

Reproducible Models and Replicable Implementations

Current Trends in Computational Neuroscience

Hans Ekkehard Plessner, UMB

Sharon Crook, ASU

Andrew Davison, CNRS

Presented at SIAM Computational Science & Engineering
Reno, NV, 4 March 2011





Reproducible Models and Replicable Implementations: Current Trends in Computational Neuroscience by Hans E Plessner, UMB/IMT is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](#).

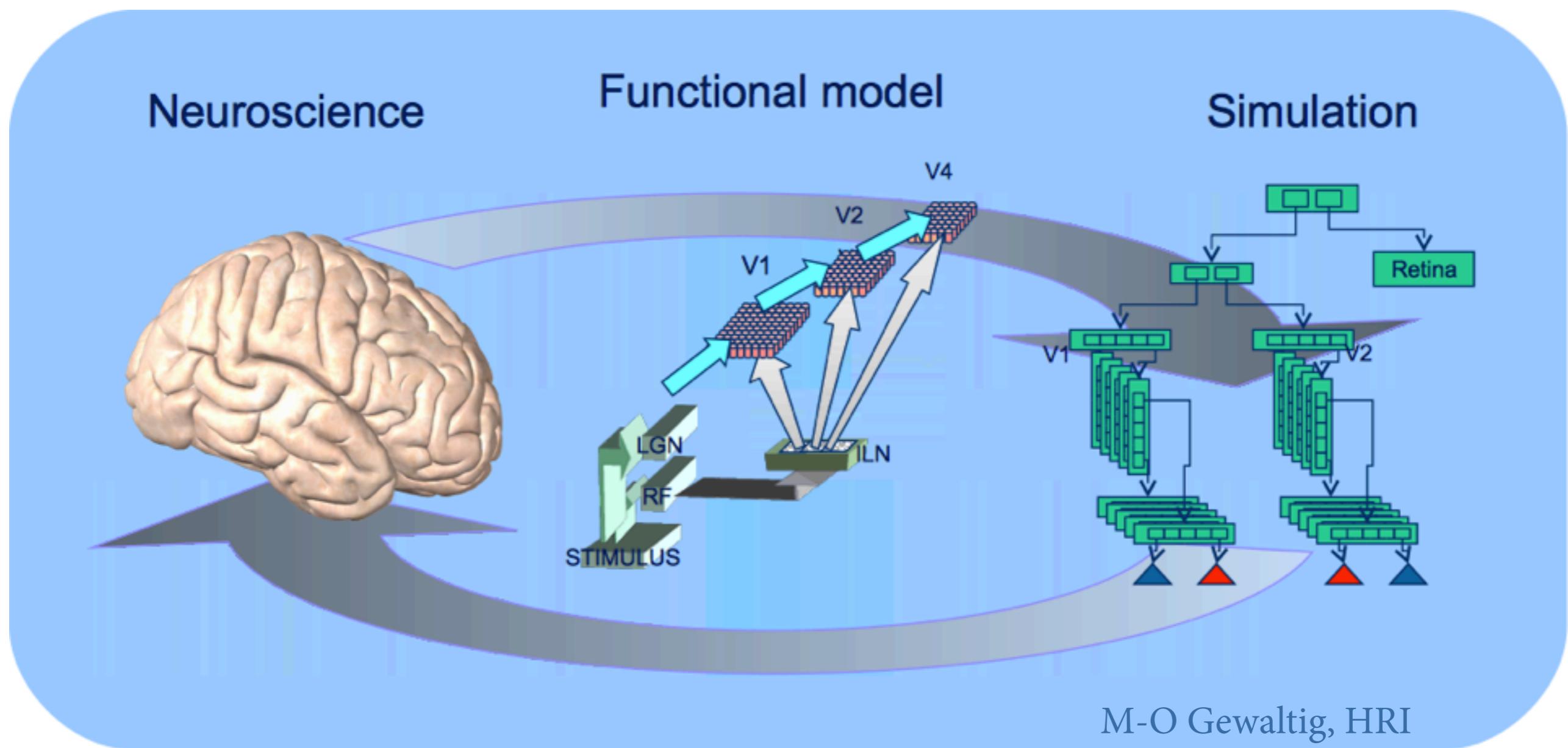
Permissions beyond the scope of this license may be available at <http://arken.umb.no/~plessner>.



Computational Neuroscience



What is Computational Neuroscience?

