IBM Research Division

A Scalable Solution for Specifying and Solving Arbitrary Dense Neural Tissue Model Graphs in a Domain Decomposition

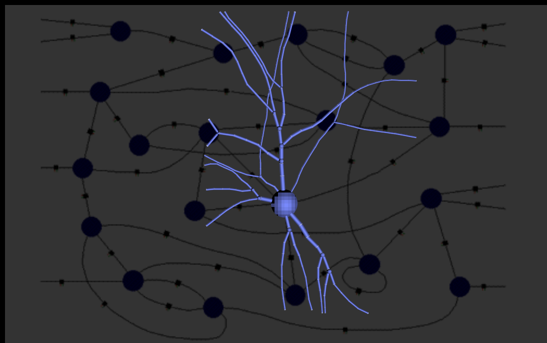
Plans and Implications for I/O Bound Applications and Analysis

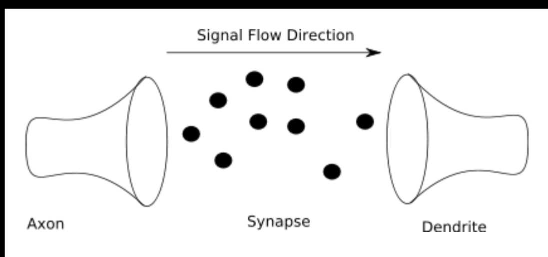
Presentation to BGAS Workshop, Jülich Supercomputing Centre

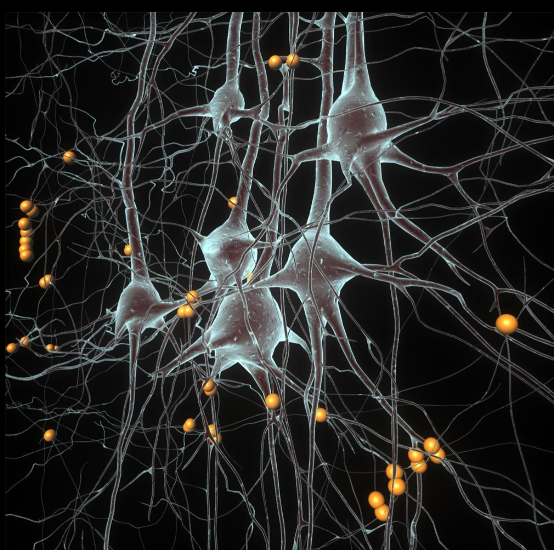
James Kozloski, kozloski@us.ibm.com
T. J. Watson Research Center, Yorktown Heights, NY

Overview

- Neural Tissue Simulator
- Model Graphs
  - Model Definition
  - Graph Specification
- Scaling
- Neural Current Analyzer





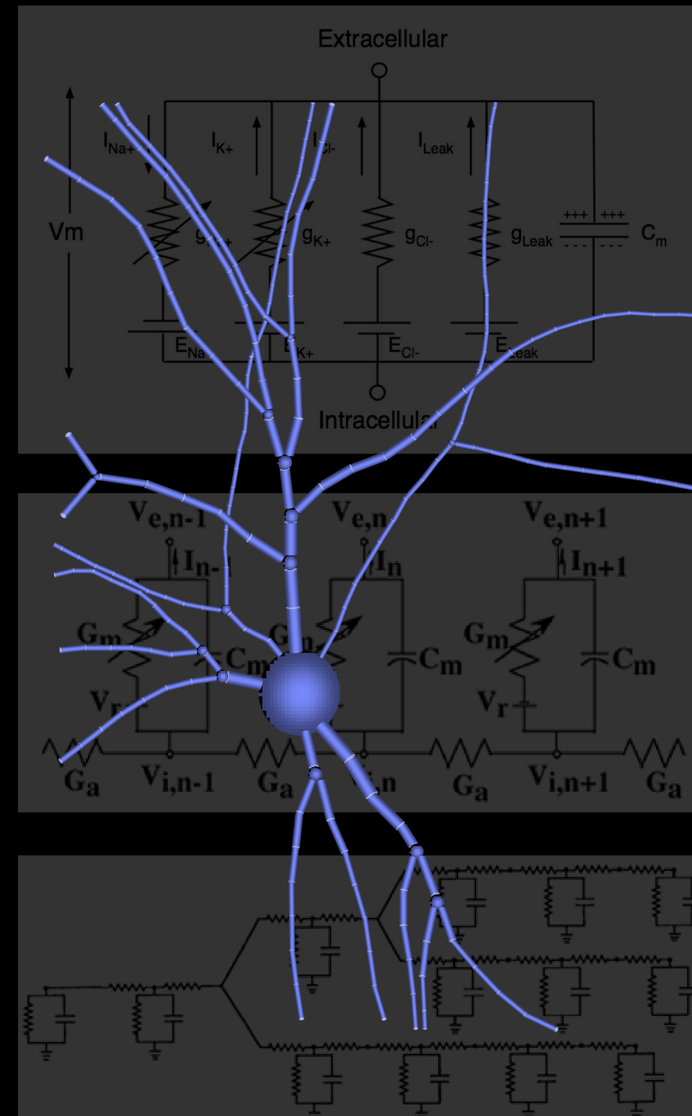


"Neural Tissue Simulation"

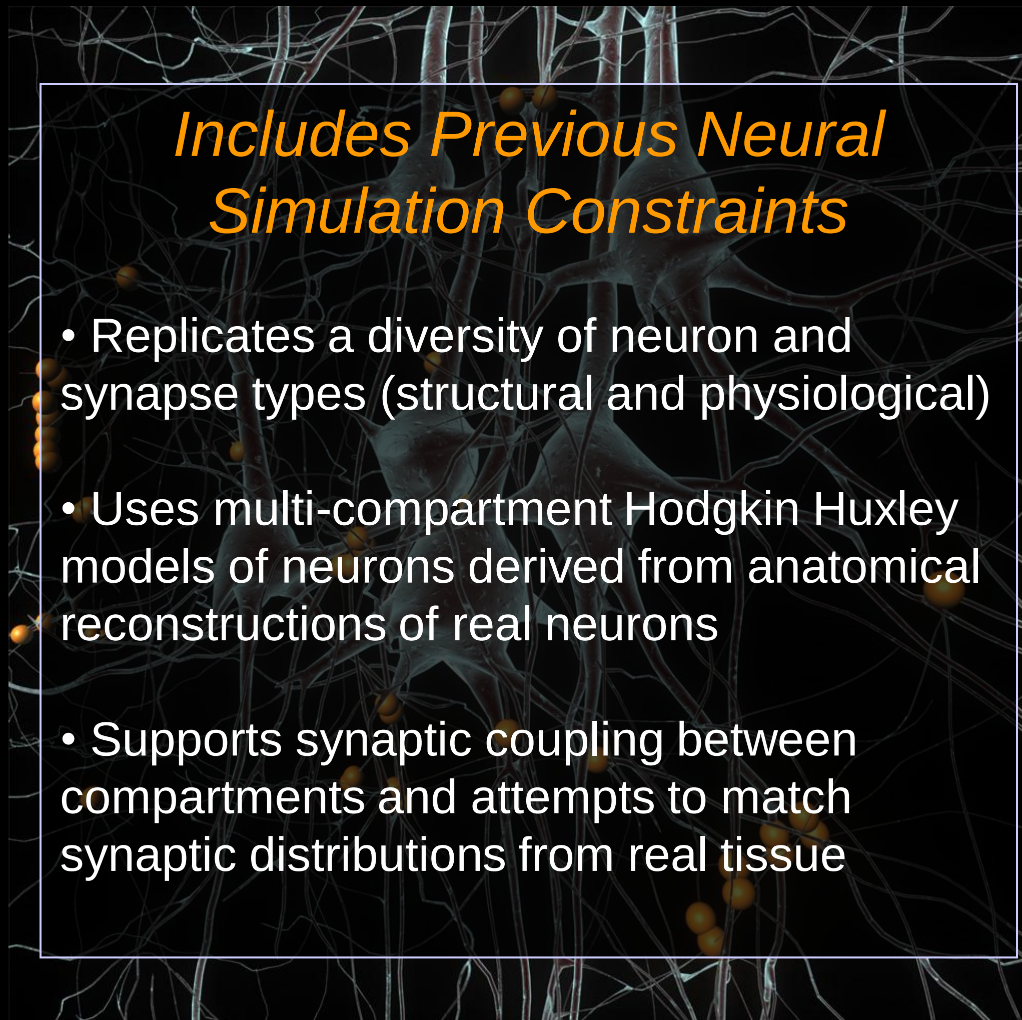
- Single-compartment models of Hodgkin and Huxley
- Multi-compartment models of single fibers
- Multi-compartment models of branched neuronal arbors
- Multi-compartment models of whole neurons

- Neural Networks
- Coupling trans-neuronal compartments
  - Models of synaptic release and receptors
  - Models of gap junctions
- Biological Neural Networks
- Neural Tissue Simulation

?



