



## ModelDB: A Database to Support Computational Neuroscience

MICHAEL L. HINES\*

*Department of Computer Science, Yale University, New Haven CT, USA*

michael.hines@yale.edu

THOMAS MORSE

*Department of Neurobiology, Yale University, New Haven, CT, USA*

MICHELE MIGLIORE

*Institute of Biophysics, National Research Council, Palermo, Italy*

NICHOLAS T. CARNEVALE

*Department of Psychology, Yale University, New Haven, CT, USA*

GORDON M. SHEPHERD

*Department of Neurobiology, Yale University, New Haven, CT, USA*

*Received January 15, 2004; Revised January 15, 2004; Accepted March 2, 2004*

Action Editor: John P. Miller

**Abstract.** Wider dissemination and testing of computational models are crucial to the field of computational neuroscience. Databases are being developed to meet this need. ModelDB is a web-accessible database for convenient entry, retrieval, and running of published models on different platforms. This article provides a guide to entering a new model into ModelDB.

**Keywords:** database model

### 1. Introduction

In computational neuroscience, as in science in general, it is essential that results be testable in depth by other laboratories using the same methods. The obstacles to replicating and testing published computational models are considerable. For example, most papers do not include a complete model specification, and typographical errors are common. In principle one should be able to determine the model specification from the

authors' computer code, but there are many reasons why this seldom works out in practice—in the interval between completion of the work and publication of the article, hard drives crash, archived copies are lost, key authors move to new positions leaving no one in the lab who can identify the files that were actually used.

Thus there is a critical need for a database where published models can be archived so that they can be openly accessed, downloaded, and tested. ModelDB has been developed to answer this need (Migliore et al., 2003). It now contains over 100 models that have been presented in the scientific literature, and its curators work with

\*To whom correspondence should be addressed.

authors to ensure that new entries to the database are properly searchable and easy to download. ModelDB appears well suited to be used by authors of articles in the Journal of Computational Neuroscience as a site for depositing models, preferably at the time of publication.

This paper describes the process of entering a model presented in an existing publication into ModelDB. Much of the same information that an author provides at the time of online manuscript submission is also needed to enable the search functions of ModelDB. Therefore we may be able to implement simplified procedures to expedite model entry at the time a manuscript is submitted for publication.

## 2. Depositing a New Model in ModelDB

### 2.1. Overview of Model Entry

To best understand the process of entering a model, it helps to first examine ModelDB from the perspective of a user who is searching for a model. ModelDB's home page (<http://senselab.med.yale.edu/senselab/modeldb/>) (Fig. 1) has links for finding and retrieving models, searching for publications related

to modeling, and for authors to deposit new models in the database. Model searches produce lists of models that match search criteria such as the name of any author, the name of the model, the type of neuron, the presence of a particular property (currents, receptors, transmitters), the existence of a particular word anywhere in the model's files, the related concept (e.g. synaptic plasticity), or the model's simulation environment or programming language (e.g. C/C++, GENESIS, SNNAP). There are also links that generate lists of all available models of networks, neurons, synapses, neuromuscular junctions, axons, or voltage gated ion channels. These lists contain links to pages that describe individual models and allow users to download models that are of interest.

In order for these searches to produce accurate results, there are two kinds of data that you must provide when you deposit your model into ModelDB. First and most obvious is a zip file that contains the computer file or files that constitute the model itself. The indispensable core of a model submission is the code that constitutes a complete specification of those attributes of the original biological system that are represented in the model. Depending on the model, these attributes may include cellular and subcellular anatomy, biophysical and pharmacological properties,

ModelDB provides an accessible location for storing and efficiently retrieving compartmental neuron models. ModelDB is tightly coupled with [NeuronDB](#). Models can be coded in any language for any environment, though ModelDB has been initially constructed for use with [NEURON](#) and [GENESIS](#). Model code can be viewed before downloading and browsers can be set to auto-launch the models. [Help](#)