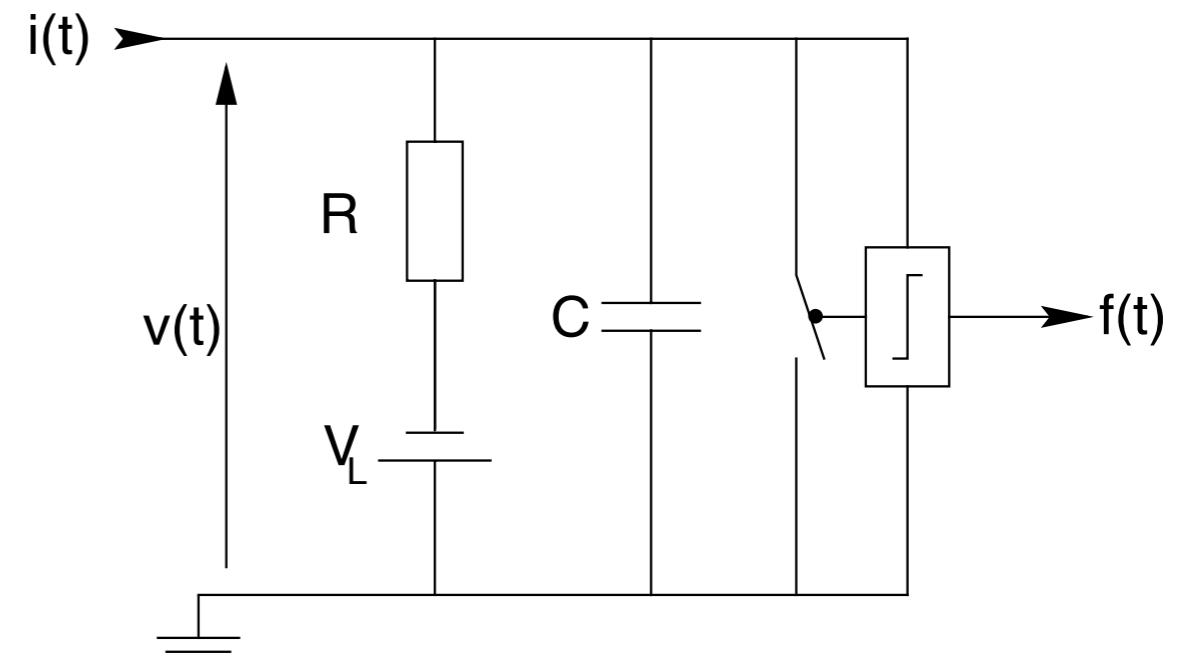
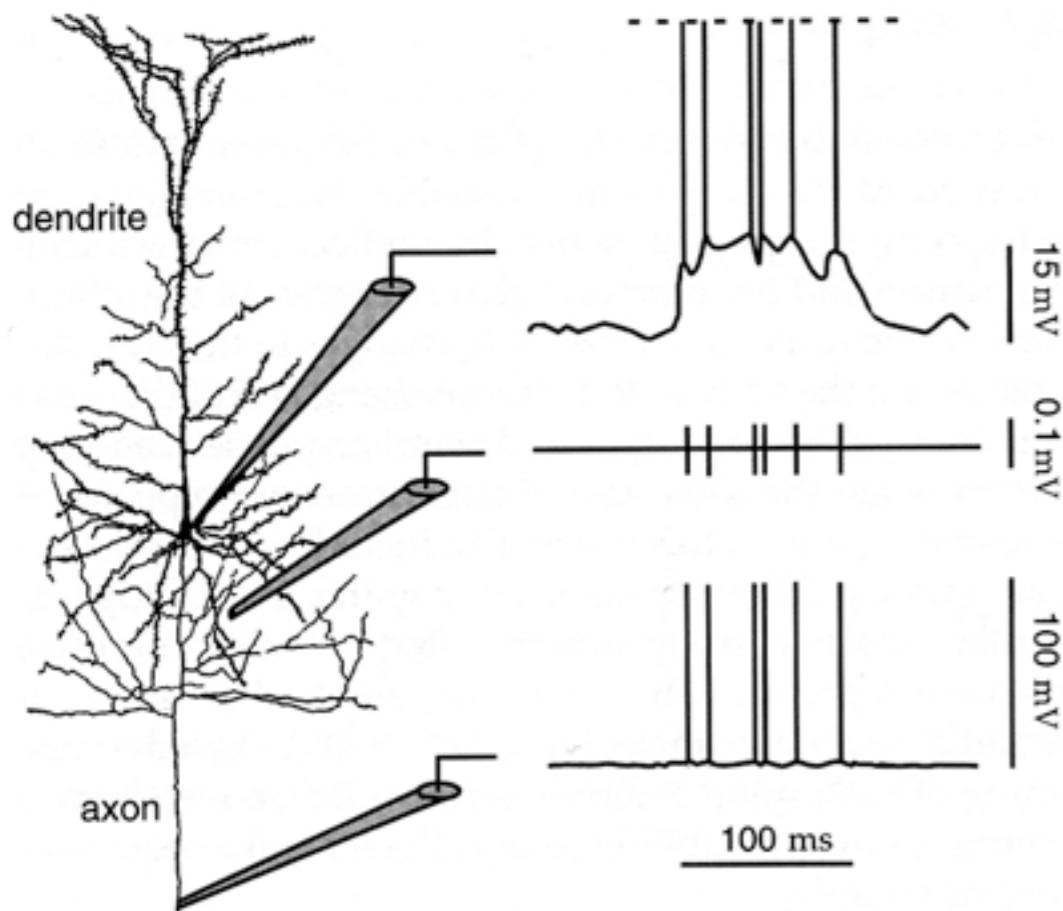


The goal of neural modeling is to relate, in nervous systems, function to structure on the basis of operation.