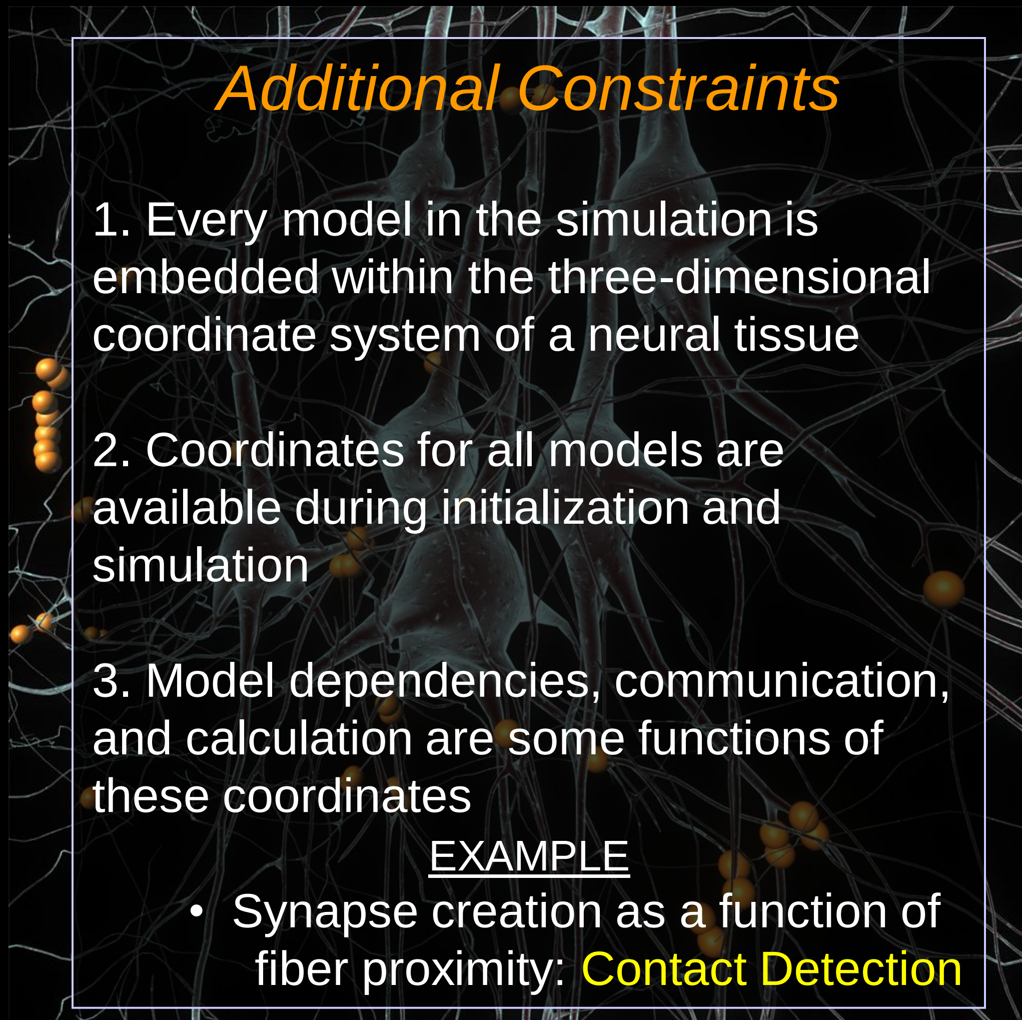
Neural Tissue Simulation



Includes Previous Neural Simulation Constraints

- Replicates a diversity of neuron and synapse types (structural and physiological)
- Uses multi-compartment Hodgkin Huxley models of neurons derived from anatomical reconstructions of real neurons
- Supports synaptic coupling between compartments and attempts to match synaptic distributions from real tissue

Neural Tissue Simulation



Additional Constraints

1. Every model in the simulation is embedded within the three-dimensional coordinate system of a neural tissue

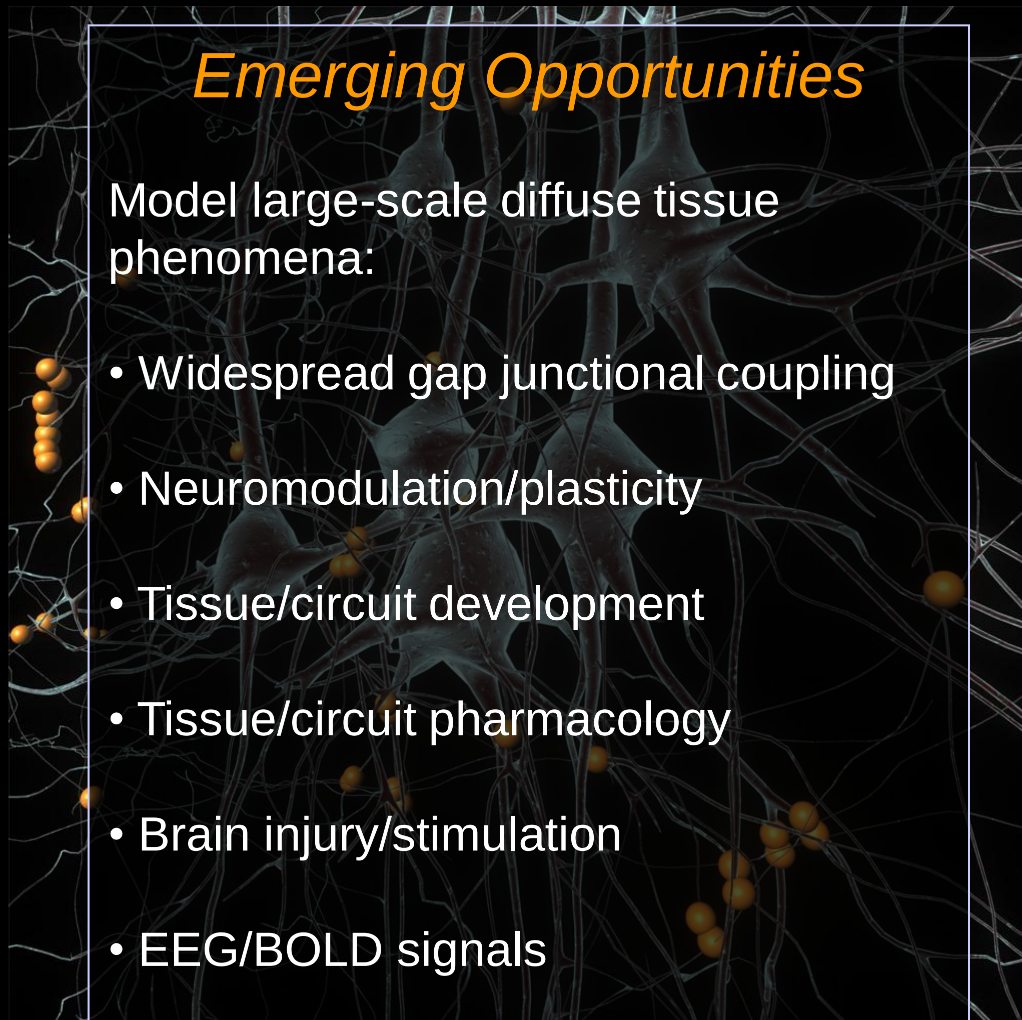
2. Coordinates for all models are available during initialization and simulation

