S, Garikipati K, Karniadakis G, Lytton WW, Perdikaris P, Petzold L, Kuhl E. 2019. Integrating machine learning and multiscale modeling-perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digital Medicine* 2:115. DOI: https://doi.org/10.1038/s41746-019-0193-y, PMID: 31799423
exchange of biochemical network models. Bioinformatics (Oxford, England) 19:524–531. DOI: https://doi.org/10.1093/bioinformatics/btg015, PMID: 12611808

Hucka M, Nickerson DP, Bader GD, Bergmann FT, Cooper J, Demir E, Garry N, Golebiewski M, Myers CJ, Schreiber F, Waltemath D, Le Novère N. 2015. Promoting Coordinated Development of Community-Based Information Standards for Modeling in Biology; The COMBINE Initiative. Frontiers in Bioengineering and Biotechnology 3:19. DOI: https://doi.org/10.3389/fbioe.2015.00019, PMID: 25759811

International Brain Initiative. 2020. International Brain Initiative: An Innovative Framework for Coordinated Global Brain Research Efforts. Neuron 105:212–216. DOI: https://doi.org/10.1016/j.neuron.2020.01.002, PMID: 31972144

Jedlicka P, Benusova L, Abraham WC. 2015. A Voltage-Based STDP Rule Combined with Fast BCM-Like Metaplasticity Accounts for LTP and Concurrent “Heterosynaptic” LTD in the Dentate Gyrus In Vivo. PLOS Computational Biology 11:e1004558. DOI: https://doi.org/10.1371/journal.pcbi.1004558, PMID: 26544038

Jedrzejewski-Szmek Z, Blackwell KT. 2016. Asynchronous τ-leaping. The Journal of Chemical Physics 144:125104. DOI: https://doi.org/10.1063/1.4944575, PMID: 27036481

Jedrzejewski-Szmek Z, Abrahao KP, Jedrzejewska-Szmek J, Lovinger DM, Blackwell KT. 2018. Parameter Optimization Using Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES), an Approach to Investigate Differences in Channel Properties Between Neuron Subtypes. Frontiers in Neuroinformatics 12:47. DOI: https://doi.org/10.3389/fninf.2018.00047, PMID: 30108495

Kanehisa M, Goto S. 2000. KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Research 28:27–30. DOI: https://doi.org/10.1093/nar.28.1.27, PMID: 10592173

Keeley S, Fenton AA, Rinzel J. 2017. Modeling fast and slow gamma oscillations with interneurons of different subtype. Journal of Neurophysiology 117:950–965. DOI: https://doi.org/10.1152/jn.00490.2016, PMID: 27922782

Kerr RA, Bartol TM, Kaminsky B, Dittrich M, Chang JCJ, Baden SB, Sejnowski TJ, Stiles JR. 2008. Fast monte carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. SIAM Journal on Scientific Computing 30:3126. DOI: https://doi.org/10.1137/070692017, PMID: 20151023

Kirkwood A, Rioult MC, Bear MF. 1996. Experience-dependent modification of synaptic plasticity in visual cortex. Nature 381:526–528. DOI: https://doi.org/10.1038/381526a0, PMID: 8632826

Klinger E, Rickert D, Hasenauer J. 2018. pyABC: distributed, likelihood-free inference. The FEBS Journal 1038/nature05453, PMID: 17151600

Knight BW. 1972. Dynamics of encoding in a population of neurons. The Journal of General Physiology 59:734–766. DOI: https://doi.org/10.1085/jgp.59.6.734, PMID: 29762723

Kohl P, Noble D, Hunter PJ, Hedley WJ, Nelson MR, Bellivant DP, Nielsen PF. 2001. A short introduction to systems biology. Philosophical Transactions of the Royal Society of London. Series A 359:1073–1089. DOI: https://doi.org/10.1098/rsta.2001.0817

Kirkwood A, Fenton AA, Rinzel J. 2017. Modeling fast and slow gamma oscillations with interneurons of different subtype. Journal of Neurophysiology 117:950–965. DOI: https://doi.org/10.1152/jn.00490.2016, PMID: 27922782

Klipp E, Liebermeister W, Wiering C, Kowald A. 2016. Systems Biology: A Textbook. John Wiley & Sons.