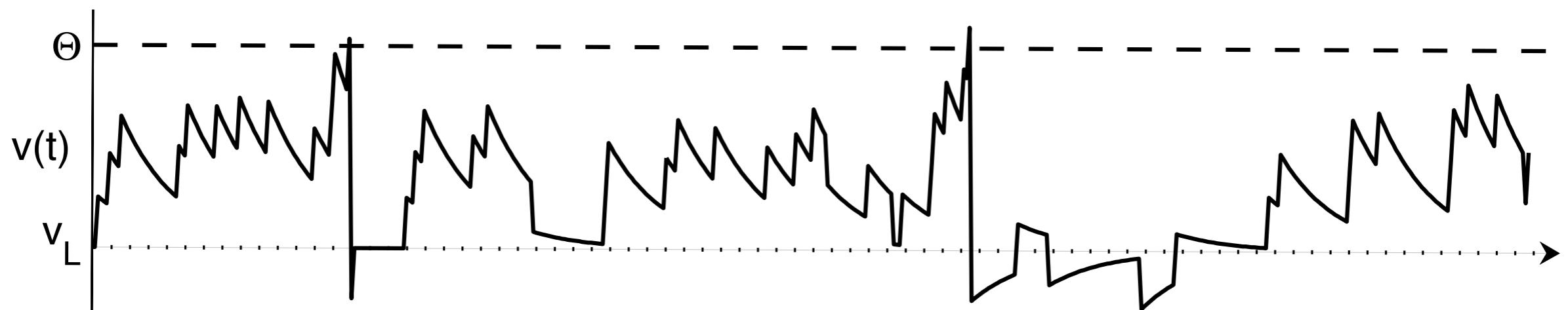
MacGregor & Lewis, 1977

Keeping it simple: point neurons

Complexity of the neuron ... abstracted as simple RC-circuit

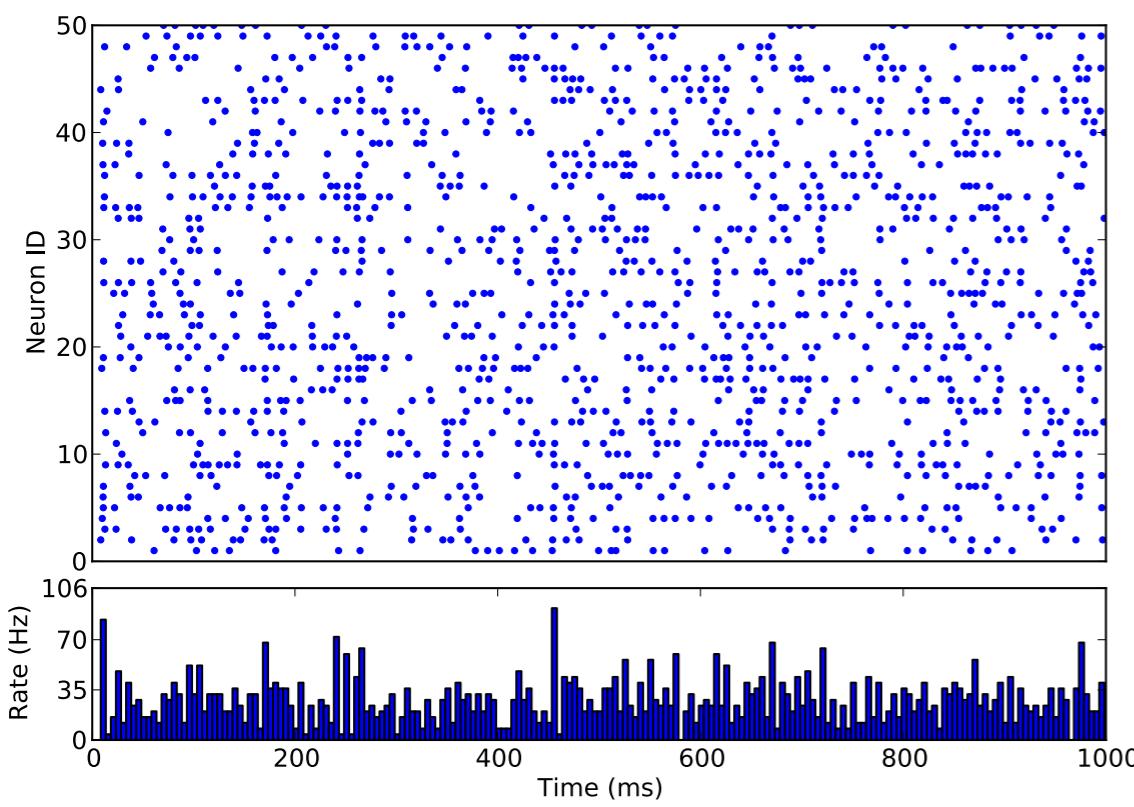
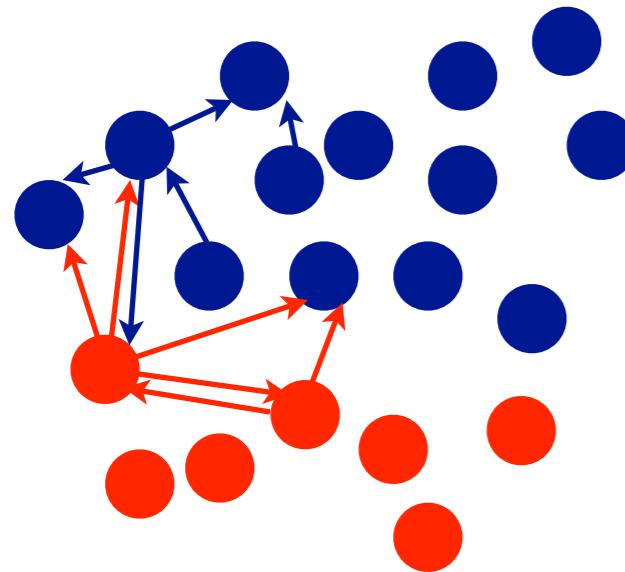


$$C\dot{V} = -\frac{(V - V_L)}{R} + i(t)$$

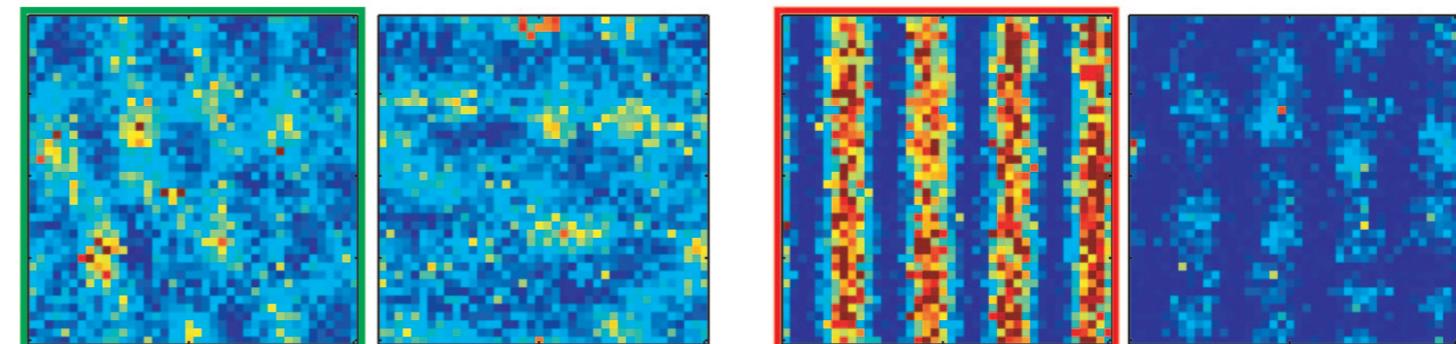
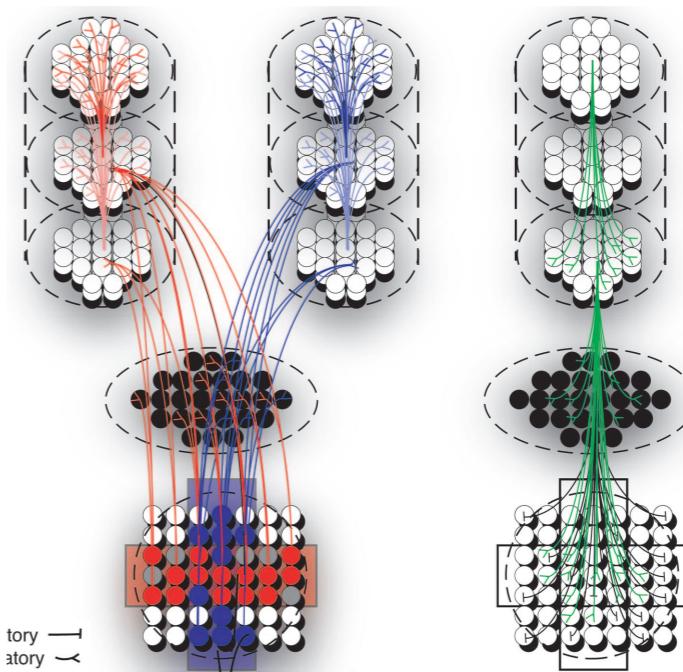


Then explore network dynamics!