3. Model dependencies, communication, and calculation are some functions of these coordinates

EXAMPLE

- Synapse creation as a function of fiber proximity: Contact Detection

Neural Tissue Simulation

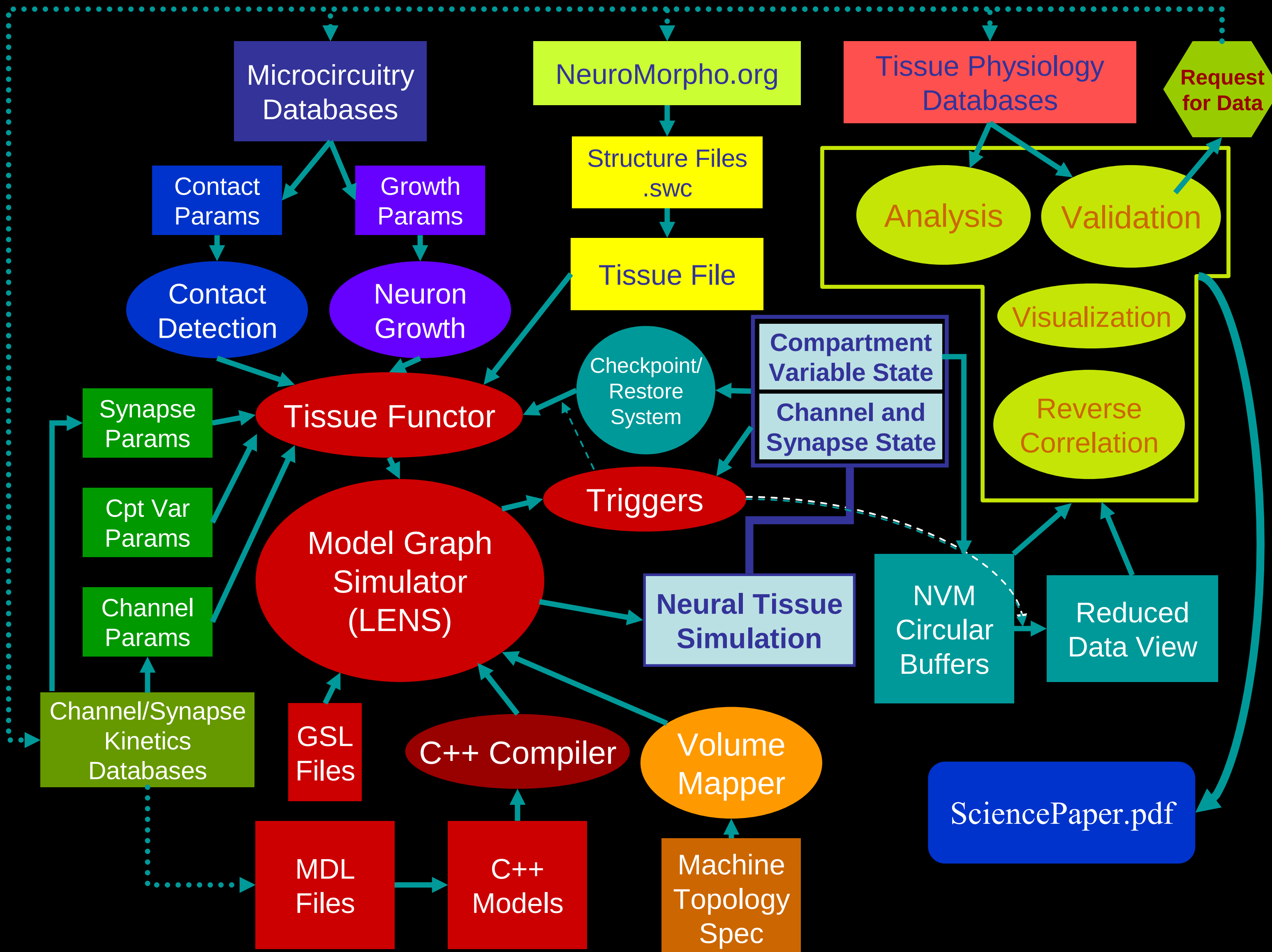


Emerging Opportunities

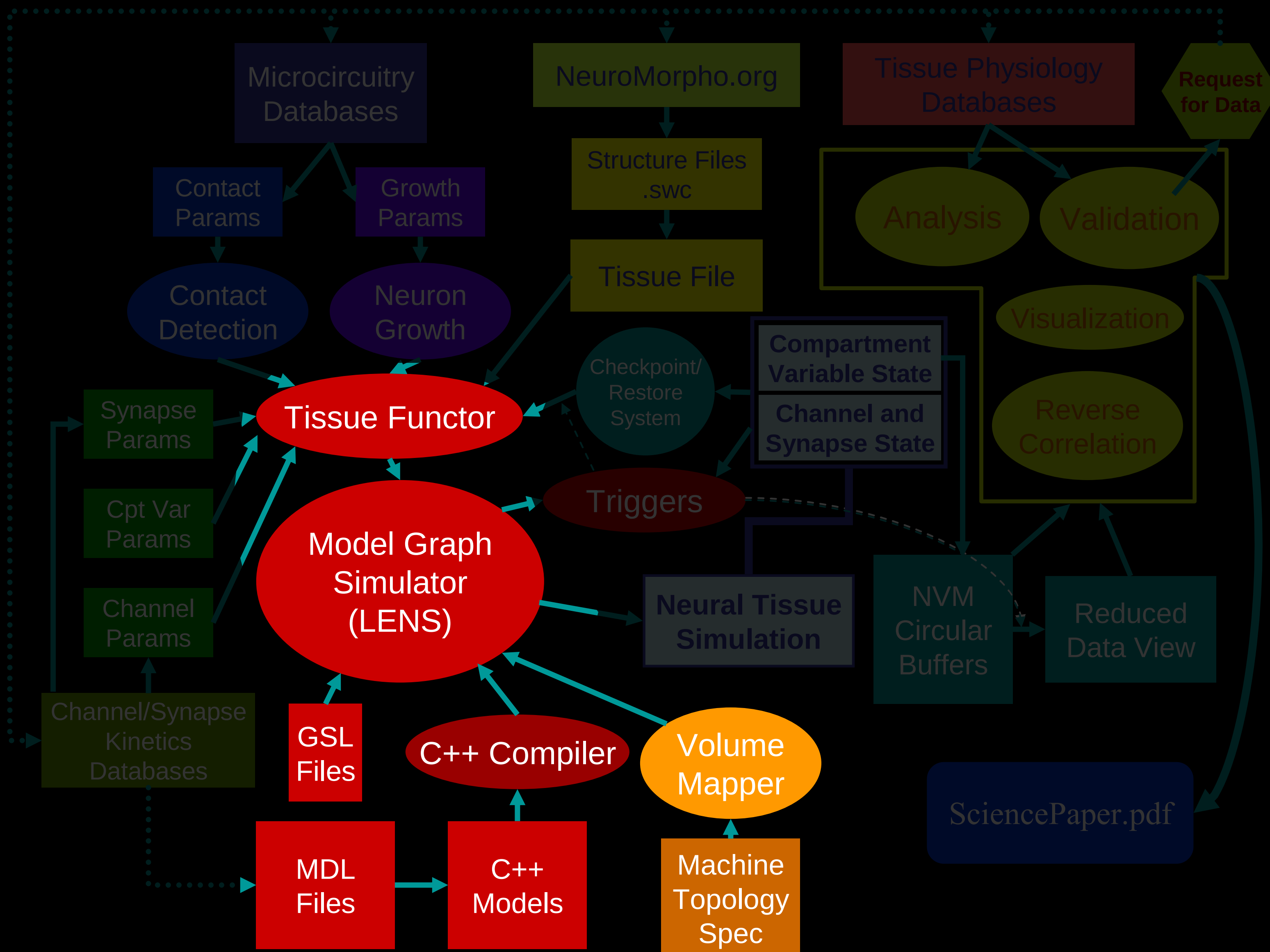
Model large-scale diffuse tissue phenomena:

- Widespread gap junctional coupling
- Neuromodulation/plasticity
- Tissue/circuit development
- Tissue/circuit pharmacology
- Brain injury/stimulation
- EEG/BOLD signals

Neural Tissue Simulation Workflow



Neural Tissue Simulation Workflow

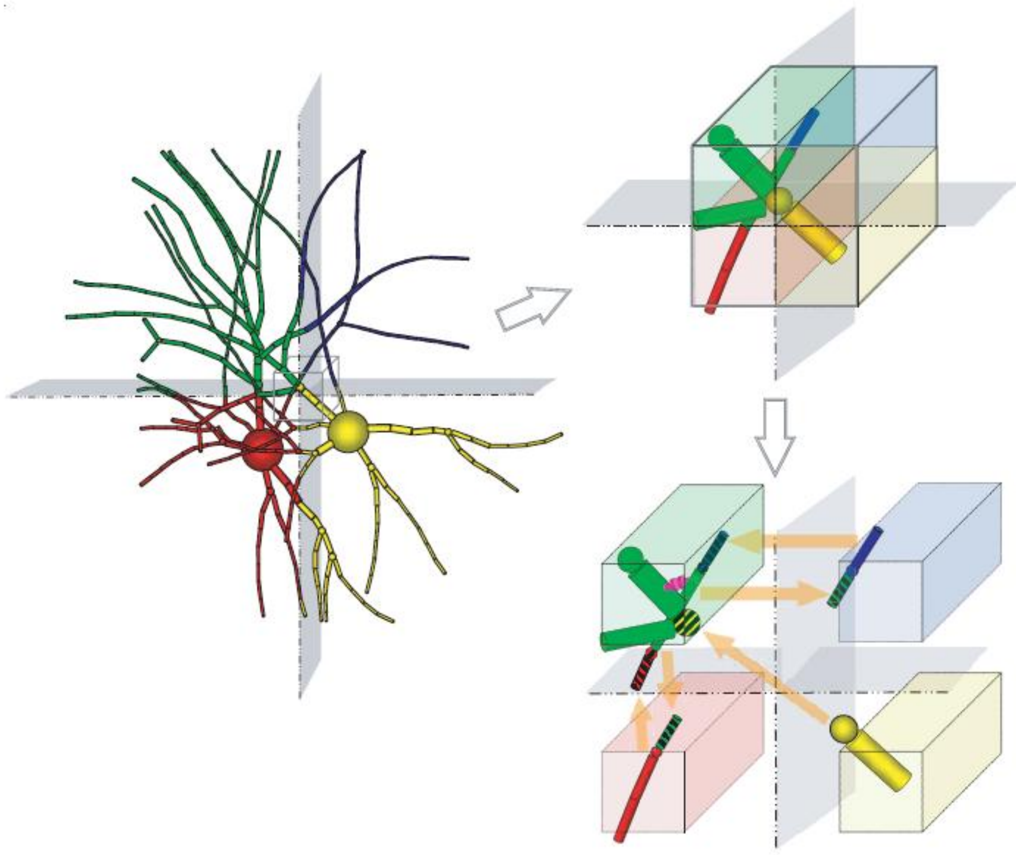


Model Graph Simulator: Infrastructure

Architectural Overview

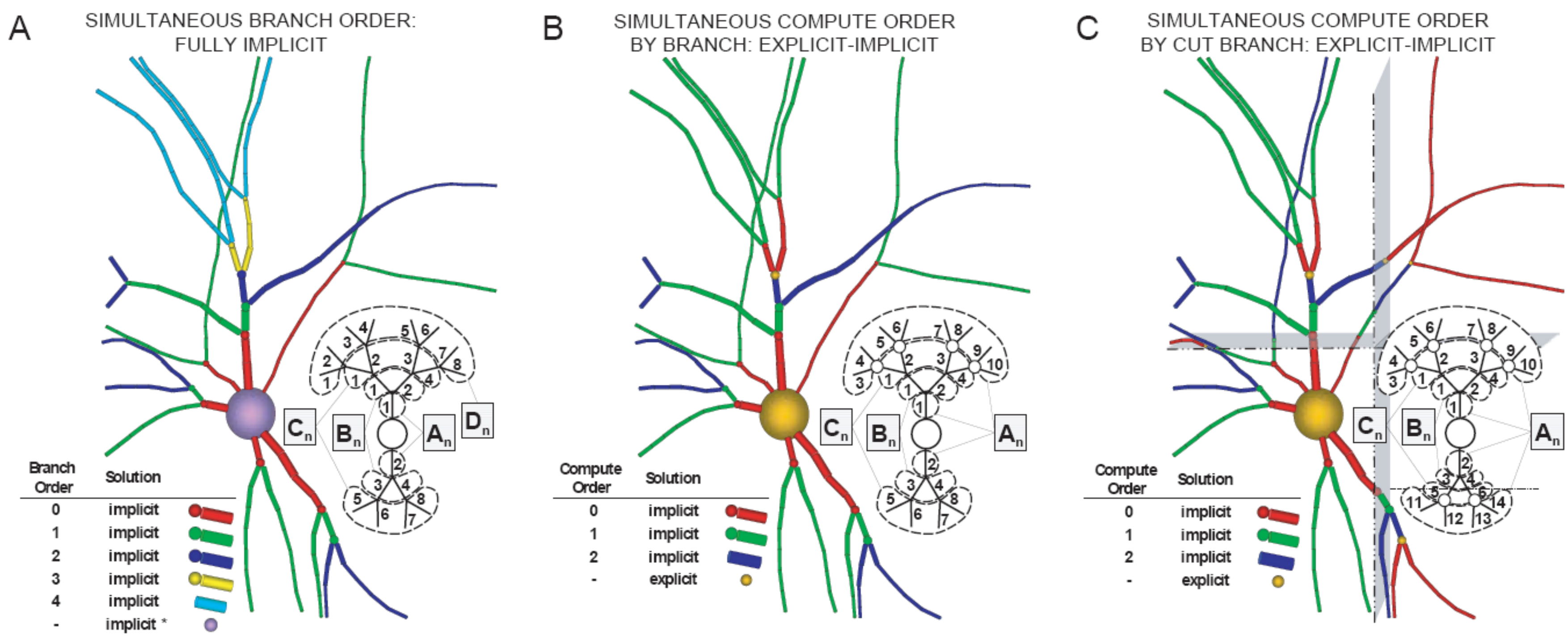
- Language for expressing model state, computational phases, communicated state, and model interfaces (MDL)
- Language for composing arbitrary parameterized graphs (GSL)
- Automatic partitioning into work units for multi-threaded execution (SMP)
- Dynamically constructed, simulation-specific MPI collective communication for multi-process execution of computational phases (MPP)

Model Graph Simulator: Tissue Volume Decomposition



Novel Numerical Approach

- John Wagner, Manager IBM Research Australia /Computational Biology Co-laboratory

