New Developments in ModelDB

INCF Exhibit Booth

Thomas M. Morse[1], Rixin Wang[2], Luis Marenco[2], N. Ted Carnevale[1], Michael L. Hines[1], Gordon M. Shepherd[1].

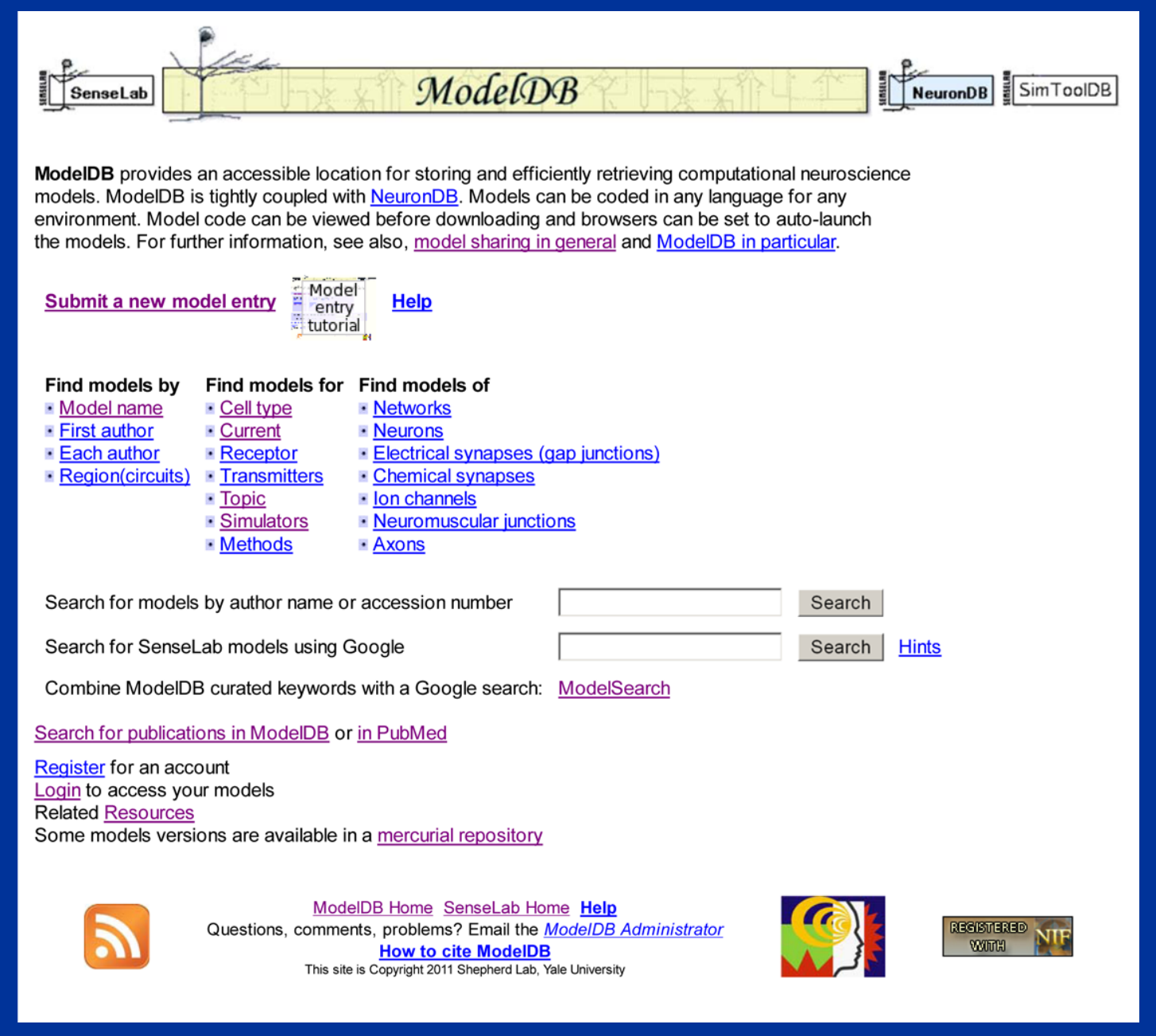
[1]Department of Neurobiology, [2]IYale Center for Medical Informatics , Yale University School of Medicine, New Haven, CT, USA. Contact e-mail: tom.morse@yale.edu

Abstract:

Over the last year (ending 10/31/2011) we have had 75 models contributed, of which 24 were neural network, 34 were neuron or other electrically excitable cell, and the remainder were at smaller spatial scales (dendrite, synapse, axon, channel), or methods. ModelDB has grown to over 660 models. This large number requires new search tools to find models. We have developed a new search engine, ModelSearch, that combines an open-ended Google search with the precise control of ModelDB curated keywords. The primary usefulness is to be able to search for terms which have not been curated. To do this modelers enter terms in a google search box while restricting the results to also contain curated terms such as a particular simulator by selecting from pop-up lists of ModelDB keywords. This search engine was developed in response to external requests, most of which were to find Neuron simulator models with constraints. Example constraints were network models and python code, or models that contained integrate and fire neurons. Each of these constraints contained uncurated terms (python Neuron code, and integrate and fire neurons). Google alone however provides too many false positive results from the term Neuron which often represents the presence of that cell type in a model rather than the Neuron simulator. Tutorial examples will be presented in a live demonstration..

