ModelView

extracting model structure
and presenting it on the web with NEURON

14 May 2014
Open Source Brain
New Haven, CT
ModelDB

- 880 published computational neuroscience models
- 67 simulators
- Part of the SenseLab suite of databases

modeldb.yale.edu
What is needed for reproducibility?
What is needed for reproducibility?

Model

an approximation of the system of interest
e.g. in NEURON or NeuroML
What is needed for reproducibility?

**Model**

an approximation of the system of interest
e.g. in NEURON or NeuroML

**Experimental protocol**

what was done with the model to produce the data
What is needed for reproducibility?

Model

an approximation of the system of interest

E.g. in NEURON or NeuroML

Experimental protocol

what was done with the model to produce the data

in every ModelDB entry
Models May Contain Many Files

- 0 to 4: 28.5%
- 5 to 9: 14.6%
- 10 to 19: 21.7%
- 20 to 49: 23.5%
- 50 to 99: 6.7%
- 100 to 199: 3.7%
- 200+:

Total models may contain many files.
Model Files May Be Large

- 1,000 to 2,000: 30.7%
- 1,000 to 2,000: 20.6%
- 500 to 1,000: 13.6%
- 500 to 1,000: 10.9%
- 500 to 1,000: 9.1%
- 500 to 1,000: 12.1%
- 100,000+: 3.0%
- 10,000 to 100,000: 0.1%
Model Files Might Be Structured...

<cell name = "SimpleCell">

<segments xmlns="http://morphml.org/morphml/schema">
  <segment id ="0" name="Soma" cable="0">
    <proximal x="0" y="0" z="0" diameter="10"/>
    <distal x="10" y="0" z="0" diameter="10"/>
  </segment>
  <segment id ="1" name="Dendrite" parent="0" cable="1">
    <proximal x="10" y="0" z="0" diameter="3"/>
    <!-- Note 3D point same as parent, diam different-->
    <distal x="20" y="0" z="0" diameter="3"/>
  </segment>
</segments>

<cables xmlns="http://morphml.org/morphml/schema">
  <cable id="0" name="SomaCable" />
  <cable id="1" name="DendriteCable" />
</cables>

</cell>
Amyloid beta (IA block) effects on a model CA1 pyramid cell (Morse et al. 2010)

Accession: 87284

The model simulations provide evidence oblique dendrites in CA1 pyramid neurons are susceptible to hyper-excitability by amyloid beta block of the transient K+ channel.


Citations Citation Browser

Model Information (Click on a link to find other models with that property)

Model Type: Neuron or other electrically excitable cell

Brain Region(s): Organism:

Cell Type(s): Hippocampus CA1 pyramid cell

Channel(s): I_NaL, I_high threshold, I_N, I_low threshold, I_A, I_K, I_L

Gap Junction(s):

Receptor(s):

Gene(s):

Simulation Environment:

Model Concept(s): Dendritic Action Potentials; Active Dendrites; Detailed Neuronal Models; Pathophysiology; Aging/Alzheimer's; Implementation:

Citation(s): Carnevale NT, Tarda (2006) Alzheimers. Tom Munro at Yale University.

Search NeuronDB for information about: Hippocampus CA1 pyramid cell; I_NaL; I_high threshold; I_N; I_Low threshold; I_A; I_K; I_L

Model files Download zip file Auto-launch Download the displayed file Simulation Platform Model view Help downloading and running models

// fig2A_c19662.hoc
// to replace original model's default cell with c19662

// cvode (active(I)) = uncomment for cvode however cvode incompat. for calcium analysis

// fig3A_c19662.hoc
// /opt/fig2A.hoc from Migliore et al 2005 oblique model
// to replace original model's default cell with c19662

// xmpen "geo5828804.hoc" // a cell thats shape is similar to the one in Chen et al. 2001.
// load_file("obliques_primary_tuft.hoc") // defines SectionList
// load_file("finxseg.hoc")

// Omori et al. 2009 define a step function for Rm that is a high resistance, RmDend nec
// the soma and then 350 +/- 100 microfarads from the soma drops to 3/10 that value, RmDist
// for the soma. Soma value is 300, RmDend value is 4/5. 31622 lower limit from Omori et al. 2009
// Michele's original value: 2868

// load_file("leaky_distal.hoc") // procedures make_leaky_distal_Rm, make_const_distal_Rm

// now handled in paper...hoc: dt = 0.025 // dt =0.1 is OK for demo
gna = 0.25

// for all ( nseg2=27 )

// for all ( tot=20 )
soma (stim new IClamp(rel)
stim.amplitude = 2.5 in original Migliore et al. 2005 model
stim.dur=1.5
stim.delay=1

// the below get reset later but must be defined to show up in gui text editors
grec = 0.01 // used for setting the T, L, and N Ca2+ currents max conductance
Krec = 0.01 // used for setting the Ca2+ sensitive K max conductance
gt = 0.01 // used for setting the T-Ca current max cond. to a particular value
How can we present this information in a human accessible way?
Part I: extracting model structure
The single most important piece of metadata is the simulator type, because that affects how everything else is interpreted.

**Run protocols** describe how to run a simulation.

Metadata may be **static** or **dynamic**.