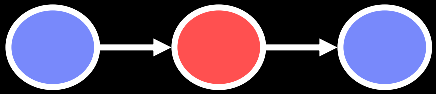


Model Definition: Phases

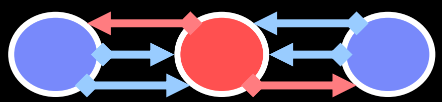
```
InitPhases = { initialize };
RuntimePhases = { run1, run2, run3, run4, run5, run6 };

NodeType HHJunction { predictState->run1,
                      correctState->run6 };

NodeType HHBranch { forwardEliminateCO0->run2,
                    forwardEliminateCO1->run3,
                    backSubstituteCO1->run4,
                    backSubstituteCO0->run5 };
```

Time Step

Multiphase Algorithm



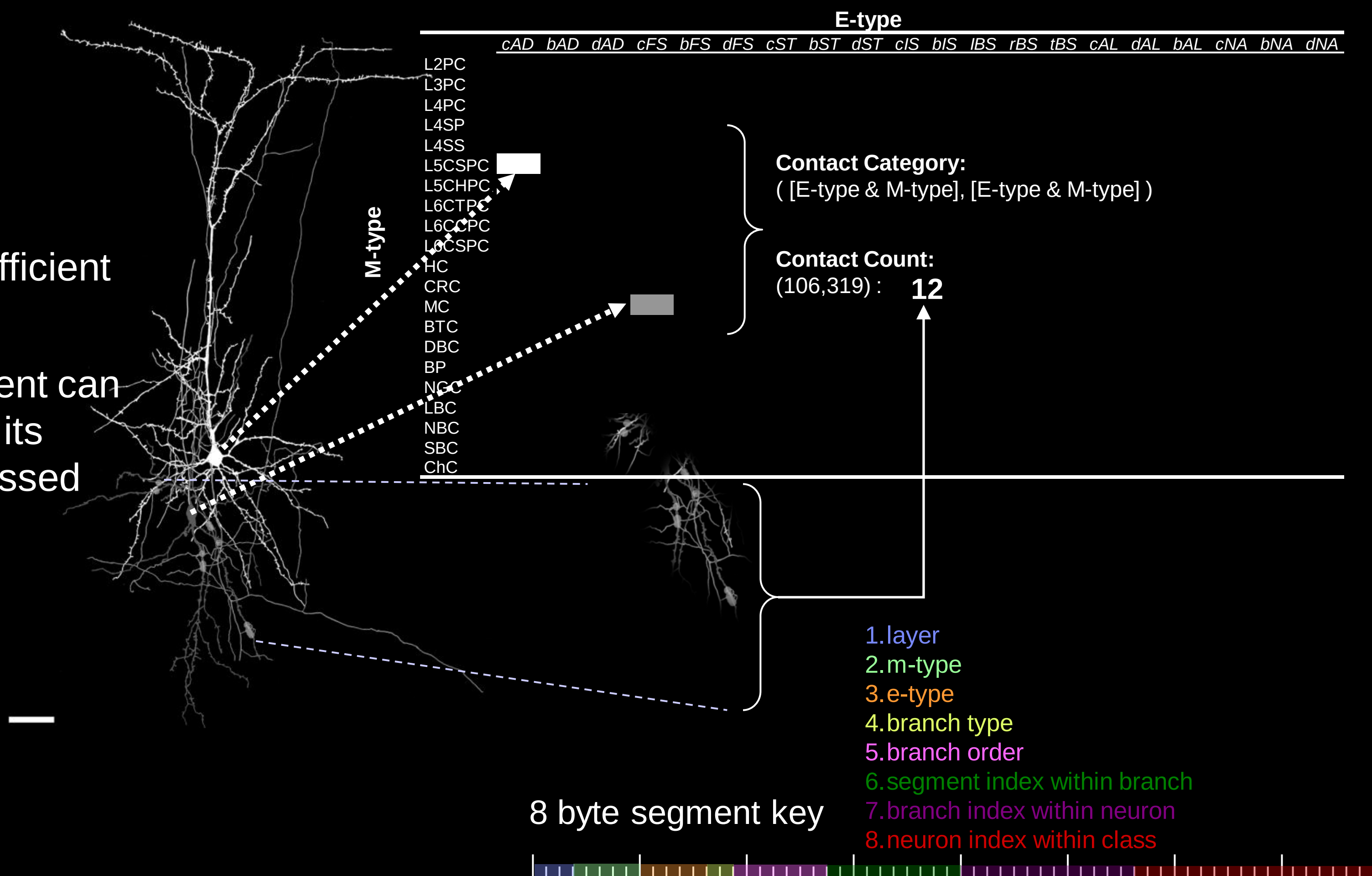
Model Definition: Interfaces



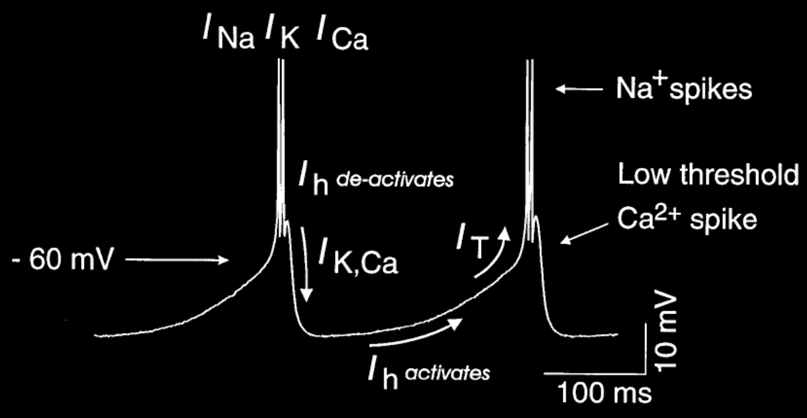
```
Node NaChannel Implements ConductanceArrayProducer, ReversalPotentialProducer
{
 double [] m;
 double [] h;
 double [] g;
 double [] gbar;
 double []* V;

 Connection Pre Node (PSet.identifier=="compartment") Expects VoltageArrayProducer
 {
  VoltageArrayProducer.voltageArray >> V;
 }

 Connection Pre Node (PSet.identifier=="IC") Expects NaConcentrationProducer {
  NaConcentrationProducer.Na >> Shared.Na_IC;
 }
```

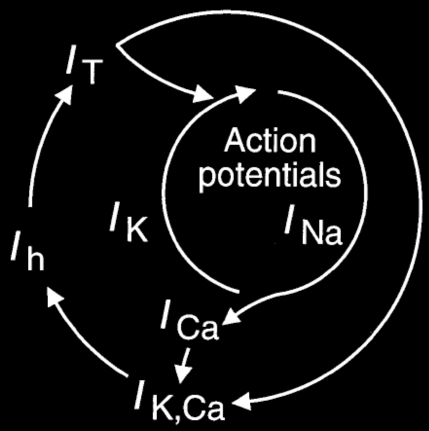
Graph Specification: Key Component Identities

- Each segment describes itself
- Segment key is compressed for efficient communication
- With key, a segment can be identified, and its neuron data accessed

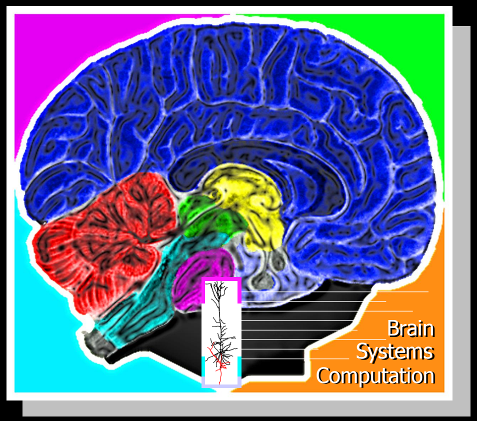


Modeling Objective: Inferior Olive



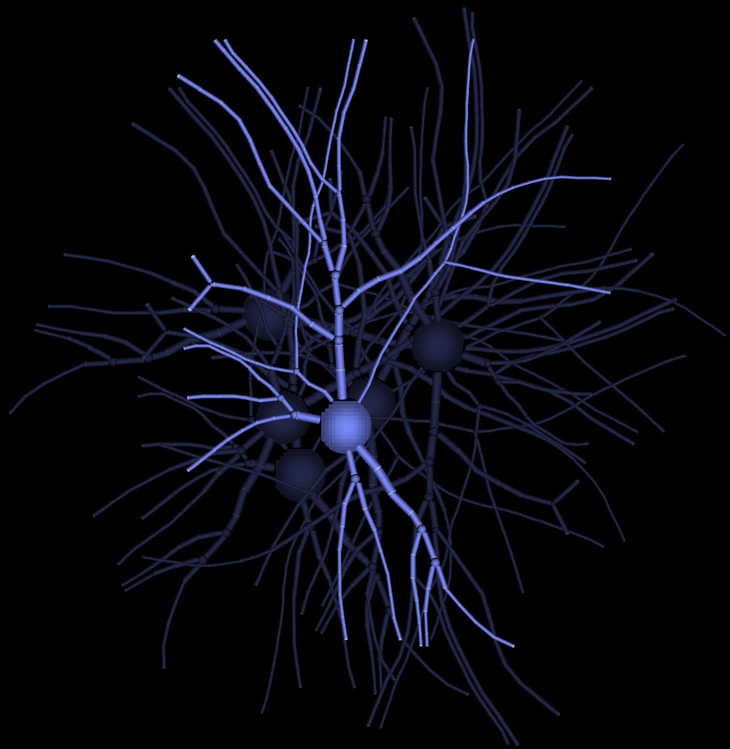


- Oscillations generated by intrinsic interplay between membrane currents
- Subthreshold oscillations are not driven by spike *input*, but instead constrain and drive spike *output*

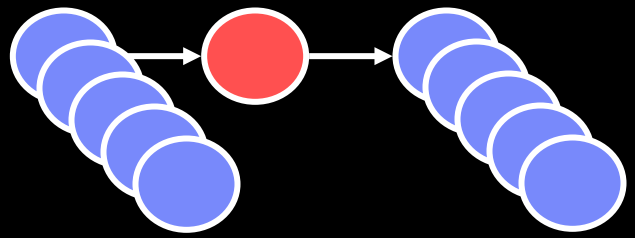


T. Bal and D. McCormick, "Synchronized oscillations in the Inferior Olive are controlled by the hyperpolarization-activated cation current I_h", J. Neurophysiol. 77:3145-3156, 1997.