Introduction:

For each tutorial search first select the ModelSearch link from the ModelDB home page:



ModelDB provides an accessible location for storing and efficiently retrieving computational neuroscience models. ModelDB is tightly coupled with NeuronDB. Models can be coded in any language for any environment. Model code can be viewed before downloading and browsers can be set to auto-launch the models. For further information, see also, model sharing in general and ModelDB in particular.

Submit a new model entry | Model entry tutorial | Help

Find models by	Find models for	Find models of
Model name	Cell type	Networks
First author	Current	Neurons
Each author	Receptor	Electrical synapses (gap junctions)
Region(circuits)	Transmitters	Chemical synapses
	Topic	Ion channels
	Simulators	Neuromuscular junctions
	Methods	Axons

Search for models by author name or accession number [ ] Search

Search for SenseLab models using Google [ ] Search Hints

Combine ModelDB curated keywords with a Google search: ModelSearch

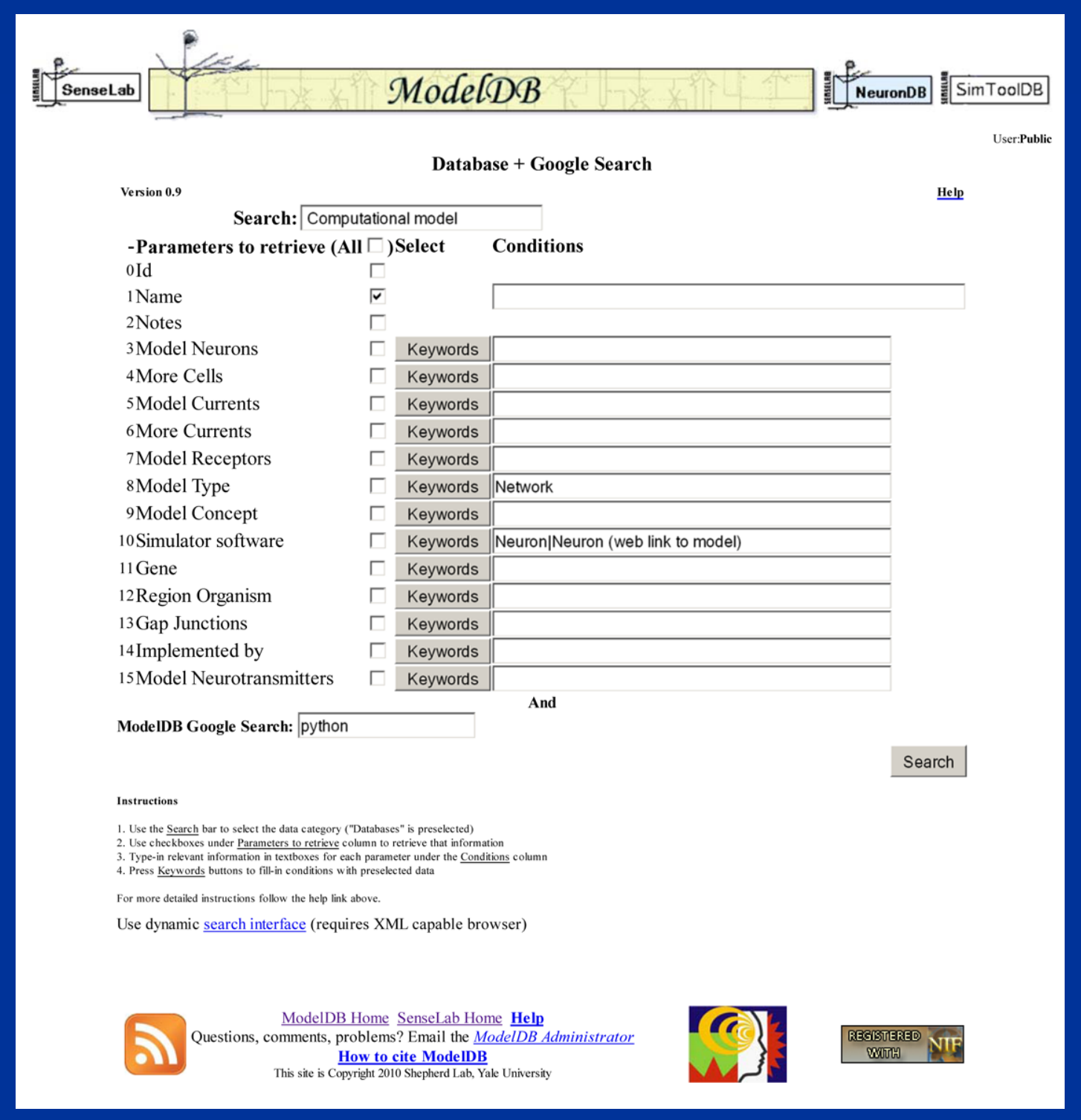
Search for publications in ModelDB or in PubMed

Register for an account
Login to access your models
Related Resources
Some models versions are available in a mercurial repository

ModelDB Home SenseLab Home Help
Questions, comments, problems? Email the ModelDB Administrator
How to cite ModelDB
This site is Copyright 2011 Shepherd Lab, Yale University

Example 1:

Search for NEURON simulator models which contain both python code and are network models. In the ModelSearch page shown below click the Keyword button next to Model Type and select network in the pop up list and close the box. Similarly select the Neuron simulator and Neuron simulator (web link to model) options by clicking on the Keywords button next to Simulators. type in python and press enter or click the search button.