Knights T, Ragan-Kelley B, Pe RF, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C, Team JD. 2016. Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows, Positioning and Power in Academic Publishing: Players, Agents and Agendas. IOS Press. DOI: https://doi.org/10.3233/978-1-6499-649-1-87

Koehler T, Liepert J, Hess B, Schadle A, Huttel M, Wagner T, Ribo B, Brandt P, Müller-Steinhagen H. 2003. EEG analysis of epileptiform activity in patients with temporal lobe epilepsy using wavelet transforms. Neuroimage 4:211–221. DOI: https://doi.org/10.1006/nimg.2002.1220, PMID: 12974494

Koh P, Noble D, Hunter PJ, Hedley WJ, Nelson MR, Bellivant DP, Nielsen PF. 2001. A short introduction to CellML. Philosophical Transactions of the Royal Society of London. Series A 359:1073–1089. DOI: https://doi.org/10.1098/rsta.2001.0817

Kreutz C, Timmer J. 2009. Systems biology: experimental design. The FEBS Journal 276:923–942. DOI: https://doi.org/10.1111/j.1742-4658.2008.06843.x

Lein ES, Hawrylycz MJ, Ao N, Ayres M, Bensinger A, Bernard A, Boe AF, Boguski MS, Brockway KS, Byrnes EJ, Chen L, Chen L, Chen T-M, Chin MC, Chong J, Crook BE, Czaplinska A, Dang CN, Datta S, Dee NR, et al. 2007. Genome-wide atlas of gene expression in the adult mouse brain. Nature 445:168–176. DOI: https://doi.org/10.1038/nature05453, PMID: 17151600

Li YX, Hawrylycz MJ, Ao N, Ayres M, Bensinger A, Bernard A, Boe AF, Boguski MS, Brockway KS, Byrnes EJ, Chen L, Chen L, Chen T-M, Chin MC, Chong J, Crook BE, Czaplinska A, Dang CN, Datta S, Dee NR, et al. 2007. Genome-wide atlas of gene expression in the adult mouse brain. Nature 445:168–176. DOI: https://doi.org/10.1038/nature05453, PMID: 17151600

Li YY, Rinzel J. 1994. Equations for InsP3 receptor-mediated [Ca2+]i oscillations derived from a detailed kinetic model: a Hodgkin-Huxley like formalism. Journal of Theoretical Biology 166:461–473. DOI: https://doi.org/10.1006/jtbi.1994.1041, PMID: 8176949

Li C, Donizelli M, Rodriguez N, Dhuriri H, Endler L, Chelliah V, Li L, He E, Henry A, Stefan MI, Snoeij JL, Hucka M, Le Novère N, Laibe C. 2010. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology 4:92. DOI: https://doi.org/10.1186/1752-0509-4-92, PMID: 20587024

Li L, Stefan MI, Le Novère N. 2012. Calcium input frequency, duration and amplitude differentially modulate the relative activation of calcineurin and CaMKII. PLOS ONE 7:e43810. DOI: https://doi.org/10.1371/journal.pone.0043810, PMID: 22962589

Liepe J, Filippi S, Komorowski M, Stumpf MPH. 2013. Maximizing the information content of experiments in systems biology. PLOS Computational Biology 9:e1002888. DOI: https://doi.org/10.1371/journal.pcbi.1002888, PMID: 23382663

Lisman J. 1989. A mechanism for the Hebb and the anti-Hebb processes underlying learning and memory. PNAS 86:9574–9578. DOI: https://doi.org/10.1073/pnas.86.23.9574, PMID: 2556718

Lisman J. 2017. Glutamatergic synapses are structurally and biochemically complex because of multiple plasticity processes: long-term potentiation, long-term depression, short-term potentiation and scaling. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 372:20160260. DOI: https://doi.org/10.1098/rstb.2016.0260, PMID: 28093558
2011. NineML: the network interchange for neurosciences modeling language. BMC Neuroscience 12:P330. DOI: https://doi.org/10.1186/1471-2202-12-S1-P330