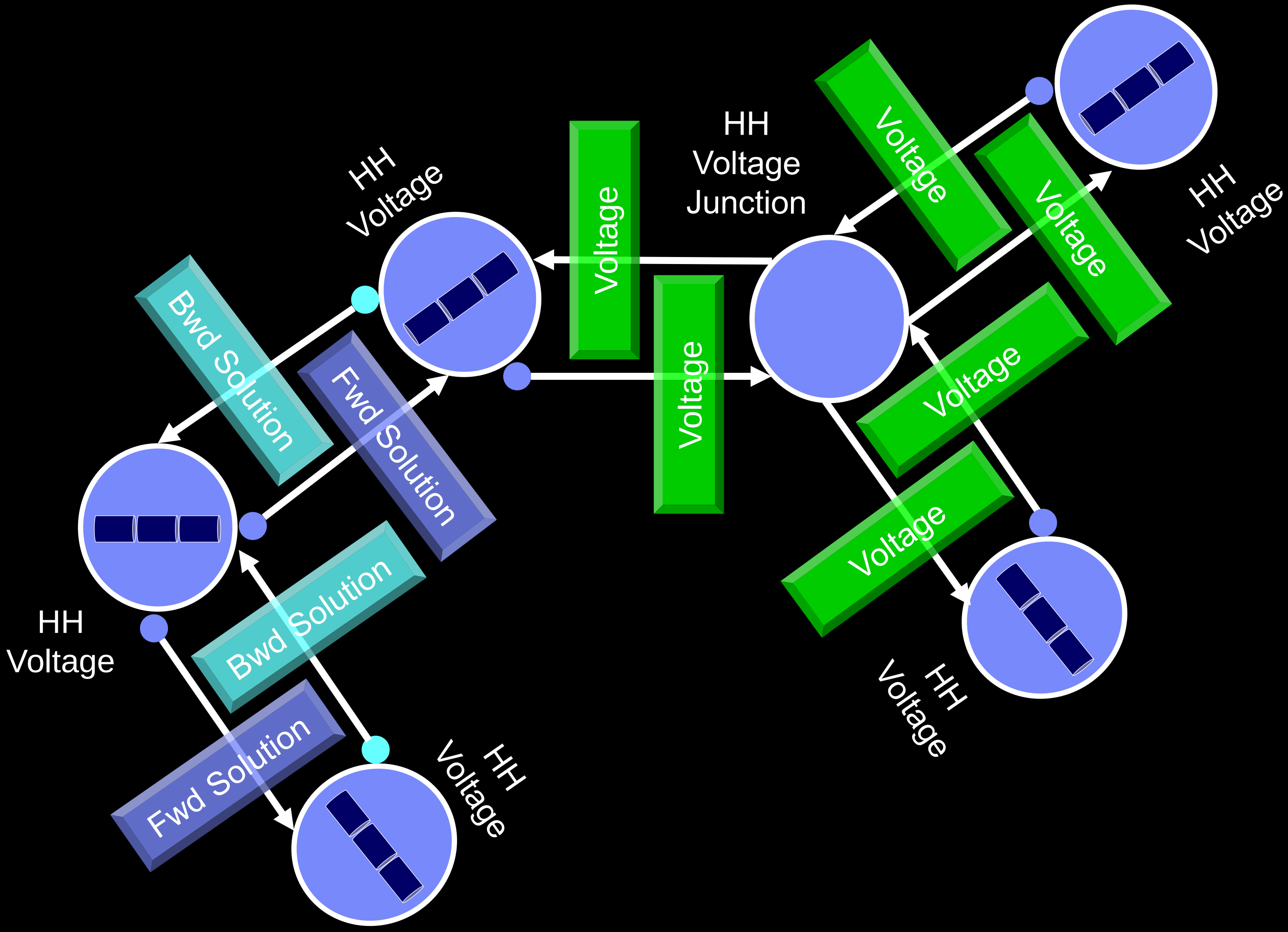
Modeling Calcium Dynamics in IO neurons



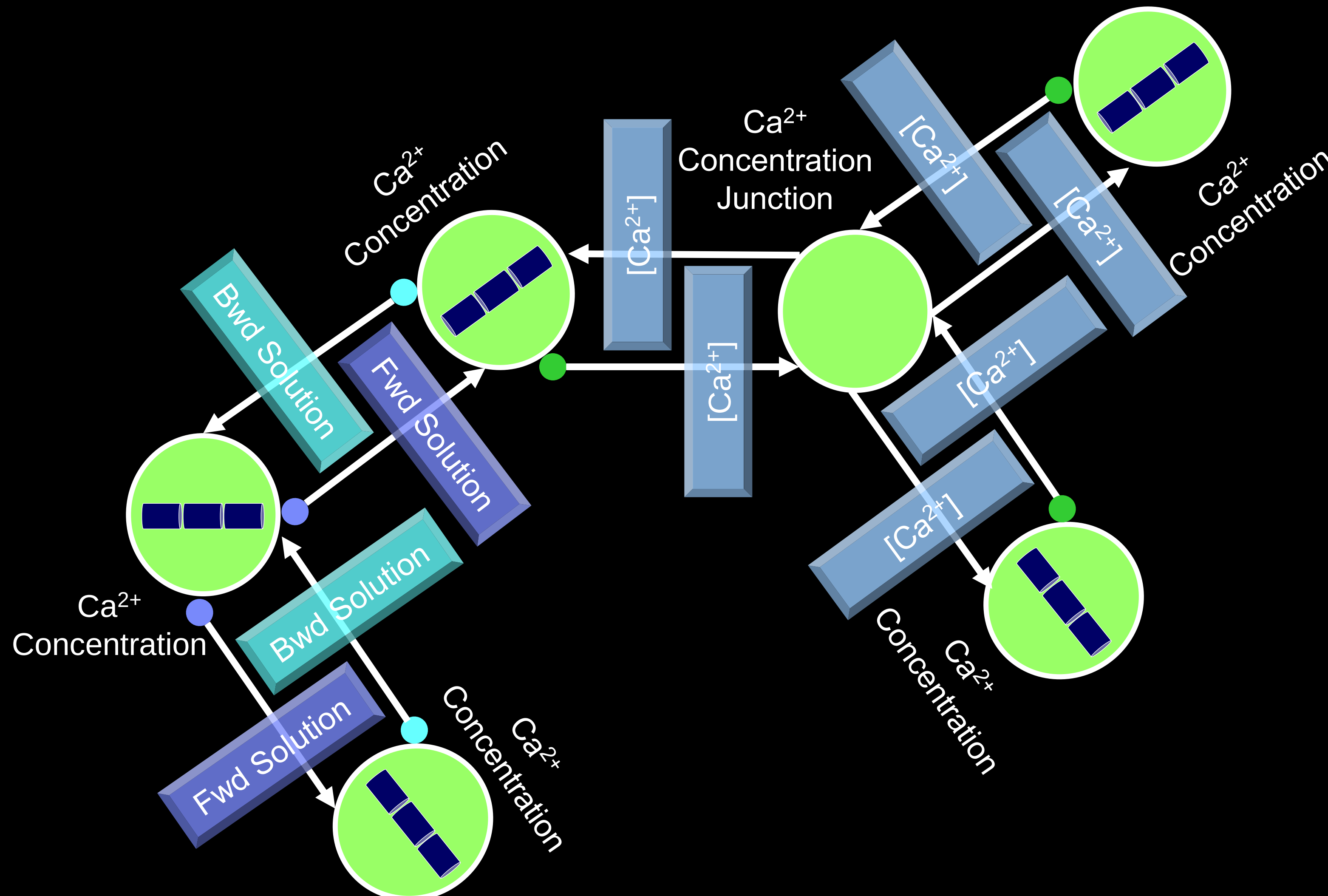
```
InitPhases = { initialize };
RuntimePhases = { run1, run2, run3, run4, run5, run6 };

NodeType HHVoltageJunction { predictState->run1,
                             correctState->run6 };

NodeType HHVoltage { forwardEliminateCO0->run2,
                     forwardEliminateCO1->run3,
                     backSubstituteCO1->run4,
                     backSubstituteCO0->run5 };

NodeType CaConcentration { forwardEliminateCO0->run2,
                           forwardEliminateCO1->run3,
                           backSubstituteCO1->run4,
                           backSubstituteCO0->run5 };
```