Database + Google Search

User:Public

Version 0.9 Help

Search: Computational model

-Parameters to retrieve (All □)	Select		Conditions
0 Id	□		
1 Name	☑		
2 Notes	□		
3 Model Neurons	□	Keywords	
4 More Cells	□	Keywords	
5 Model Currents	□	Keywords	
6 More Currents	□	Keywords	
7 Model Receptors	□	Keywords	
8 Model Type	□	Keywords	Network
9 Model Concept	□	Keywords	
10 Simulator software	□	Keywords	Neuron|Neuron (web link to model)
11 Gene	□	Keywords	
12 Region Organism	□	Keywords	
13 Gap Junctions	□	Keywords	
14 Implemented by	□	Keywords	
15 Model Neurotransmitters	□	Keywords	

And

ModelDB Google Search: python

Search

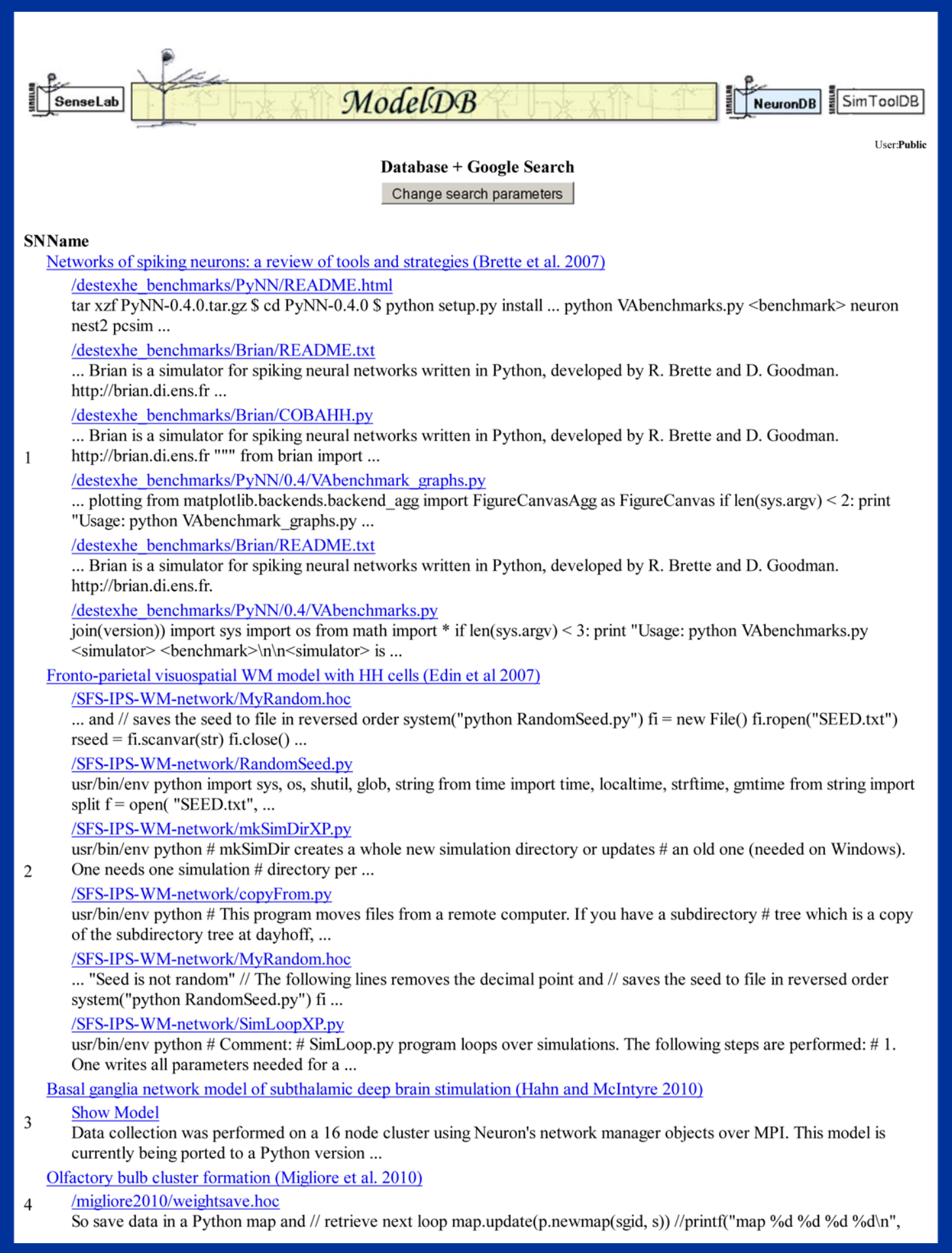
Instructions

1. Use the Search bar to select the data category ("Databases" is preselected)
2. Use checkboxes under Parameters to retrieve column to retrieve that information
3. Type-in relevant information in textboxes for each parameter under the Conditions column
4. Press Keywords buttons to fill-in conditions with preselected data

For more detailed instructions follow the help link above.