Ranjan R, Khazen G, Gambazzi L, Ramaswamy S, Hill SL, Schürmann F, Markram H. 2011. Channelpedia: an integrative and interactive database for ion channels. Frontiers in Neuroinformatics 5:36. DOI: https://doi.org/10.3389/fninf.2011.00036, PMID: 22232598

Raue A, Kreutz C, Maiwald T, Bachmann J, Schilling M, Klingmüller U, Timmer J. 2009. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. Bioinformatics (Oxford, England) 25:1923–1929. DOI: https://doi.org/10.1093/bioinformatics/btp358, PMID: 19505944

Raue A, Steiert B, Schauer M, Kreutz C, Maiwald T, Hass H, Vanlier J, Tönsing C, Adlung L, Engesser R, Mader W, Heinemann T, Hasenauer J, Schilling M, Höfer T, Klipp E, Theis F, Klingmüller U, Schöberl B, Timmer J. 2015. Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. Bioinformatics (Oxford, England) 31:3558–3560. DOI: https://doi.org/10.1093/bioinformatics/btv405, PMID: 26142188

Ray S, Bhalla US. 2008. PyMOOSE: Interoperable Scripting in Python for MOOSE. Frontiers in Neuroinformatics 2:6. DOI: https://doi.org/10.3389/neuro.11.006.2008, PMID: 19129924

Ray S, Deshpande R, Dudani N, Bhalla US. 2008. A general biological simulator: the multiscale object oriented simulation environment, MOOSE. BMC Neuroscience 9:P93. DOI: https://doi.org/10.1186/1471-2202-9-S1-P93

Rübel O, Tritt A, Dichter B, Braun T, Cain N, Clack N, Davidson TJ, Dougherty M, Fillion-Robin JC, Graddis N, Rübel O, Deshpande R, Dudani N, Bhalla US. 2008. A general biological simulator: the multiscale object oriented simulation environment, MOOSE. BMC Neuroscience 9:P93. DOI: https://doi.org/10.1186/1471-2202-9-S1-P93

Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S. 2007. Global Sensitivity Analysis: The Primer. Chichester, UK: John Wiley. DOI: https://doi.org/10.1002/9780470725184

Schaff J, Fink CC, Slepchenko B, Carson JH, Loew LM. 1997. A general computational framework for modeling cellular structure and function. Biophysical Journal 73:1135–1146. DOI: https://doi.org/10.1016/S0006-3495(97)78146-3, PMID: 9284281

Schmidt H, Jirstrand M. 2006. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. Bioinformatics (Oxford, England) 22:514–515. DOI: https://doi.org/10.1093/bioinformatics/bti799, PMID: 16317076

Schmiester L, Schätzle Y, Bergmann FT, Camba T, Dudkin E, Egert J, Fröhlich F, Fink CC, Slepchenko B, Carson JH, Loew LM. 1997. A general computational framework for modeling cellular structure and function. Biophysical Journal 73:1135–1146. DOI: https://doi.org/10.1016/S0006-3495(97)78146-3, PMID: 9284281

Schmitzer P, Schütte C, Grüber M, Griebel M, Dürre P, Schäffer A, Schütte KP, Dürre P, Schröder TM, Jahnke K, Hänggi P, Griebel M, Dürre P, Schröder TM, Jahnke K, Hänggi P. 2011. Metamodelling and global sensitivity analysis: the Primer. Chichester, UK: John Wiley. DOI: https://doi.org/10.1002/9781118098099

Schulz DJ, Goaillard JM, Marder E. 2006. Variable channel expression in identified single and electrically coupled neurons in different animals. Nature Neuroscience 9:356–362. DOI: https://doi.org/10.1038/nn1639, PMID: 16444270

Sivakumaran S, Harinarayanan S, Mishra J, Bhalla US. 2003. The Database of Quantitative Cellular Signaling: management and analysis of chemical kinetic models of signaling networks. Bioinformatics (Oxford, England) 19:408–415. DOI: https://doi.org/10.1093/bioinformatics/btf860, PMID: 12584128