Graph View: Hybrid Voltage Solver



Graph View: Hybrid Calcium Solver

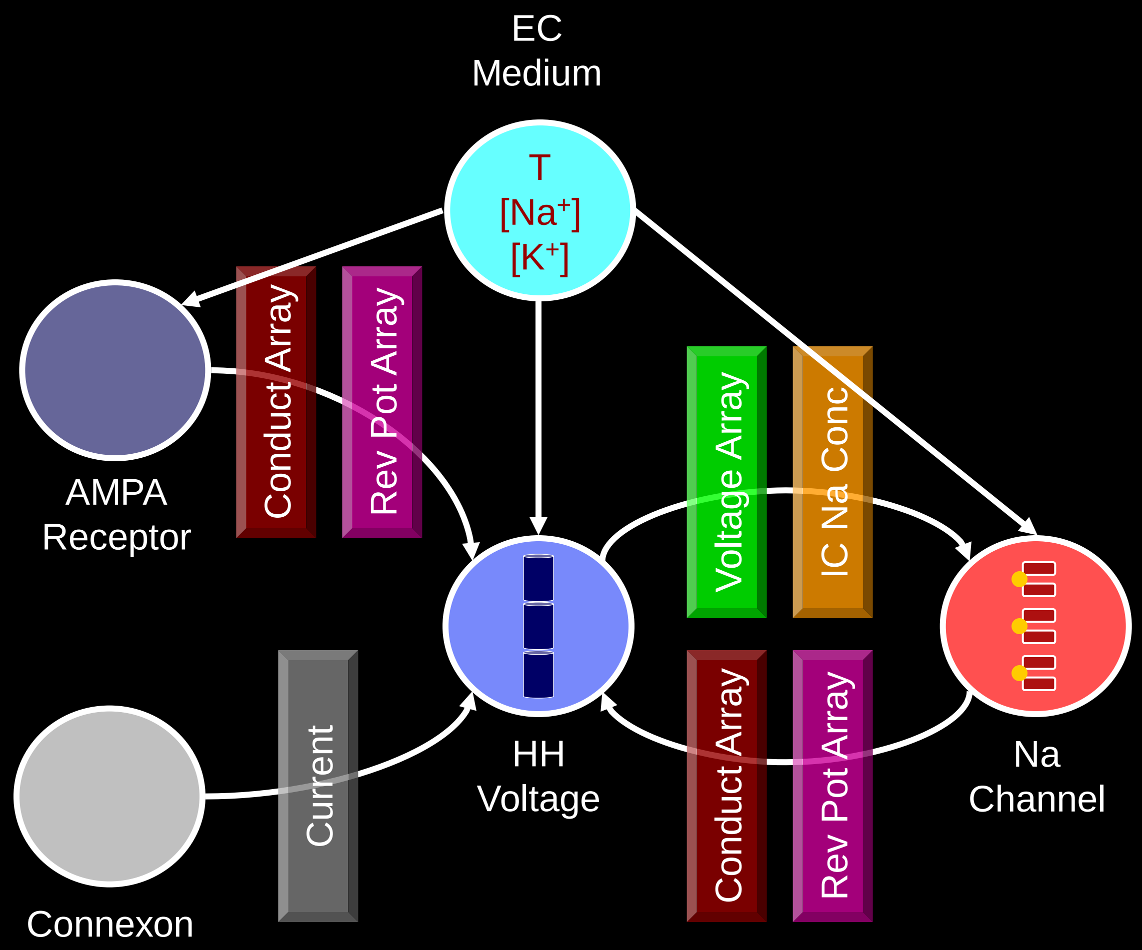


Graph Specification: Compartment Variables

COMPARTMENT_VARIABLE_TARGETS 4
BRANCHTYPE
0 Voltage, Calcium
1 Voltage
2 Voltage, Calcium
3 Voltage, Calcium

COMPARTMENT_VARIABLE_COSTS 2
Voltage 1.0
Calcium 0.95

Graph View: Synapses and Channels



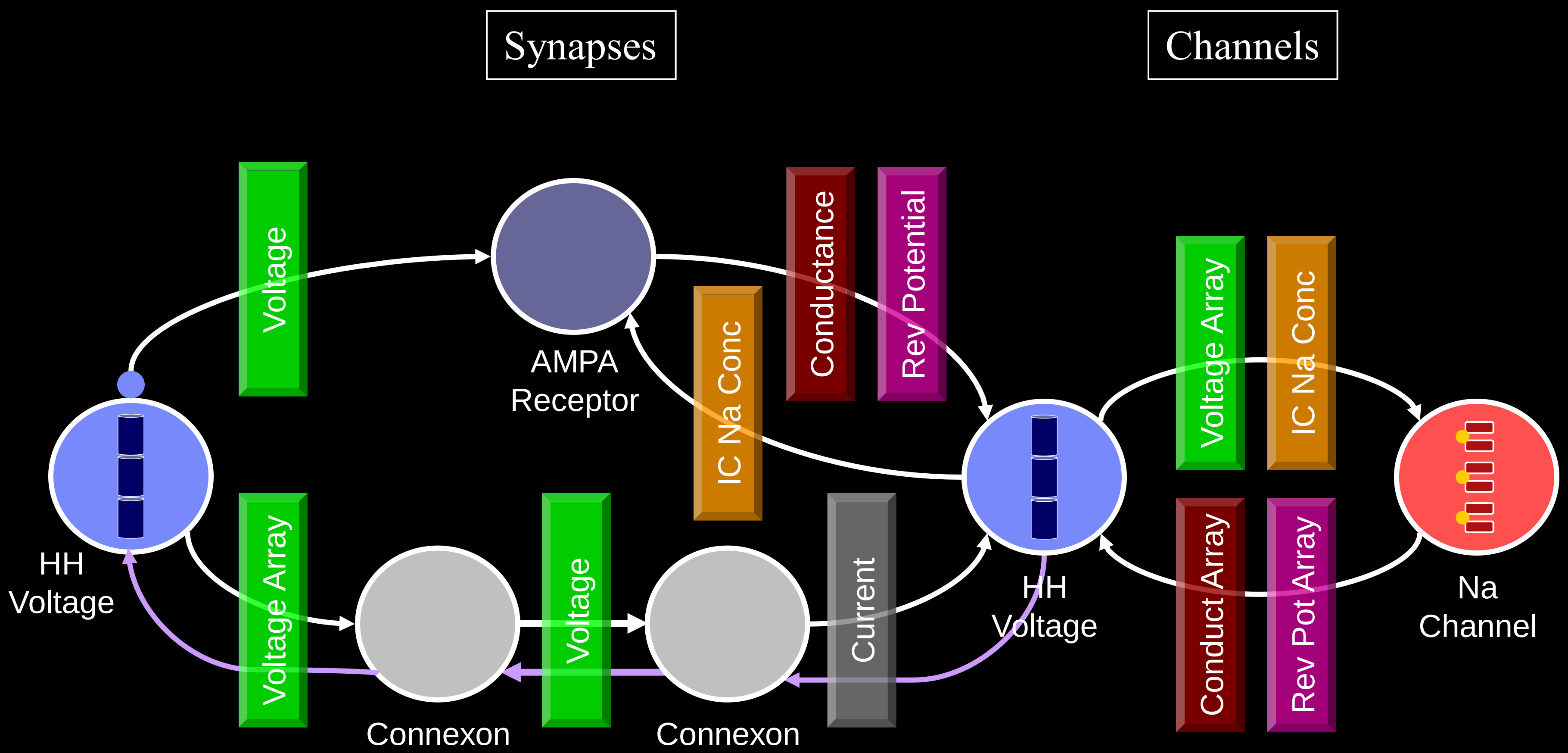
Graph Specification: Channels

CHANNEL_TARGETS 4
BRANCHTYPE
0 Na [Voltage] [Voltage], KDR [Voltage] [Voltage], Cah [Voltage] [Voltage, Calcium], KCa [Calcium] [Voltage]
1 Na [Voltage] [Voltage], KDR [Voltage] [Voltage]
2 Cah [Voltage] [Voltage, Calcium], KCa [Calcium] [Voltage]
3 Cah [Voltage] [Voltage, Calcium], KCa [Calcium] [Voltage]

CHANNEL_COSTS 4
Na 0.414243
KDR 0.254051
Cah 0.414243
KCa 0.359252

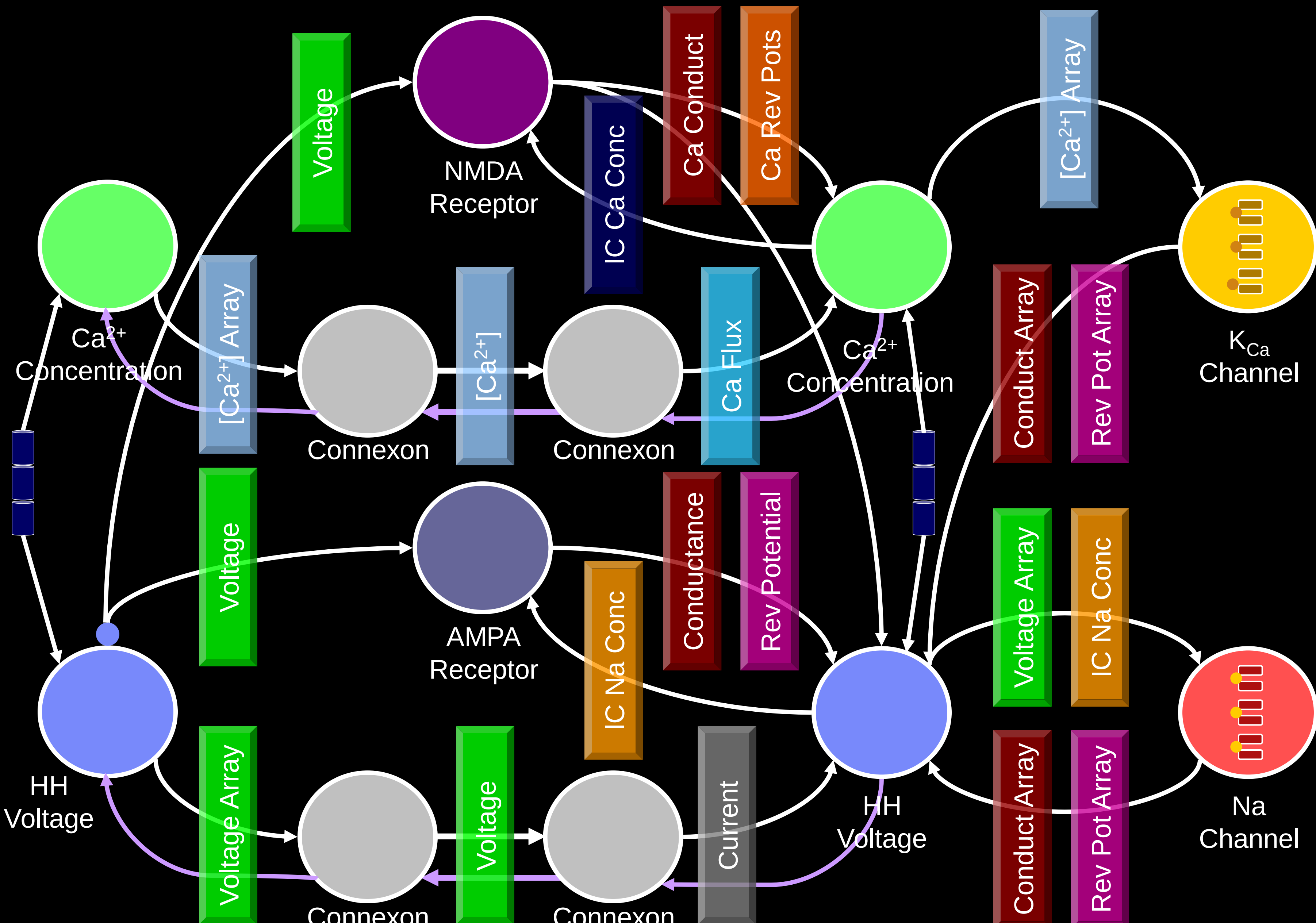
CHANNEL_PARAMS 2
Na 2
BRANCHTYPE
0 <gbar={0.7}>
1 <gbar={1.8}>