Use dynamic search interface (requires XML capable browser)

ModelDB Home SenseLab Home Help
Questions, comments, problems? Email the ModelDB Administrator
How to cite ModelDB
This site is Copyright 2010 Shepherd Lab, Yale University



Database + Google Search

User:Public

Change search parameters

SNName

Networks of spiking neurons: a review of tools and strategies (Brette et al. 2007)

1
/destexhe_benchmarks/PyNN/README.html
tar xzf PyNN-0.4.0.tar.gz $ cd PyNN-0.4.0 $ python setup.py install ... python VAbenchmarks.py <benchmark> neuron nest2 pcsim ...
/destexhe_benchmarks/Brian/README.txt
... Brian is a simulator for spiking neural networks written in Python, developed by R. Brette and D. Goodman. http://brian.di.ens.fr ...
/destexhe_benchmarks/Brian/COBAHH.py
... Brian is a simulator for spiking neural networks written in Python, developed by R. Brette and D. Goodman. http://brian.di.ens.fr """ from brian import ...
/destexhe_benchmarks/PyNN/0.4/VAbenchmark_graphs.py
... plotting from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas if len(sys.argv) < 2: print "Usage: python VAbenchmark_graphs.py ...
/destexhe_benchmarks/Brian/README.txt
... Brian is a simulator for spiking neural networks written in Python, developed by R. Brette and D. Goodman. http://brian.di.ens.fr.
/destexhe_benchmarks/PyNN/0.4/VAbenchmarks.py
join(version)) import sys import os from math import * if len(sys.argv) < 3: print "Usage: python VAbenchmarks.py <simulator> <benchmark>\n\n<simulator> is ...

Fronto-parietal visuospatial WM model with HH cells (Edin et al 2007)

2
/SFS-IPS-WM-network/MyRandom.hoc
... and // saves the seed to file in reversed order system("python RandomSeed.py") fi = new File() fi.ropen("SEED.txt") rseed = fi.scanvar(str) fi.close() ...
/SFS-IPS-WM-network/RandomSeed.py
usr/bin/env python import sys, os, shutil, glob, string from time import time, localtime, strftime, gmtime from string import split f = open("SEED.txt", ...
/SFS-IPS-WM-network/mkSimDirXP.py
usr/bin/env python # mkSimDir creates a whole new simulation directory or updates # an old one (needed on Windows). One needs one simulation # directory per ...
/SFS-IPS-WM-network/copyFrom.py
usr/bin/env python # This program moves files from a remote computer. If you have a subdirectory # tree which is a copy of the subdirectory tree at dayhoff, ...
/SFS-IPS-WM-network/MyRandom.hoc
... "Seed is not random" // The following lines removes the decimal point and // saves the seed to file in reversed order system("python RandomSeed.py") fi ...
/SFS-IPS-WM-network/SimLoopXP.py
usr/bin/env python # Comment: # SimLoop.py program loops over simulations. The following steps are performed: # 1. One writes all parameters needed for a ...

Basal ganglia network model of subthalamic deep brain stimulation (Hahn and McIntyre 2010)

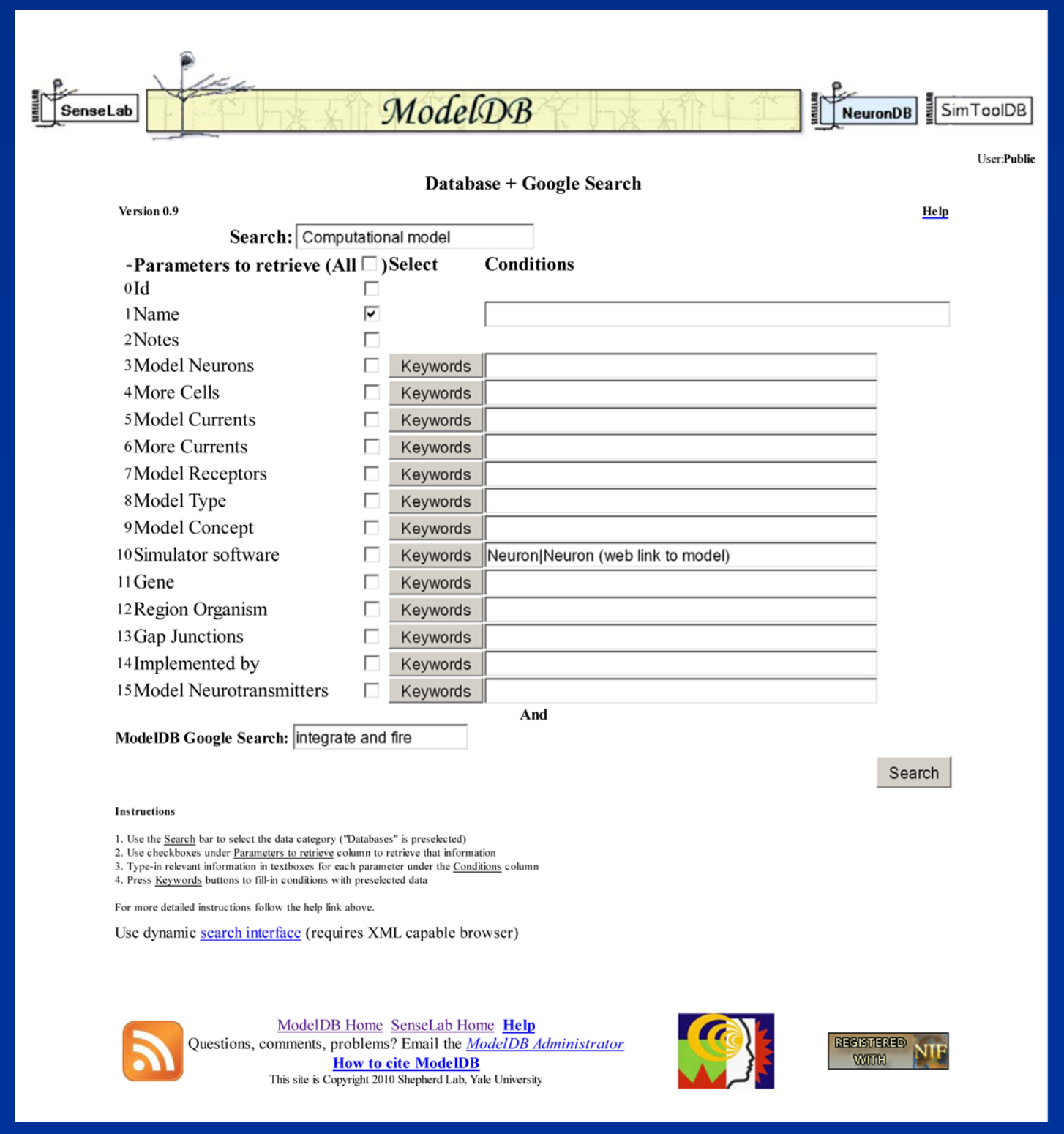
3
Show Model
Data collection was performed on a 16 node cluster using Neuron's network manager objects over MPI. This model is currently being ported to a Python version ...