Random network



Structured network



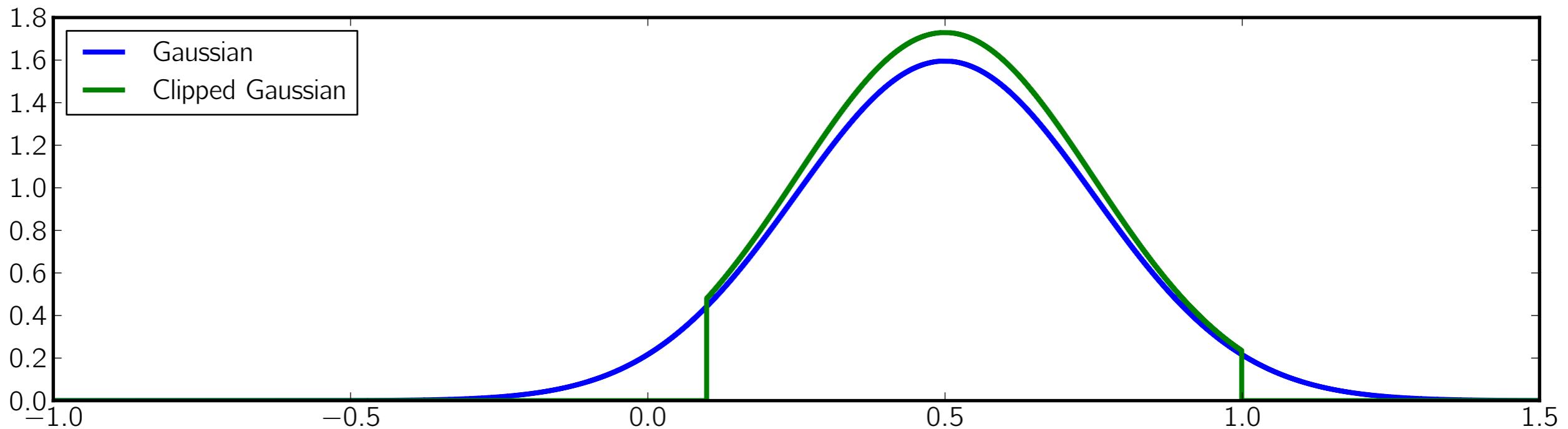
Hill & Tononi (2005)

Simulations are exciting — but reliable?

- Computational neuroscience
 - no conservation laws
 - no clear-cut separation of scales
 - no general agreement on which aspects of network activity are essential (spike rate vs spike time debate)
 - highly abstract models difficult to compare to experimental data quantitatively
- Highly dependent on *reliable* simulations
- Let's look at a real-life case

Case 2: The clipped Gaussian

- Well-known paper on plasticity
- Parameters chosen from Gaussian distribution, according to paper
- Results could not be reproduced independently
- Analysis of original C-code provided by authors:
 - ➡ Parameters were chosen from *clipped* Gaussian



The (sad) state of the art

- Few published results can be reproduced independently
- Authors often struggle to replicate their own results
- Systematic comparison and evaluation of models are rare
- Authors rarely discuss why and how models ended up as they are
- Models are seldom re-used

Reproduction *vs* Replication

Chris Drummond

Replicability is not Reproducibility: Nor is it Good Science

ICML Montreal 2009



Replication: necessary, difficult, & insufficient

- Replication
 - Re-create identical results
 - Essentially book-keeping
 - Requires tools & discipline
 - No new insights: tests implementation, not ideas
- *Internal replication*
Joe recreates results on original machine
- *External replication*
Jane recreates Joe's results on her machine using Joe's code
- *Cross replication*
Alice recreates Joe's results using a different simulator, based on a simulator-independent model description

How much detail does replication require?

- Very simple point-neuron model

$$\dot{V} = -\frac{V}{\tau} + \frac{I_E}{C}$$

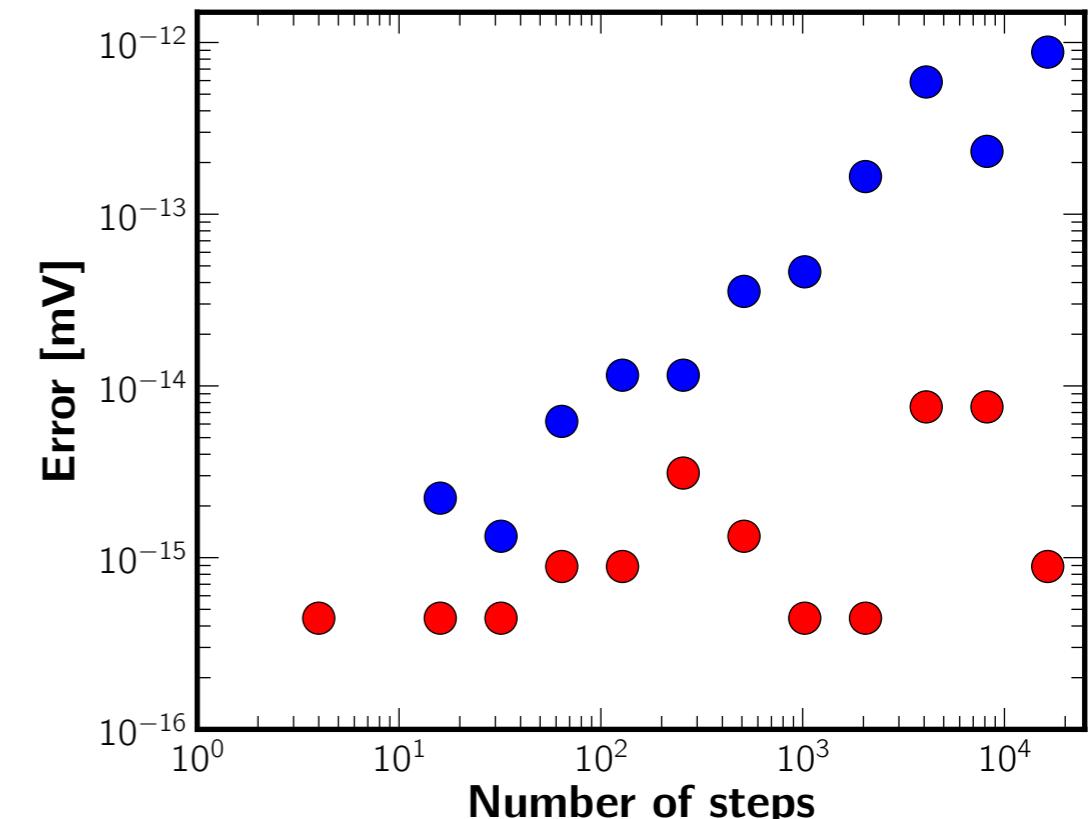
- Exact updating rule ($a = I_E \tau / C$)