Graph View: Synapses and Channels

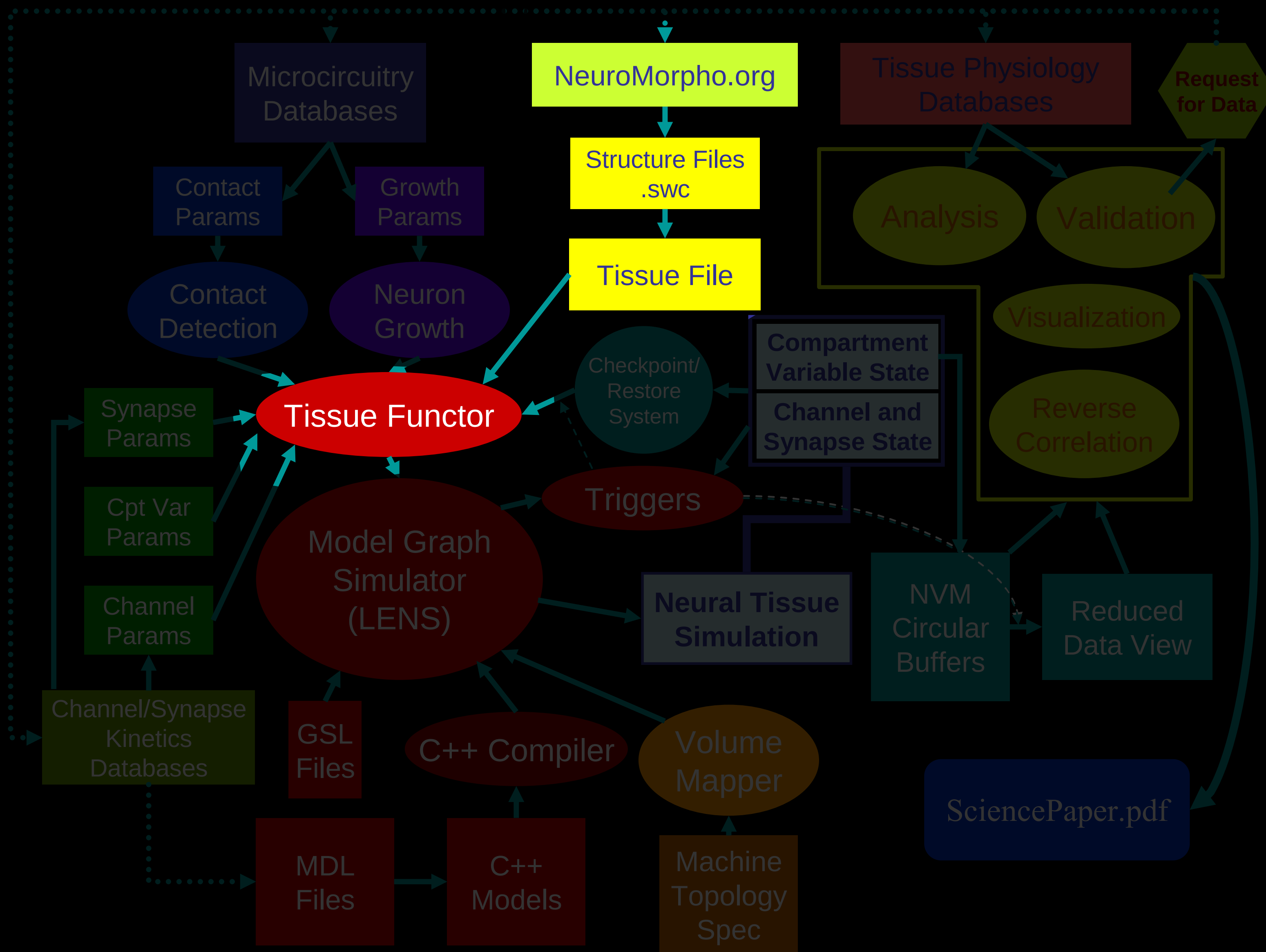


Graph Specification: Synapses

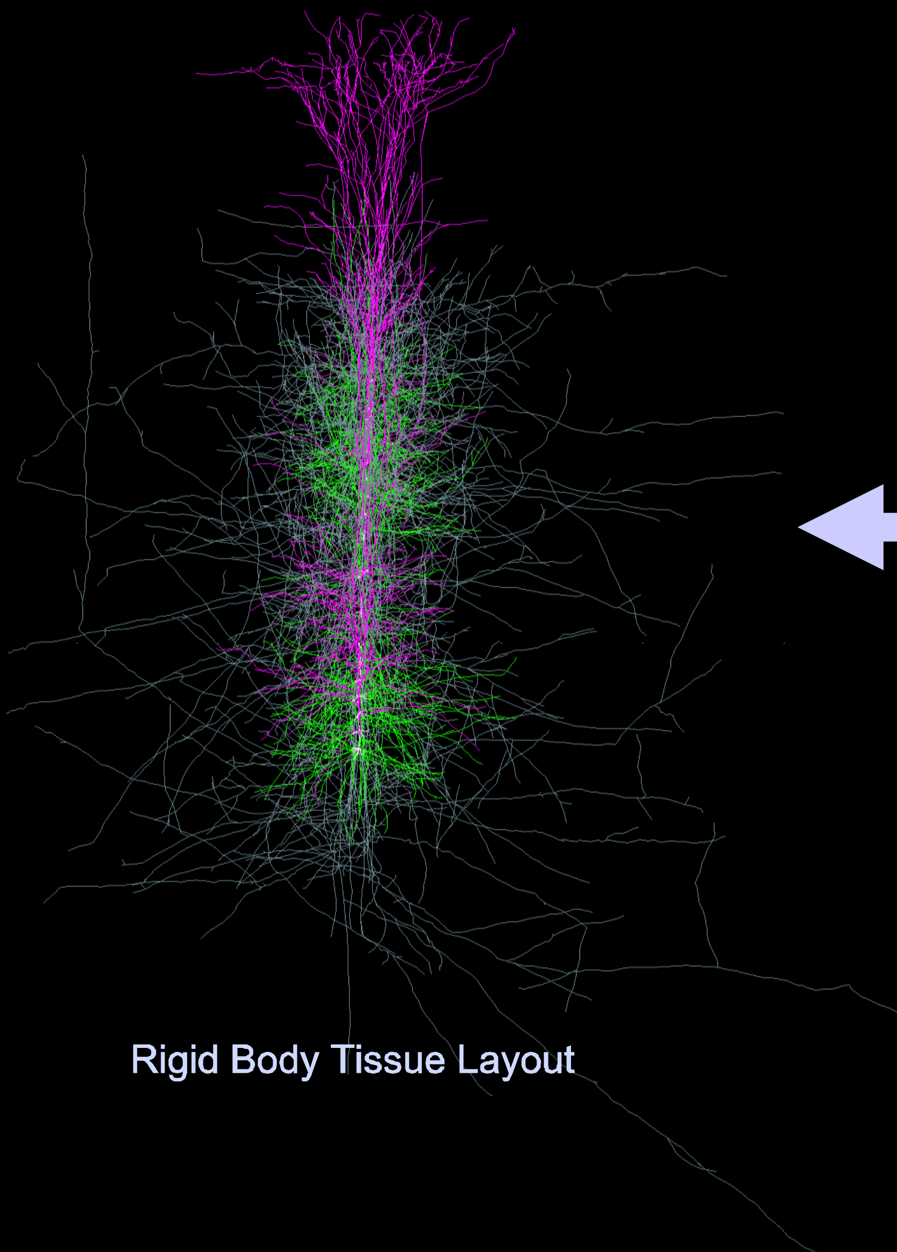
```
ELECTRICAL_SYNAPSE_TARGETS 2
BRANCHTYPE ETYPE
BRANCHTYPE ETYPE
1 0 1 0 AxAxGap [Voltage] 0.001
2 1 2 1 DenDenGap [Voltage] 0.001

ELECTRICAL_SYNAPSE_COSTS 2
AxAxGap 0.005309
DenDenGap 0.005309

CHEMICAL_SYNAPSE_TARGETS 6
BRANCHTYPE ETYPE
BRANCHTYPE ETYPE
1 1 2 0 GABAA [Voltage] [Voltage] 0.1667
1 1 2 1 GABAA [Voltage] [Voltage] 0.1667
1 1 3 0 GABAA [Voltage] [Voltage] 0.1667
1 0 2 0 AMPA [Voltage] [Voltage] 1.0
1 0 2 1 AMPA [Voltage] [Voltage] 1.0
1 0 3 0 AMPA [Voltage] [Voltage] 1.0 NMDA [Voltage] [Voltage, Calcium] 1.0

CHEMICAL_SYNAPSE_COSTS 2
AMPA 0.296407
GABAA 0.149978
```


Voltage
NMDA Receptor
IC Ca Conc
Ca Conduct
Ca Rev Pots
[Ca2+] Array
Ca2+ Concentration
[Ca2+] Array
[Ca2+]
Ca Flux
Connexon
Connexon
Ca2+ Concentration
Conduct Array
Rev Pot Array
KCa Channel
Voltage
AMPA Receptor
IC Na Conc
Conductance
Rev Potential
Voltage Array
IC Na Conc
HH Voltage
Voltage Array
Voltage
Current
Connexon
Connexon
HH Voltage
Conduct Array
Rev Pot Array
Na Channel

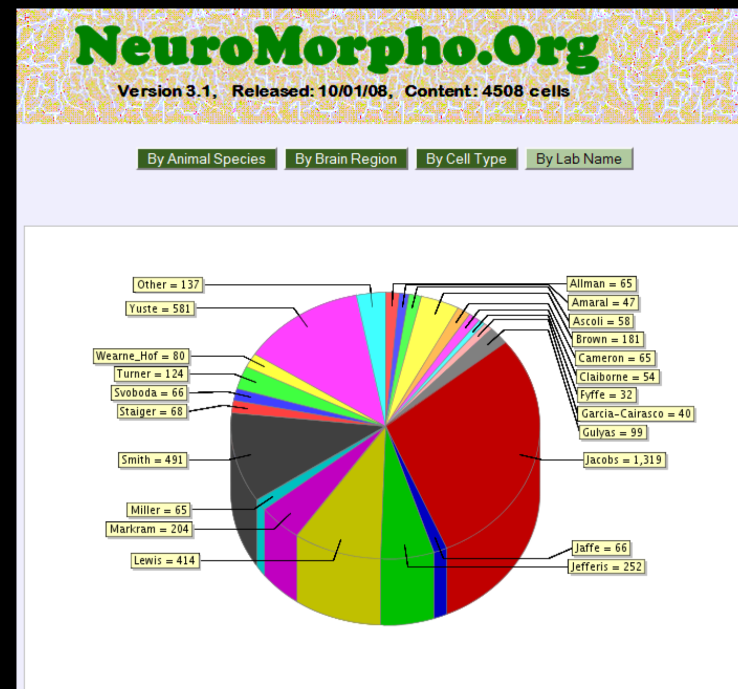
Neural Tissue Simulation Workflow



SIMULATED "MINICOLUMN"