Olfactory bulb cluster formation (Migliore et al. 2010)

4
/migliore2010/weightsave.hoc
So save data in a Python map and // retrieve next loop map.update(p.newmap(sgid, s)) //printf("map %d %d %d %d\n",

Example 2:

earch for NEURON simulator models which also contain integrate and fire neurons. In the ModelSearch form again select the NEURON simulator. Type “integrate and fire” into the Google search box and press enter.



Database + Google Search

User:Public

Version 0.9 Help

Search: Computational model

-Parameters to retrieve (All □)	Select		Conditions
0 Id	□		
1 Name	☑		
2 Notes	□		
3 Model Neurons	□	Keywords	
4 More Cells	□	Keywords	
5 Model Currents	□	Keywords	
6 More Currents	□	Keywords	
7 Model Receptors	□	Keywords	
8 Model Type	□	Keywords	
9 Model Concept	□	Keywords	
10 Simulator software	□	Keywords	Neuron|Neuron (web link to model)
11 Gene	□	Keywords	
12 Region Organism	□	Keywords	
13 Gap Junctions	□	Keywords	
14 Implemented by	□	Keywords	
15 Model Neurotransmitters	□	Keywords	

And

ModelDB Google Search: integrate and fire

Search

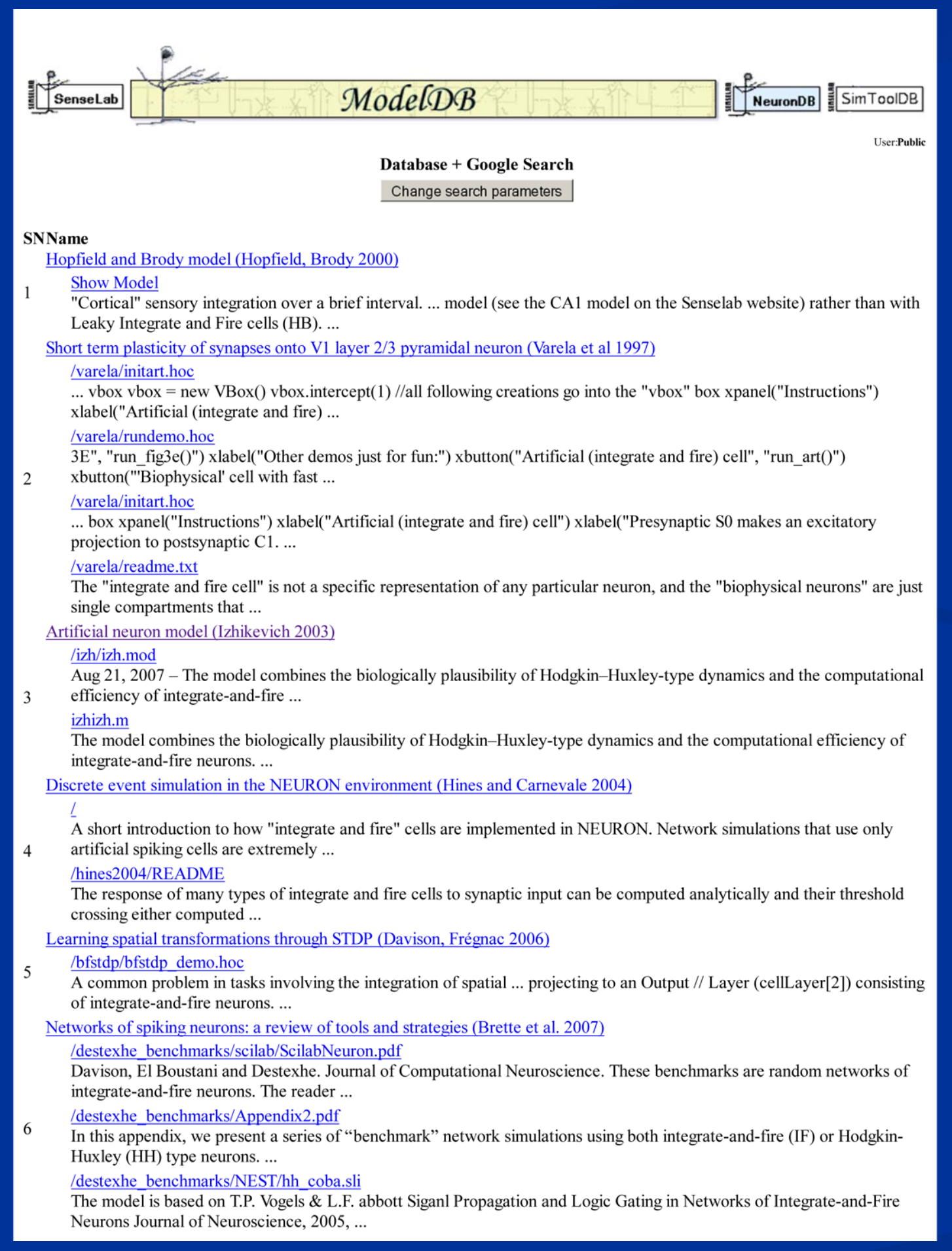
Instructions

1. Use the Search bar to select the data category ("Databases" is preselected)
2. Use checkboxes under Parameters to retrieve column to retrieve that information
3. Type-in relevant information in textboxes for each parameter under the Conditions column
4. Press Keywords buttons to fill-in conditions with preselected data

For more detailed instructions follow the help link above.

Use dynamic search interface (requires XML capable browser)

ModelDB Home SenseLab Home Help
Questions, comments, problems? Email the ModelDB Administrator
How to cite ModelDB
This site is Copyright 2010 Shepherd Lab, Yale University



Database + Google Search

User:Public

Change search parameters

SNName

Hopfield and Brody model (Hopfield, Brody 2000)

1
Show Model