- Search for models by author name
- List models sorted by [first author](#), by [each author](#) or by [model name](#)
- Find models of a particular [Neuron](#) type
- Find models containing a particular Property: [Currents](#), [Receptors](#), or [Transmitters](#)
- Find models that relate to a [Concept](#), e.g. synaptic plasticity, pattern recognition, etc.
- Find models that run in a particular [Simulation environment](#)
- List models of: [Networks](#), [Neurons](#), [Synapses](#) (and ligand-gated ion channels), [Neuromuscular Junctions](#), [Axons](#), voltage-gated [Ion Channels](#)
- Find models containing the following words    Case Sensitive
- [Search for publications in ModelDB](#) or [in PubMed](#)

---

- [Create or edit](#) a private model

Figure 1. ModelDB's home page (<http://senselab.med.yale.edu/senselab/modeldb/>) contains many links for model searches. Near the bottom of the page is "Create or edit", which is for authors who are depositing or editing their own models.

network connectivity, musculoskeletal dynamics, etc.. Ideally you might also include just enough control and interface code so that a naive user can easily run an “oscilloscope level” simulation that reproduces one or more of the figures from the published paper. Finally, it is a good idea to include a plain text “readme” file with a few comments and instructions on how to use the computer code. If questions arise about what to include, these can be quickly and easily resolved by discussion with one of ModelDB’s curators.

The second kind of data that must be entered is the information that users need to find the model and understand what it is about. This information includes a concise statement of the purpose of the model and how to use it (often derived from the paper’s abstract), a brief manifest of the biological properties that are represented in it, and a complete citation of the paper in which the model was described. Once this information has been entered, the ModelDB curators will take care of entering both the citation of your paper and the complete reference list of that paper into ModelDB’s own searchable database of publications related to modeling.

## 2.2. Creating a Private Model

The process of depositing a new model into ModelDB begins when you create a “private” model in the database. Access to a private model is controlled by a password that you select so that only you, the ModelDB curators, and individuals to whom you give the password can see the information and files that you entered into the database. While your model is private, you can make whatever changes you like to it.

ModelDB has been designed to make the task of depositing a new model quite straightforward. Everything you need to do is managed by a single web page that simplifies and standardizes this process.

Start by going to ModelDB’s WWW site (<http://senselab.med.yale.edu/senselab/modeldb/>) and clicking on the link “Create or edit a private model.” This brings up the Create Model page (Fig. 2). Type the name of the model you want to deposit into the “Name” field. Beneath this is the “Access Code” field; you will want to make up a keyword and type it here. As mentioned above, this keyword is used to control access to the model until it is made public. The “Notes” field is for a short description of the model, a citation of the paper in which the model was described, and any other brief comments that you want users to see when they find it.

Figure 2. The top of the Create Model page. This page is used to enter new models into ModelDB.

Farther down the Create Model page is a set of controls for specifying the attributes of your model, so that user searches will return complete and accurate results. These include the neurons (Fig. 3), receptors, currents, and transmitters that may be represented in the model, its anatomical level (ranging from synapse at one extreme, to network at the other), relevant concepts (e.g. synaptic plasticity, dendritic spikes, synchronization, pattern recognition), the software used to implement it

Figure 3. The Create Model page contains several scrollable lists for specifying searchable model attributes. For example, here is the list for specifying what kind of neurons are represented in your model. You can choose any number of items from this list by clicking on them, and you can also type additional items in the “Other Neuron” field if necessary.

The canonical cortical circuitry of a universal robot	
Upload file	Edit
Delete	Clone
This model belongs to Rossum the younger.	
This is a simplified representation of the already simplified neural circuitry that underlies cognition and behavior in Rossum's Universal Robots, not to be confused with that of the more complex and less successful organisms attributed to his philosopher father. Based on the organisms described in Karel Capek's play "R.U.R.--Rossum's Universal Robots" (1920).	
Model Information (Click on a link to find other models with that property)	
Model Type:	Network
Cell Type(s):	Synthetic neural elements composed of "brain batter"
Channel(s):	I Na, I K, I Cl
Receptor(s):	Gaba, Ampa, NMDA
Transmitter(s):	Gaba, Glutamate
Simulation Environment:	C++
Model Concept(s):	Pattern Recognition, Adaptive Control, Autonomous System
Implementer(s):	Dr. Ing. Rossum

Figure 4. After you “create” a model entry in ModelDB, you will see a page like this that lets you edit your entries and upload a zip file that contains the computer code for your model.