$$V_{k+1} = V_k e^{-h/\tau} + a(1 - e^{-h/\tau})$$

- Two different implementations

`V[k+1] = v[k] * exp(-h/tau) - a * expm1(-h/tau)`

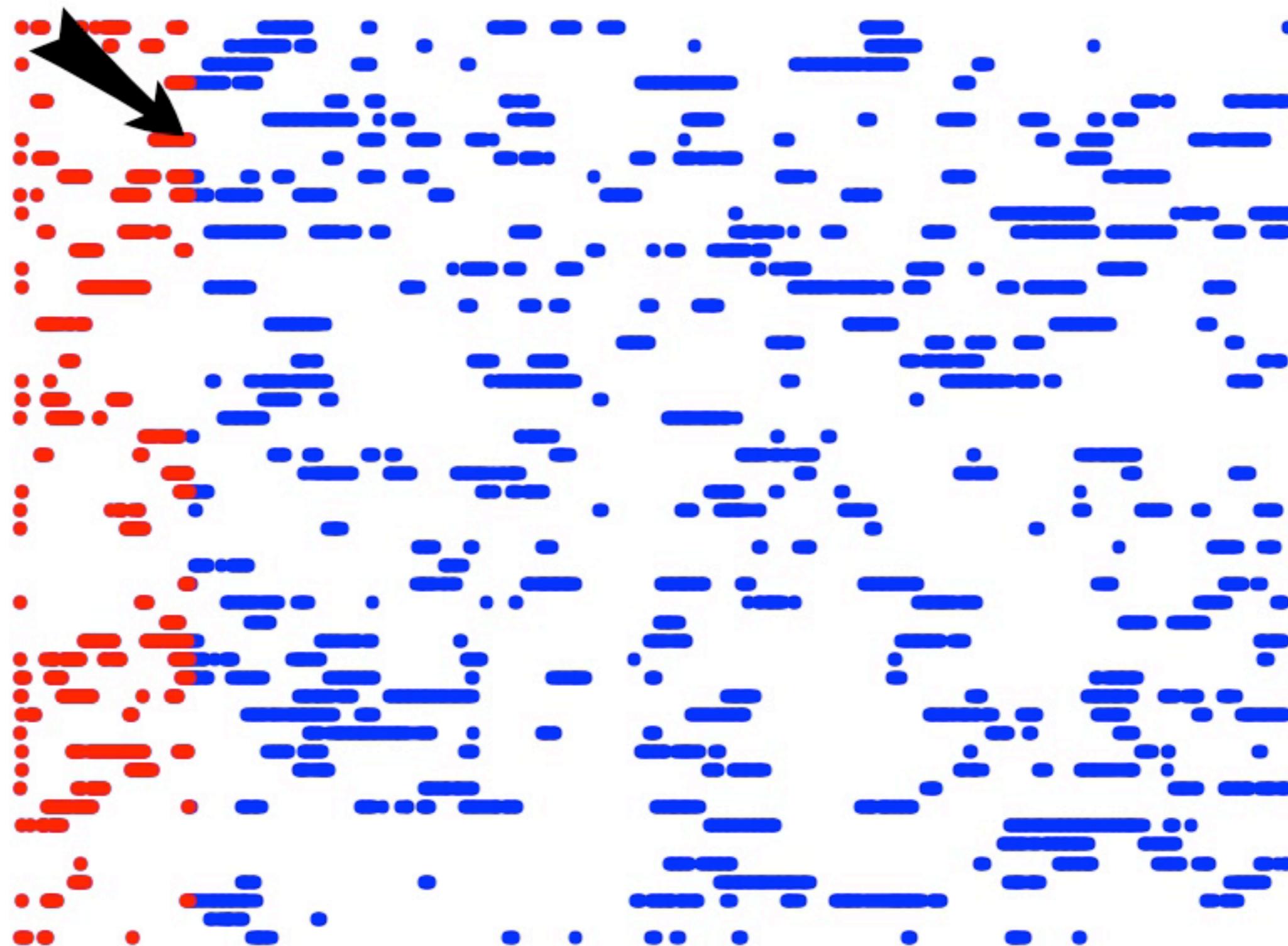
`V[k+1] = v[k] - (a - v[k]) * expm1(-h/tau)`



- Different numerical properties

→ Replication requires that we specify implementation!

But do such tiny differences matter?