"Cortical" sensory integration over a brief interval. ... model (see the CA1 model on the Senselab website) rather than with Leaky Integrate and Fire cells (HB). ...

Short term plasticity of synapses onto V1 layer 2/3 pyramidal neuron (Varela et al 1997)

2
/varela/initart.hoc
... vbox vbox = new VBox() vbox.intercept(1) //all following creations go into the "vbox" box xpanel("Instructions") xlabel("Artificial (integrate and fire) ...
/varela/rundemo.hoc
3E", "run_fig3e()") xlabel("Other demos just for fun:") xbutton("Artificial (integrate and fire) cell", "run_art()") xbutton("'Biophysical' cell with fast ...
/varela/initart.hoc
... box xpanel("Instructions") xlabel("Artificial (integrate and fire) cell") xlabel("Presynaptic S0 makes an excitatory projection to postsynaptic C1. ...
/varela/readme.txt
The "integrate and fire cell" is not a specific representation of any particular neuron, and the "biophysical neurons" are just single compartments that ...

Artificial neuron model (Izhikevich 2003)

3
/izh/izh.mod
Aug 21, 2007 – The model combines the biologically plausibility of Hodgkin–Huxley-type dynamics and the computational efficiency of integrate-and-fire ...
izhizh.m
The model combines the biologically plausibility of Hodgkin–Huxley-type dynamics and the computational efficiency of integrate-and-fire neurons. ...

Discrete event simulation in the NEURON environment (Hines and Carnevale 2004)

4
/
A short introduction to how "integrate and fire" cells are implemented in NEURON. Network simulations that use only artificial spiking cells are extremely ...
/hines2004/README
The response of many types of integrate and fire cells to synaptic input can be computed analytically and their threshold crossing either computed ...

Learning spatial transformations through STDP (Davison, Frégnac 2006)

5
/bfstdp/bfstdp_demo.hoc
A common problem in tasks involving the integration of spatial ... projecting to an Output // Layer (cellLayer[2]) consisting of integrate-and-fire neurons. ...