Soiland-Reyes S, Stelfon P, Crosas M, Castro LJ, Fernández JM, Gailland E, Grauer M, Kiggins JT, Liu N, Ozturk D, Schroeder W, Soltesz I, Sommer FT, Svoboda K, Lydia N, Frank LM, et al. 2019. NWB:N 2.0: An Accessible Data Standard for Neurophysiology. bioRxiv. DOI: https://doi.org/10.1101/523035

Sáray S, Rössert CA, Appukuttan S, Migliore R, Vitale P, Lupascu CA, Bologna LL, Geit WV, Romani A, Sivakumaran S, Hariharaputan S, Mishra J, Bhalla US. 2003. The Database of Quantitative Cellular Signaling: management and analysis of chemical kinetic models of signaling networks. Bioinformatics (Oxford, England) 19:408–415. DOI: https://doi.org/10.1093/bioinformatics/btf860, PMID: 12584128

Soiland-Reyes S, Stelfon P, Crosas M, Castro LJ, Coppen F, Fernández JM, Garijo D, Grünig B, La Rosa M, Leo S, Ö Carragān E, Portier M, Trisovic A, Groth P, Goble C, Peroni S, RO- Crate Community. 2021. Packaging research artefacts with RO-Crate. Data Science 1:1–42. DOI: https://doi.org/10.3233/DS-210053

Stapor P, Weindl D, Ballnus B, Hug S, Loos C, Fiedler A, Krause S, Hrös S, Fröhlich F, Hasenauer J. 2018. PESTO: Parameter ESTimation TOolbox. Bioinformatics (Oxford, England) 34:705–707. DOI: https://doi.org/10.1093/bioinformatics/btt676, PMID: 29069312

Stein M, Gabdoulline RR, Wade RC. 2007. Bridging from molecular simulation to biochemical networks. Current Opinion in Structural Biology 17:166–172. DOI: https://doi.org/10.1016/j.sbi.2007.03.014, PMID: 17395455

Stiles JR, Van Helden D, Bartol TM, Salpeter EE, Salpeter MM. 1996. Miniature endplate current rise times less than 100 microseconds from improved dual recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle. PNAS 93:5747–5752. DOI: https://doi.org/10.1073/pnas.93.12.5747, PMID: 8650164

Stiles JR, Bartol TM. 2001. Monte Carlo methods for simulating realistic synaptic microphysiology using MCell. De Schutter E (Ed). Computational Neuroscience: Realistic Modeling for Experimentalists. Boca Raton: CRC Press. p. 87–127. DOI: https://doi.org/10.1201/9781420039290

Stimberg M, Brette R, Goodman DF. 2019. Brian 2, an intuitive and efficient neural simulator. eLife 8:e47314. DOI: https://doi.org/10.7554/eLife.47314, PMID: 31429824
Stoewer A, Kellner C, Jan B, Thomas W, Jan G. 2014. File format and library for neuroscience data and metadata. Frontiers in Neuroinformatics 8:e00027. DOI: https://doi.org/10.3389/conf.fninf.2014.00027

Sun J, Garibaldi JM, Hodgman C. 2012. Parameter estimation using meta-heuristics in systems biology: a comprehensive review. IEEE/ACM Transactions on Computational Biology and Bioinformatics 9:185–202. DOI: https://doi.org/10.1109/TCBB.2011.63, PMID: 21464505

Sunkin SM, Ng L, Lau C, Dolbeare T, Gilbert TL, Thompson CL, Hawrylycz M, Dang C. 2013. Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system. Nucleic Acids Research 41:D996–D1008. DOI: https://doi.org/10.1093/nar/gks1042, PMID: 23193282

Teeters JL, Harris KD, Millikan KJ, Olshausen BA, Sommer FT. 2008. Data sharing for computational neuroscience. Neuroinformatics 6:47–55. DOI: https://doi.org/10.1007/s12021-008-9009-y, PMID: 18259695

Teeters JL, Godfrey K, Young R, Dang C, Friedsam C, Wark B, Asari H, Peron S, Li N, Peyrache A, Denisov G, Siegel JH, Olsen SR, Martin C, Chun M, Tripathy S, Blanche TJ, Harris K, Buzsáki G, Koch C, et al. 2015. Neurodata Without Borders: Creating a Common Data Format for Neurophysiology. Neuron 88:629–634. DOI: https://doi.org/10.1016/j.neuron.2015.02.015, PMID: 26590340