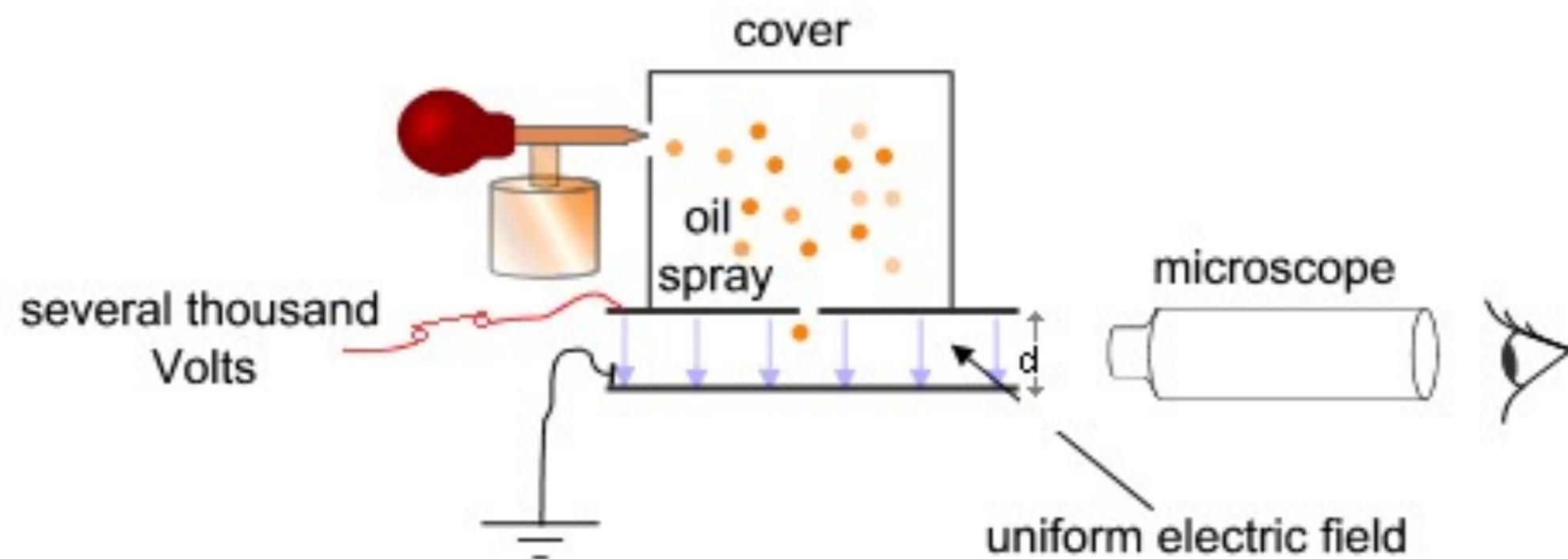


Reproduction

- Independent
- Test hypotheses and models
- Validates concepts
- Requires reflection

From oil drops to first-passage times

Millikan's oil drop experiment



Wikipedia

Erwin Schrödinger

Zur Theorie der Fall- und Steigversuche an Teilchen mit Brownscher Bewegung
Physikalische Zeitschrift 16:289 (1915)

Reproduction needs replication

- Reproducible models describe scientific ideas
 - Independent reproduction will generally fail to replicate original results precisely
 - Requires learned judgement of discrepancies
 - Requires means to understand failure
- Replicable implementation

Improving Scientific Practice in Computational Neuroscience



Good model description practice

- Systematic approach to model description in papers
- Standardize tables/checklists
- Standards for graphic representation of models
- Nordlie, Gewaltig, & Plesser
PLoS Comp Biol 5:e1000456 (2009)

A		Model Summary
Populations		Three: excitatory, inhibitory, external input
Topology		—
Connectivity		Random convergent connections
Neuron model		Leaky integrate-and-fire, fixed voltage threshold, fixed absolute refractory time (voltage clamp)
Channel models		—
Synapse model		α -shaped current inputs
Plasticity		—
Input		Independent fixed-rate Poisson spike trains to all neurons (during initial stimulation period)
Measurements		Spike activity

B		
Populations		
Name	Elements	Size
E	laf neuron	$N_E = 4N_I$
I	laf neuron	N_I
E_{ext}	Poisson generator	$C_E(N_E + N_I)$

C			
Connectivity			
Name	Source	Target	Pattern
EE	E	E	Random convergent $C_E \rightarrow 1$, weight J , delay D
IE	E	I	Random convergent $C_E \rightarrow 1$, weight J , delay D
EI	I	E	Random convergent $C_I \rightarrow 1$, weight $-gJ$, delay D
II	I	I	Random convergent $C_I \rightarrow 1$, weight $-gJ$, delay D
Ext	E_{ext}	E \cup I	Non-overlapping $C_E \rightarrow 1$, weight J , delay D

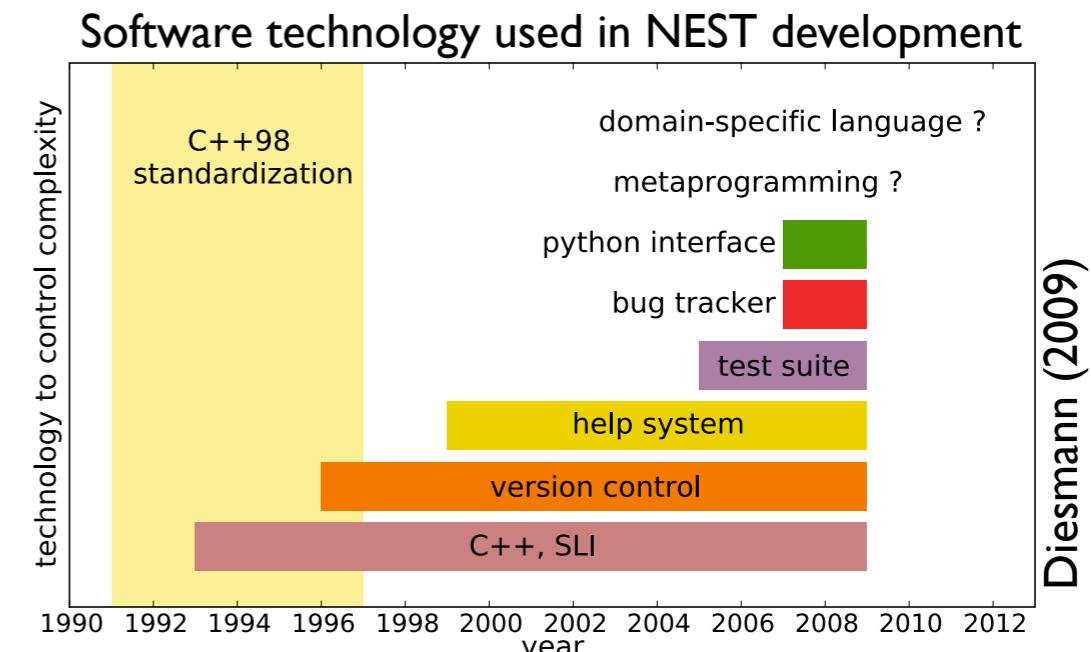
D		Neuron and Synapse Model	
Name		laf neuron	
Type		Leaky integrate-and-fire, α -current input	
Subthreshold dynamics		$\begin{aligned} \tau \dot{V}(t) &= -V(t) + RI(t) & \text{if} & \quad t > t^* + \tau_{\text{rp}} \\ V(t) &= V_r & \text{else} & \end{aligned}$ $I(t) = \frac{\tau}{R} \sum_{\tilde{t}} w \alpha(t - (\tilde{t} + \Delta)) \Theta(t - (\tilde{t} + \Delta))$	
Spiking		$\begin{aligned} \text{If } V(t-) < \theta \wedge V(t+) \geq \theta \\ 1. \text{ set } t^* = t \\ 2. \text{ emit spike with time-stamp } t^* \end{aligned}$	

E		Input
Type	Description	
Poisson generators		Fixed rate v_{ext} , C_E generators per neuron, each generator projects to one neuron; active only during initial stimulation period