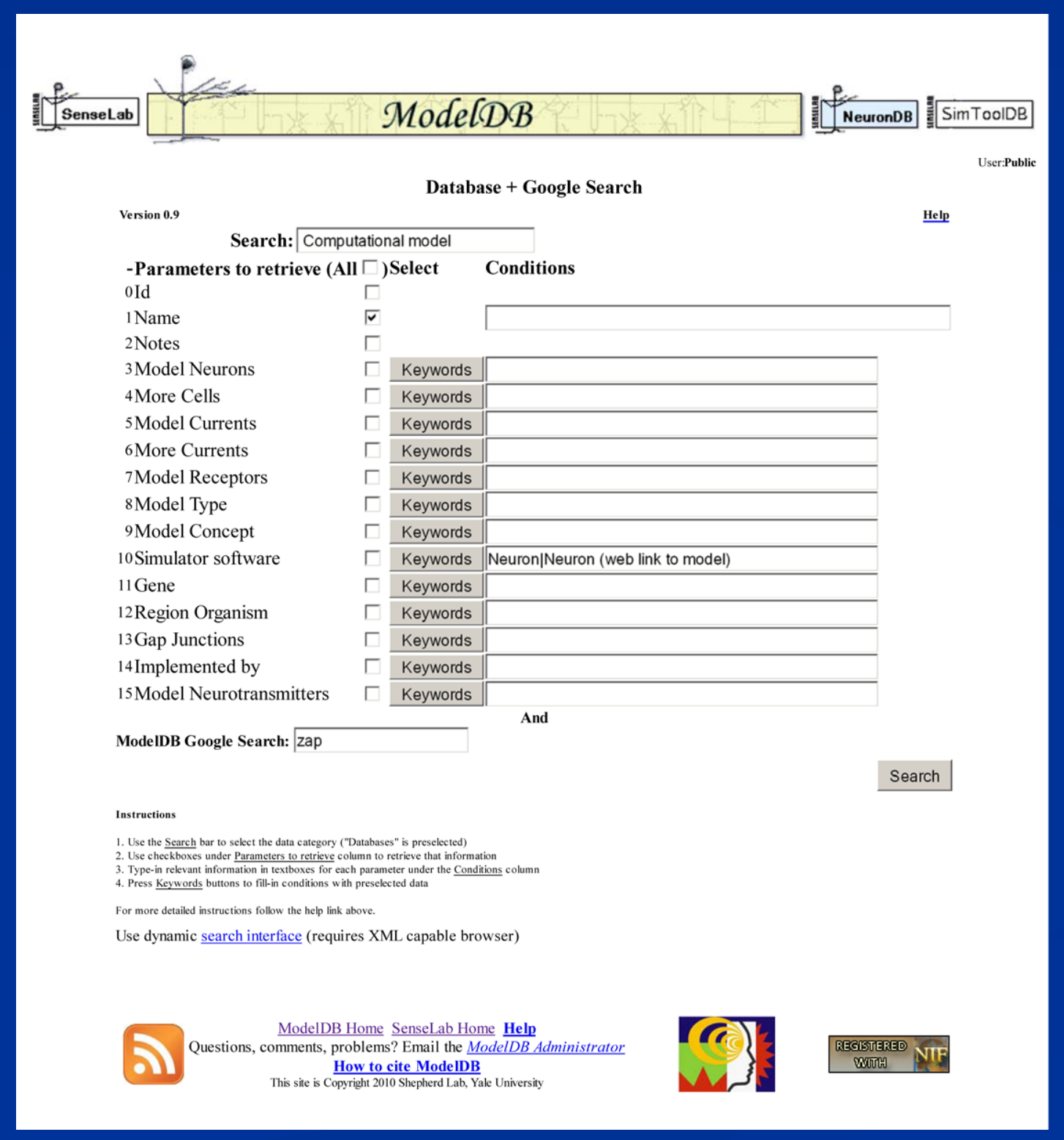
Networks of spiking neurons: a review of tools and strategies (Brette et al. 2007)

6
/destexhe_benchmarks/scilab/ScilabNeuron.pdf
Davison, El Boustani and Destexhe. Journal of Computational Neuroscience. These benchmarks are random networks of integrate-and-fire neurons. The reader ...
/destexhe_benchmarks/Appendix2.pdf
In this appendix, we present a series of “benchmark” network simulations using both integrate-and-fire (IF) or Hodgkin-Huxley (HH) type neurons. ...
/destexhe_benchmarks/NEST/hh_coba.sli
The model is based on T.P. Vogels & L.F. abbott Siganl Propagation and Logic Gating in Networks of Integrate-and-Fire Neurons Journal of Neuroscience, 2005, ...

The search results are collected into groups under a link to each model entry. The file and path to the file and google paragraph are provided.

Example 3:

Search for Neuron simulator models that contain the zap protocol. The zap protocol is an injected current whose frequency rises from 0 to a predetermined value (e.g. 100 Hz). As before select the NEURON simulator and enter zip into the Google search box and press enter.



Database + Google Search

User:Public

Version 0.9 Help

Search: Computational model

-Parameters to retrieve (All □)	Select		Conditions
0 Id	□		
1 Name	☑		
2 Notes	□		
3 Model Neurons	□	Keywords	
4 More Cells	□	Keywords	
5 Model Currents	□	Keywords	
6 More Currents	□	Keywords	
7 Model Receptors	□	Keywords	
8 Model Type	□	Keywords	
9 Model Concept	□	Keywords	
10 Simulator software	□	Keywords	Neuron|Neuron (web link to model)
11 Gene	□	Keywords	
12 Region Organism	□	Keywords	
13 Gap Junctions	□	Keywords	
14 Implemented by	□	Keywords	
15 Model Neurotransmitters	□	Keywords	