Tenee S, Halnes G, Einevoll GT. 2018. Uncertainty: A Python Toolbox for Uncertainty Quantification and Sensitivity Analysis in Computational Neuroscience. Frontiers in Neuroinformatics 12:49. DOI: https://doi.org/10.3389/conf.fninf.2018.00049, PMID: 30154710

Teytelman L, Stoliarchouk A. 2015. Protocols.io: Reducing the knowledge that perishes because we do not publish it. Information Services & Use 35:109–115. DOI: https://doi.org/10.3233/ISU-150769

Tiwari K, Kanananath S, Roberts MG, Meyer JP, Sharif Shohan MU, Xavier A, Maire M, Zyoud A, Men J, Ng S, Nguyen TVN, Glont M, Homburg B, Mallik-Shiff Rob, S. 2021. Reproducibility in systems biology modelling. Molecular Systems Biology 17:e9982. DOI: https://doi.org/10.15252/msb.20209982, PMID: 33620773

Traub RD, Jefferys JG, Miles R, Whittington MA, Tóth K. 1994. A branching dendritic model of A rodent CA3 pyramidal neurone. The Journal of Physiology 481 (Pt 1):79–95. DOI: https://doi.org/10.1113/jphysiol.1994.sp02040, PMID: 7853251

Tripathy SJ, Savitskaya J, Burton SD, Urban NN, Gerkin RC. 2014. NeuroElectro: a window to the world’s neuron electrophysiology data frontiers. In Neuroinformatics 8:40. DOI: https://doi.org/10.3389/fninf.2014.00040, PMID: 24808858

UniProt Consortium. 2021. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Research 49:D480–D489. DOI: https://doi.org/10.1093/nar/gkaa110, PMID: 33237286

Van Geit W, Gevaert M, Chindemi G, Rössert C, Courcol JD, Muller EB, Schürmann F, Segev I, Markram H. 2016. BluePyOpt: Leveraging Open Source Software and Cloud Infrastructure to Optimise Model Parameters in Neuroscience. Frontiers in Neuroinformatics 10:17. DOI: https://doi.org/10.3389/conf.fninf.2016.00017, PMID: 27375471

van Keulen SC, Martin J, Colizzi F, Frezza E, Trpevski D, Diaz NC, Vidossich P, Rothlisberger U, Hellgren Kotaleski J, Wade RC, Calloni P. 2022. Multiscale molecular simulations to investigate adenylyl cyclase-based signaling in the brain. WIREs Computational Molecular Science. DOI: https://doi.org/10.1002/wcms.1623

Vanlier J, Tiemann CA, Hilbers PAJ, van Riel NAW. 2013. Parameter uncertainty in biochemical models described by ordinary differential equations. Mathematical Biosciences 246:305–314. DOI: https://doi.org/10.1016/j.mbs.2013.03.006, PMID: 23535194

Vasilevsky NA, Brush MH, Paddock H, Ponting L, Tripathy SJ, Larocca GM, Haendel MA. 2013. On the reproducibility of science: unique identification of research resources in the biomedical literature. PeerJ 1:e148. DOI: https://doi.org/10.7717/peerj.148, PMID: 24032093

Villaverde AF, Fröhlich F, Weindl D, Hasenauer J, Banga JR. 2019. Benchmarking optimization methods for parameter estimation in large kinetic models. Bioinformatics (Oxford, England) 35:830–838. DOI: https://doi.org/10.1093/bioinformatics/bty736, PMID: 30816929

Viswan NA, HarshaRani GV, Stefan MI, Bhalla US. 2018. FindSim: A Framework for Integrating Neuronal Data and Signaling Models. Frontiers in Neuroinformatics 12:38. DOI: https://doi.org/10.3389/conf.fninf.2018.00038, PMID: 29997492