F		Measurements
Spike activity as raster plots for subset of excitatory neurons		

Professional, shared software

- Widely used packages replace homemade *ad hoc* code
- Currently: Neuron, NEST, Genesis, Moose, Brian, PCSim
- “Social” developments
 - Simulator review (Brette *et al*, 2007)
 - Teaching software at summer schools
 - Large-scale *scientific* projects (eg EU FACETS)
 - Neuroinformatics journals
 - Raising awareness among reviewers and editors
- “Technical” developments
 - Version control
 - Test suites



ModelDB: Sharing models

- Curated database of computational neuroscience models
- Only published models
- Open to any software
- Nearly 700 models
- [http://
senselab.med.yale.edu/
modeldb/](http://senselab.med.yale.edu/modeldb/)

The screenshot shows a ModelDB model entry page. At the top, there's a header with the SenseLab logo and the ModelDB logo. Below the header, the title is "Sparsely connected networks of spiking neurons (Brunel 2000)". The accession number is 42020. A detailed description follows, mentioning the dynamics of networks of sparsely connected excitatory and inhibitory integrate-and-fire neurons, including various states like regular firing, asynchronous stationary global activity, and irregular individual cell activity. It also cites the paper by Brunel (2000). Below the description are sections for "Citations" and "Citation Browser". Under "Model Information", it lists the model type as "Connectionist Network", brain regions as "None", and various parameters like cell types, channels, gap junctions, receptors, genes, and transmitters, all listed as "None". Simulation environment is NEST (formerly BLISS/SYNOD), and model concepts include Activity Patterns, Oscillations, Spatio-temporal Activity Patterns, and Simplified Models. Implementer(s) are Gewaltig, Marc-Oliver. The "Model files" section shows a zip file download link and a "Help downloading and running models" link. Inside the zip file, there are files: README.txt, brunel, brunel.sli, and readme.txt (the latter being highlighted). The README.txt file contains the abstract from the Brunel (2000) paper and information about the brunel.sli file.

Sparingly connected networks of spiking neurons (Brunel 2000)

Accession: 42020

The dynamics of networks of sparsely connected excitatory and inhibitory integrate-and-fire neurons are studied analytically (and with simulations). The analysis reveals a rich repertoire of states, including synchronous states in which neurons fire regularly; asynchronous states with stationary global activity and very irregular individual cell activity; and states in which the global activity oscillates but individual cells fire irregularly, typically at rates lower than the global oscillation frequency. See paper for more and details.

Reference: Brunel N (2000) Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J Comput Neurosci* 8:183-208 [PubMed]

Citations Citation Browser

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

Model Type: [Connectionist Network](#);

Brain Region(s)/Organism:

Cell Type(s):

Channel(s):

Gap Junctions:

Receptor(s):

Gene(s):

Transmitter(s):

Simulation Environment: [NEST \(formerly BLISS/SYNOD\)](#);

Model Concept(s): [Activity Patterns](#); [Oscillations](#); [Spatio-temporal Activity Patterns](#); [Simplified Models](#);

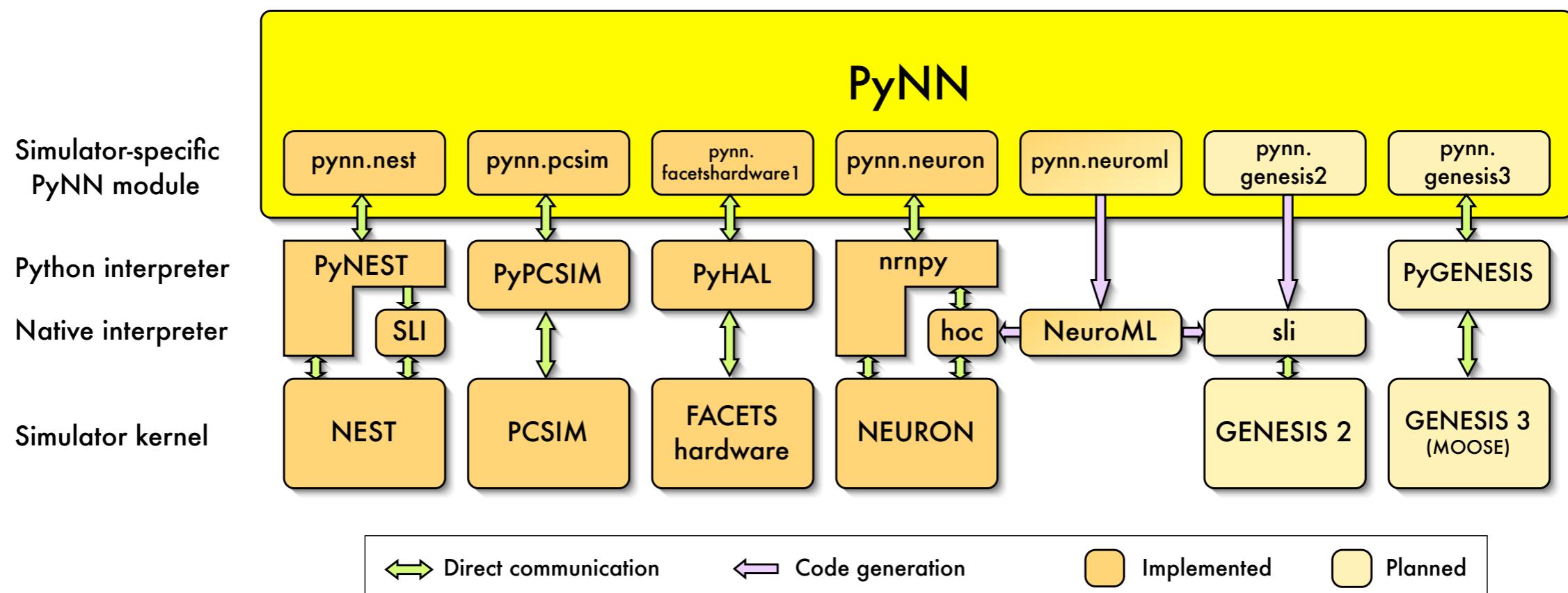
Implementer(s): [Gewaltig, Marc-Oliver](#);

Model files [Download zip file](#) [Help downloading and running models](#)

brunel readme.txt brunel.sli	Readme.txt for an implementation of the model associated with the paper: Brunel N (2000) Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. <i>J Comput Neurosci</i> 8:183-208 The brunel.sli file was supplied by Marc-Oliver Gewaltig and runs under NEST. Please contact Marc-Oliver Gewaltig marc-oliver.gewaltig@honda-ri.de for more information.
--	--