Static metadata, such as the doi of the paper that describes the model, is read only once, at preprocessing.

To facilitate automated extracting metadata from a model’s ModelDB page, we introduced ids for each of the named fields in the HTML.

Dynamic metadata, such as component reuse, is loaded at display time by querying a web service.
Source code

• Easy to extract information that the simulator requires to be expressed in a structured way (e.g. SUFFIX and POINT_PROCESS designations in NEURON).

• We use this to map specific inserted mechanisms to the source code that defines them.

• Comments and flow control statements (e.g. if) complicate this analysis.

![Diagram showing flow of Model Files to Mechanisms and their specifications](http://www.clker.com/cliparts/9/3/b/9/1194984413232885755folder_01.svg.hi.png)
Simulator introspection

- Gold standard: run the model. Ask the simulator what it did.
- This is used for most of the data in our model view tool (morphology, mechanism localization, parameters, etc...)
- Requires simulator support.

Model Files → NEURON

Q: What is the morphology?
A: ...

Q: What channels are used? Where?
A: ...

Q: What parameters?
A: ...
JSON data structure

```json
{
    'title': 'Long title for page',
    'short_title': 'Migliore et al 2004',
    'neuron': [{
        'title': 'Pyramidal Cell',
        'morphology': [[[x1, y1, z1, d1], ...]]
    }, {... another neuron ...}],
    'colorbars': [
        {'type': 'css', 'css': ...}, ...
    ],
    'tree': [{
        'text': 'row text',
        'children': [...],
        'action': [{kind=...}],
        'noop': true or false or omitted
        'include': url or omitted
    }]
}
```
Part II:
Web presentation
ModelView web app

http://senselab.med.yale.edu/modeldb/modelview/modelview.html#87284_1
Amyloid beta (IA block) effects on a model CA1 pyramidal cell (Morse et al. 2010)

Accession: 87284

The model simulations provide evidence oblique dendrites in CA1 pyramidal neurons are susceptible to hyper-excitability by amyloid beta block of the transient K+ channel, IA. See paper for details.

Morse et al. 2010

- 194 sections; 974 segments
- 1 real cell
- 0 artificial cells
- 0 NetCon objects
- 0 LinearMechanism objects
- Density Mechanisms
- 1 point processes (0 can receive events) of 1 base classes
- 7 files shared with other ModelDB models
- References
Morse et al. 2010

Density Mechanisms

18 mechanisms in use

- Ra
- cm
- pas
- na_ion
- k_ion
- ca_ion
- cacum
  (cacumm.mod)
  READS: ica
  WRITEx: cai,
  Nonspecific Current
  Present in 193
  sections

- cagk (cagk.mod)
  READS: cai, ek
  WRITEx: ik
  Present in 193
  sections
Morse et al. 2010

194 sections; 974 segments
1 real cell
0 artificial cells
0 NetCon objects
0 LinearMechanism objects
Density Mechanisms
1 point processes (0 can receive events) of 1 base classes
1 IClamp (builtin: ref)
  del = 0
dur = 0.2
amp = 0.1
Global parameters for Point Processes
KSChan definitions for Point Processes
7 files shared with other ModelDB models

root: soma

IClamp at axon(1)
(-81.9109, 438.25, -215.16)

X-Y  X-Z  Y-Z
A model of unitary responses from A/C and PP synapses in CA3 pyramidal cells (Baker et al. 2010)

CA1 pyramidal neuron: effects of R213Q and R312W Kv7.2 mutations (Miceli et al. 2013)

CA3 pyramidal neuron (Safiulina et al. 2010)

CA3 pyramidal neuron: firing properties (Hemond et al. 2008)

distr.mod
cal2.mod
can2.mod
cat.mod
ipulse2.mod
naxn.mod

References

Paper in Front. Neural Circuits

ModelDB Entry

Run Protocol

Compiling
cd CA1_abeta
nrvmodl

Launching NEURON
nrun -python

Running
from neuron import h
h.load_file("mosinit.hoc")
h.frig1and2()
NeuronWeb

• A general purpose library for presenting scientific information online.

• Uses JavaScript to dynamically rewrite the DOM to add and remove elements from the web page.

• Native elements, JQueryUI, Flot charts provide the core widgets.
NeuronWeb

- A general purpose library for presenting scientific information online.
- Level I: JavaScript layer for stand-alone web apps.
  - e.g. create a dialog with one function call, a graph with another.
  - Supports automatic layout adjustments.
- Level II: Interface controlled via websockets.
- Level III: Python library for controlling the interface.
Conclusions

- Computational neuroscience models may be complicated.
- The source code is not human interpretable.
- Developed a web-based visualization tool that combines information from ModelDB metadata, source code text-mining, and introspection.
- The web viewer is simulator independent; general clickable tree and graph support is provided. Data is loaded from pre-generated JSON or dynamically generated JSONP.
- Model View should facilitate the interpretation and reuse of computational models.
- Model View could potentially be used to convert models to work with other simulators and for identification and analysis of repeated modeling motifs.
Acknowledgements

Michael Hines
Gordon Shepherd
Nicole Flokos
Tom Morse

NATIONAL INSTITUTES
OF HEALTH

2T15LM007056
5R01DC009977
NS11613