Waltemath D, Adams R, Beard DA, Bergmann FT, Bhalla US, Britten R, Chelliah V, Cooling MT, Cooper J, Crampin EJ, Gurney K, shape S, Hoops S, Hucka M, Hunter P, Klipp E, Laibe C, Miller AK, Moraru I, Nickerson D, Nielsen P, et al. 2011. Minimum Information About a Simulation Experiment (MIASE. 12:49. DOI: https://doi.org/10.3389/conf.fninf.2014.00027, PMID: 18463116

Wang B, Xie ZR, Chen J, Wu Y. 2018. Integrating Structural Information to Study the Dynamics of Protein-Protein Interactions in Cells. Structure (London, England) 26:1414–1424. DOI: https://doi.org/10.1016/j.str.2018.07.010, PMID: 30174150

Weckesser W. 2008. VFGEN: A Code Generation Tool. 15. VFGENs. http://jeriah.org/index.php?/archives/64-VFGEN-A-Code-Generation-Tool.html

Wegscheider R. 1902. Über simultane Gleichgewichte und die Beziehungen zwischen Thermodynamik und Reaktionskinetik homogener systeme. Zeitschrift Für Physikalische Chemie 39:257–303.

Weidemann A, Richter S, Stein M, Sahle S, Gauges R, Gabdoulline R, Surovtsova I, Semmelrock N, Besson B, Rojas I, Wade R, Kummer U. 2008. SYCAMORE—a systems biology computational analysis and modeling research environment. Bioinformatics (Oxford, England) 24:1463–1464. DOI: https://doi.org/10.1093/bioinformatics/btn207, PMID: 18463116

Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S,
Evelo CT, Finkers R, Gonzalez-Beltran A, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3:160018. DOI: https://doi.org/10.1038/sdata.2016.18, PMID: 26978244

Wilmes KA, Clopath C. 2019. Inhibitory microcircuits for top-down plasticity of sensory representations. Nature Communications 10:5055. DOI: https://doi.org/10.1038/s41467-019-12972-2, PMID: 31699994

Wittig U, Kania R, Golebiewski M, Rey M, Shi L, Jong L, Algae A, Weidemann A, Sauer-Danzen H, Mir S, Krebs O, Bittkowski M, Wetsch E, Rojas I, Müller W. 2012. SABIO-RK--database for biochemical reaction kinetics. Nucleic Acids Research 40:D790–D796. DOI: https://doi.org/10.1093/nar/gkr1046, PMID: 2220587

Wittig U, Rey M, Weidemann A, Müller W. 2017. Data management and data enrichment for systems biology projects. Journal of Biotechnology 261:229–237. DOI: https://doi.org/10.1016/j.jbiotec.2017.06.007, PMID: 28606610

Wittig U, Rey M, Weidemann A, Kania R, Müller W. 2018. SABIO-RK: an updated resource for manually curated biochemical reaction kinetics. Nucleic Acids Research 46:D656–D660. DOI: https://doi.org/10.1093/nar/gkx1065, PMID: 29092055

Wolstencroft K, Krebs O, Snoep JL, Stanford NJ, Bacall F, Golebiewski M, Kuzyakov R, Nguyen Q, Owen S, Soiland-Reyes S, Straszewski J, van Niekerk DD, Williams AR, Malmström L, Rinn B, Müller W, Goble C. 2017. FAIRDOMHub: a repository and collaboration environment for sharing systems biology research. Nucleic Acids Research 45:D404–D407. DOI: https://doi.org/10.1093/nar/gkw1032, PMID: 27899646

wwPDB consortium. 2019. Protein Data Bank: the single global archive for 3D macromolecular structure data. Nucleic Acids Research 47:D520–D528. DOI: https://doi.org/10.1093/nar/gky949, PMID: 30357364

Xie L, Ge X, Tan H, Xie L, Zhang Y, Hart T, Yang X, Bourne PE. 2014. Towards structural systems pharmacology to study complex diseases and personalized medicine. PLOS Computational Biology 10:e1003554. DOI: https://doi.org/10.1371/journal.pcbi.1003554, PMID: 24830652

Zi Z. 2011. Sensitivity analysis approaches applied to systems biology models. IET Systems Biology 5:336. DOI: https://doi.org/10.1049/iet-syb.2011.0015, PMID: 22129029