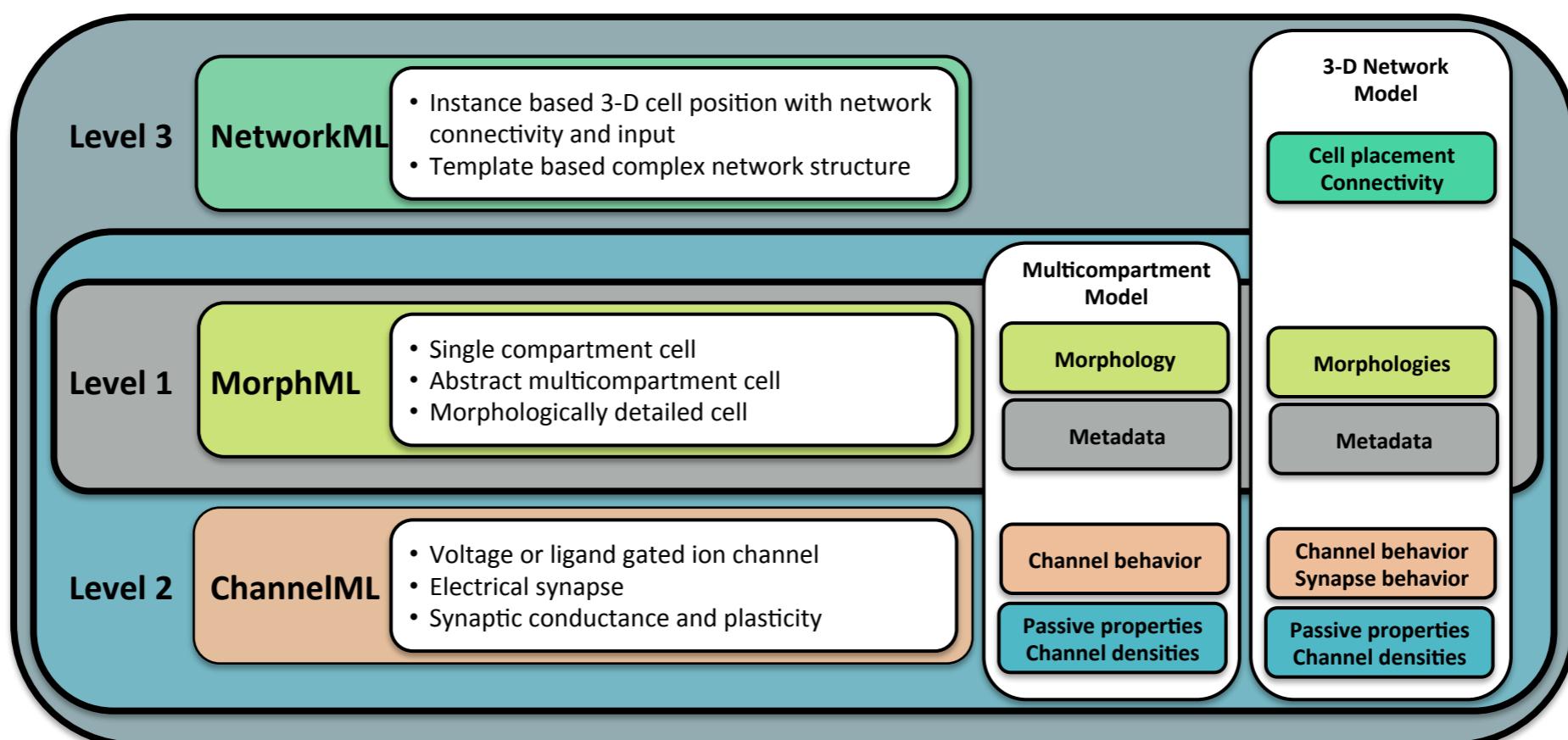
PyNN: A Meta-Simulator

- Python-based wrapper for many simulators
- Write model and simulation code once, run on all
- Facilitates model sharing and cross-validation
- Developed by Andrew Davison for FACETS project
- <http://neuralensemble.org/PyNN>



NeuroML: A model specification language

- XML-based language for model specification
- Multiple layers: channels, neuron morphologies, networks
- Code-generation for several simulators, including PyNN
- Facilitates model sharing and re-use
- <http://www.neuroml.org>



Provenance tracking: Sumatra

- Python package to enable systematic capture of the environment of numerical simulations/analyses
- Tracks simulation code, dependencies, platform information, results
- Developed by Andrew Davison as part of FACETS project
- <http://neuralensemble.org/sumatra>

The screenshot shows a web browser window titled "Sumatra: TestProject: List of records". The address bar displays "http://127.0.0.1:8002/". The main content area is titled "TestProject: List of records". A table lists four records. The columns are: Delete (checkbox), Label, Reason, Outcome, Duration, Processes, Simulator (Name, Version), Script (Repository, Main file, Version), Date, Time, and Tags.

Delete include data	Label	Reason	Outcome	Duration	Processes	Simulator		Script			Date	Time	Tags
						Name	Version	Repository	Main file	Version			
<input type="checkbox"/>	20100709-154255		'Eureka! Nobel prize here we come.'	0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:42:55	
<input type="checkbox"/>	20100709-154309			0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:43:09	
<input type="checkbox"/>	haggling	'determine whether the gourd is worth 3 or 4 shekels'	'apparently, it is worth NaN shekels.'	0.59 s		Python	2.5.2	/Users/andrew/tmp/SumatraTest	main.py	396c2020ca50	09/07/2010	15:43:20	foobar

NineML: A model description standard

- Aims for community standard for declarative model descriptions
- Inspired by SBML and CellML
- Focus on networks of point neurons
- Under development by INCF Multi-scale Modeling Task Force

Perspectives

- Community increasingly aware of need for reproducibility and replicability
 - Large-scale projects have led to development of valuable tools
 - Summer schools educate PhD-students and post-docs in use of established modeling tools
 - Neuroinformatics journals allow publication of domain-specific solutions
 - International Neuroinformatics Coordinating Facility (INCF) stimulates debate and development
 - NEST Initiative is devoted to furthering reliable simulations
- We have a long way to go, but we are (finally) moving!

Collaborators



Sharon Crook

School of Mathematical and Statistical Sciences, School of Life Sciences, Center for Adaptive Neural Systems
Arizona State University



Andrew Davison

Unité de neuroscience, information et complexité
CNRS
Gif-sur-Yvette



Eilen Nordlie



Marc-Oliver Gewaltig

Honda Research Institute Europe
Offenbach



Research Council of Norway
(eVita program)