(e.g. C, NEURON, XPP), and the name of the person who implemented the model. When you are finished with these lists, click on the “Create” button at the top of the Create Model page.

Next you will see a page that summarizes the properties that you specified (Fig. 4). This is a good time to click on the “Upload file” button in order to upload the zip file that contains your model’s computer code and readme file. You can also click on the “Edit” button if you need to make any changes to the items that appear in this page. The “Edit” button brings up the “Edit Model” page, which is basically the same as the “Create Model” page but with all of your selections highlighted.

### 2.3. Making the Private Model Public

When you are satisfied with its appearance, there is a “Make public” button that you can click to notify the ModelDB curators, who will make the model public so that any ModelDB user can find and download it. After a model has been made public, you can submit bug fixes and supplementary comments to the ModelDB curators. Old versions of public zip files are stored in the database, and the differences are summarized in automatically generated web pages.

## 3. Discussion

The most important component of a model submission is the code that is the computational equivalent of an experimental preparation (e.g. the representation of a synapse, cell, network etc.). Depending on the model and the paper, other details may also be important,

e.g. custom code for initialization, complex stimulation protocols, and sophisticated post-run data analysis, but generally these items are secondary to the specification of the biology.

### 3.1. Final Comments on Model Submissions

When you prepare the zip file that contains your code, it might be helpful to ask yourself these three questions.

1. Have I included the complete model specification? This is what most users are looking for when they search for models. The scientific value of the code that authors submit derives entirely from the fact that it is authoritative, in the sense that it amounts to a reliable and definitive statement of the biological properties embodied by the model that was actually used in the published work.
2. Is it executable? It is desirable for the model specification code to be packaged with just enough control and interface code so that someone else can easily run a simulation, and even better if the simulation reproduces one or more figures from the original paper. This is not mandatory when a model is submitted by its author (see below), but it is a good test of model integrity and the ultimate verification that the biological specification is the same as what was used for the published work.
3. Is it understandable? Code does not have to be “pretty” or commented, and authors don’t need to write any special tutorial text or create a bullet-proofed user interface for it. Code can be entirely “as is”; after all, the goal of ModelDB is to be a

resource for computational neuroscience research, not to be a showcase for textbook examples of “best programming practice.” Of course, authors who want to encourage wide recognition and use of their models may find it worthwhile to invest some time and effort to clean up their code and add tutorial material, but this is entirely a matter of personal preference.

The emphasis in this paper has been on new models, but we also welcome the entry of already published models into ModelDB, including “legacy models” of “historical” interest. We are particularly interested in the actual legacy code because it is authoritative, even if it was written years ago for an obsolete hardware and software environment and no longer runs. Even when code is not executable, one may at least be able to derive the model specification from it.

Models may also be deposited that have been implemented by someone other than the original author. This is most likely to occur when a model is ported to a new simulation environment. In such cases, it is crucial that the provided code includes not just the biological specification but also whatever is needed to reproduce at least one figure from the original paper. When the actual author provides a model specification, it may be taken on faith, but an implementer who attempts to recreate someone else’s model should provide some proof of model correctness.

### 3.2. Conclusion

ModelDB is still evolving, so some of the details presented in this article may change. Regardless of whatever changes may occur over time, each page in ModelDB has a link to a set of “help” notes that address common questions and problems, and users can always contact the curators directly to discuss special issues that might arise. We also welcome and appreciate feedback and suggestions for improvement, since an important design goal is for ModelDB to achieve a high degree of functionality while at the same time keeping its interface simple and easy to use.

### Acknowledgments

This work was supported by NIMH, NINDS, and NCI through the Human Brain Project Grant 5P01DC004732. We thank the Yale Center for Medical Informatics, especially Luis Marengo for his help with database technology advice. Partial support from the Institute of Biophysics (National Research Council, Italy) for MM is also acknowledged.

### Reference

Migliore M, Morse TM, Davison AP, Marengo L, Shepherd GM, Hines ML (2003) Making models publicly accessible to support computational neuroscience. *Neuroinformatics* 1: 135–140.