And

ModelDB Google Search: zap

Search

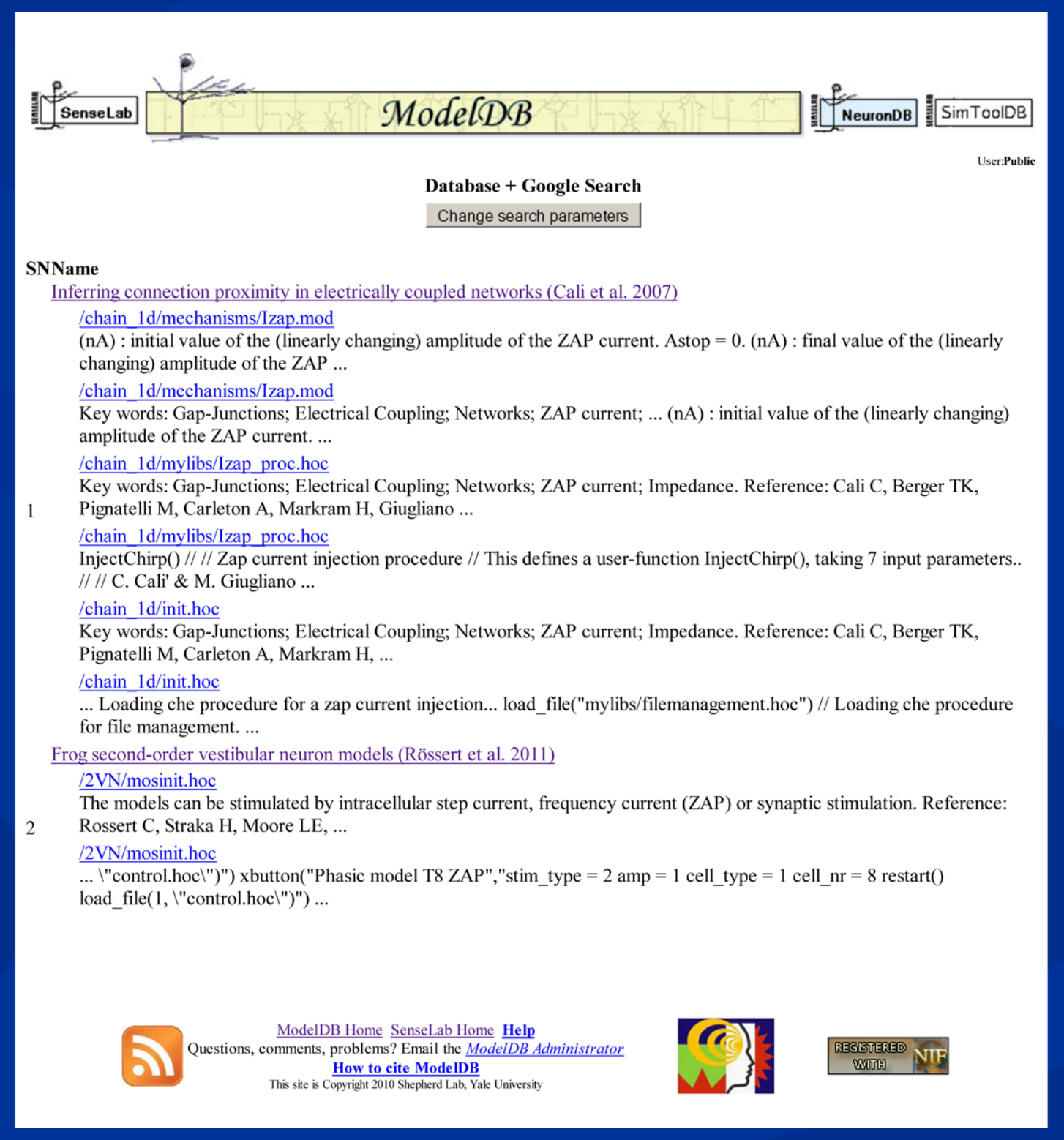
Instructions

1. Use the Search bar to select the data category ("Databases" is preselected)
2. Use checkboxes under Parameters to retrieve column to retrieve that information
3. Type-in relevant information in textboxes for each parameter under the Conditions column
4. Press Keywords buttons to fill-in conditions with preselected data

For more detailed instructions follow the help link above.

Use dynamic search interface (requires XML capable browser)

ModelDB Home SenseLab Home Help
Questions, comments, problems? Email the ModelDB Administrator
How to cite ModelDB
This site is Copyright 2010 Shepherd Lab, Yale University



Database + Google Search

User:Public

Change search parameters

SNName

Inferring connection proximity in electrically coupled networks (Cali et al. 2007)

1
/chain_1d/mechanisms/Izap.mod
(nA) : initial value of the (linearly changing) amplitude of the ZAP current. Astop = 0. (nA) : final value of the (linearly changing) amplitude of the ZAP ...
/chain_1d/mechanisms/Izap.mod
Key words: Gap-Junctions; Electrical Coupling; Networks; ZAP current; ... (nA) : initial value of the (linearly changing) amplitude of the ZAP current. ...
/chain_1d/mylibs/Izap_proc.hoc
Key words: Gap-Junctions; Electrical Coupling; Networks; ZAP current; Impedance. Reference: Cali C, Berger TK, Pignatelli M, Carleton A, Markram H, Giugliano ...
/chain_1d/mylibs/Izap_proc.hoc
InjectChirp() // // Zap current injection procedure // This defines a user-function InjectChirp(), taking 7 input parameters.. // // C. Cali' & M. Giugliano ...
/chain_1d/init.hoc
Key words: Gap-Junctions; Electrical Coupling; Networks; ZAP current; Impedance. Reference: Cali C, Berger TK, Pignatelli M, Carleton A, Markram H, ...
/chain_1d/init.hoc
... Loading che procedure for a zap current injection... load_file("mylibs/filemanagement.hoc") // Loading che procedure for file management. ...

Frog second-order vestibular neuron models (Rössert et al. 2011)

2
/2VN/mosinit.hoc
The models can be stimulated by intracellular step current, frequency current (ZAP) or synaptic stimulation. Reference: Rossert C, Straka H, Moore LE, ...
/2VN/mosinit.hoc
... \"control.hoc\")") xbutton("Phasic model T8 ZAP","stim_type = 2 amp = 1 cell_type = 1 cell_nr = 8 restart() load_file(1, \"control.hoc\")") ...

ModelDB Home SenseLab Home Help
Questions, comments, problems? Email the ModelDB Administrator
How to cite ModelDB
This site is Copyright 2010 Shepherd Lab, Yale University

Acknowledgement

We are grateful for the support of NIH grant 5R01DC009977-03

Technical notes: How ModelSearch works:

ModelSearch works by first storing the result from searching ModelDB’s database for the Keywords selected by the modeler, and then storing the result of the entire Google search. These stored results are then combined: Only those model entries are retained in the search results which were found to be present in both the attributes (ModelDB curated keywords) and google searches. The reported results have the hits collected under each model. The google links are transformed into a path to the file within the model entries file archive and the google paragraph is provided below the link.