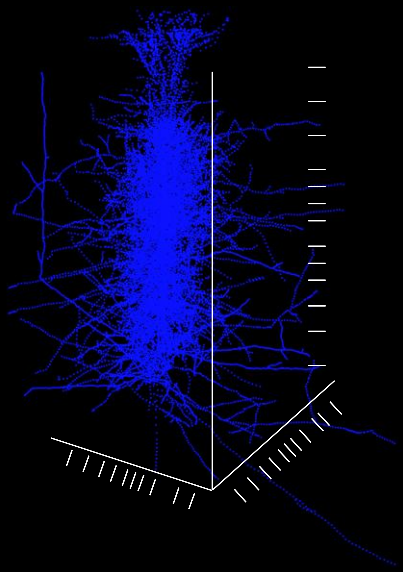
Rigid Body Tissue Layout



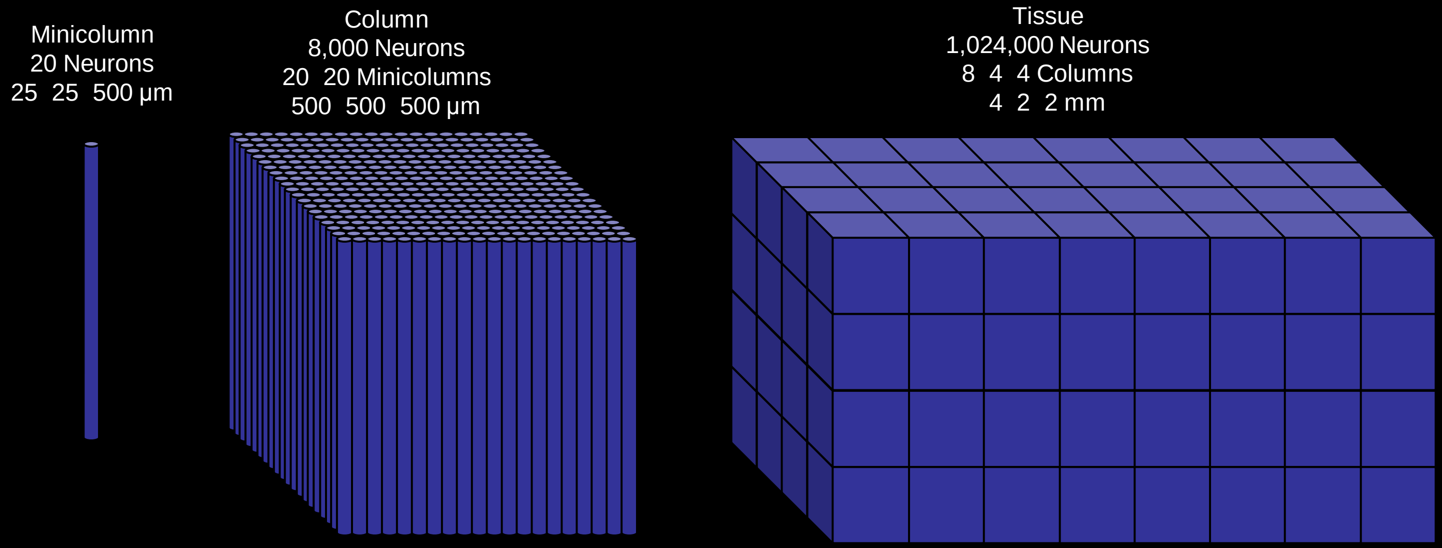
NEURAL TISSUE SIMULATION ON BLUE GENE

SIMULATION APPROACH:

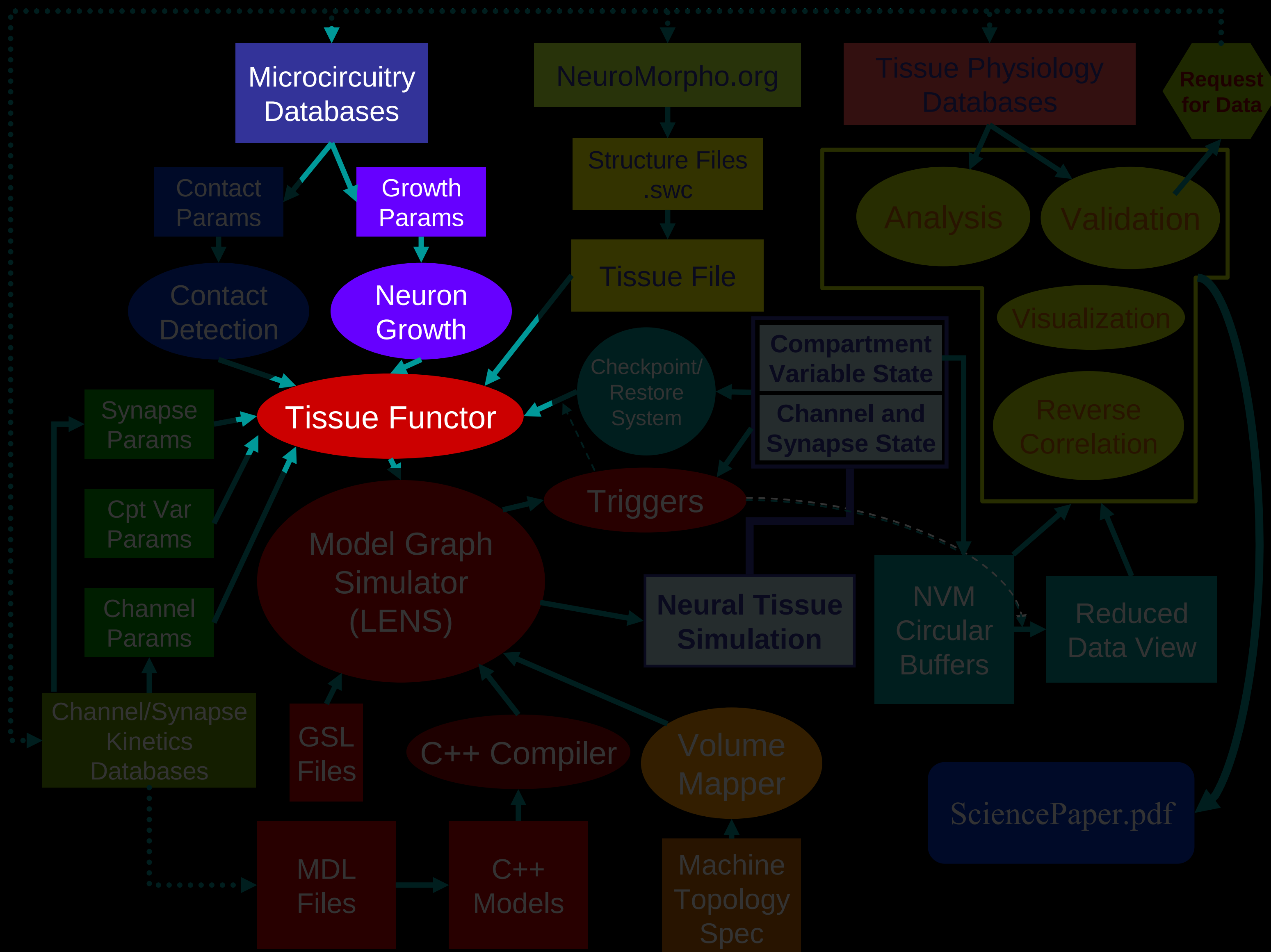
- Distribute tissue points weighted by computational complexity
- Scale out tissue simulation across all three dimensions
- Maintain realistic neuron and synapse densities at each scale



Simulation Element	Number	Processor Balance
Neurons	1,024,000,	N/A
Branches	344,474,059	84,100 7,406
Junctions	208,947,659	51,012 4,026
Compartments	1,083,289,600	264,475 7,582
Na Channels	330,613,914	80,716 7,440
KDR Channels	330,613,914	80,716 7,440
AMPA Synapses	8,186,972,360	1,998,772 720,155
GABAA Synapses	2,255,068,948	550,553 169,064
Connexons	7,626,124	1,861 820



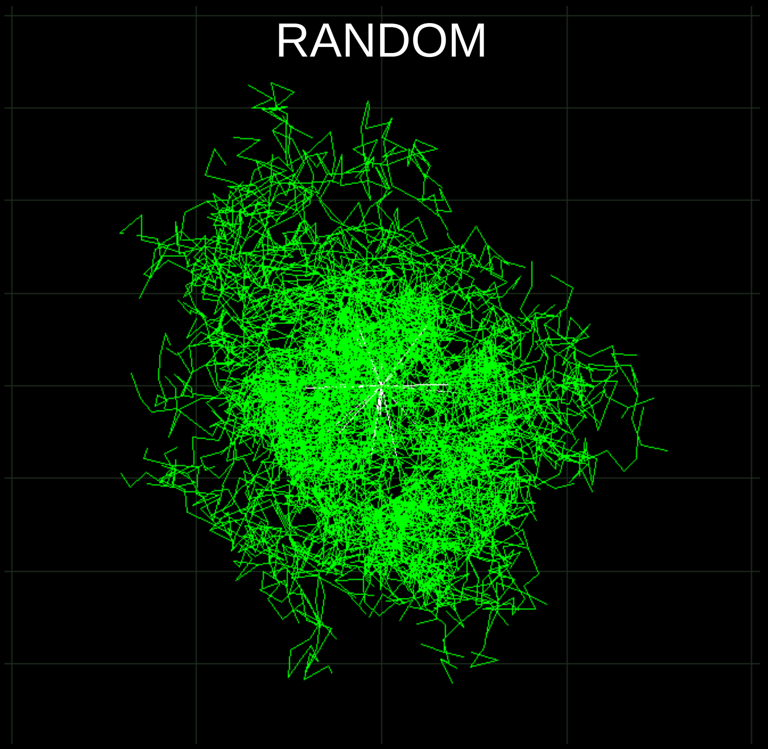
Neural Tissue Simulation Workflow

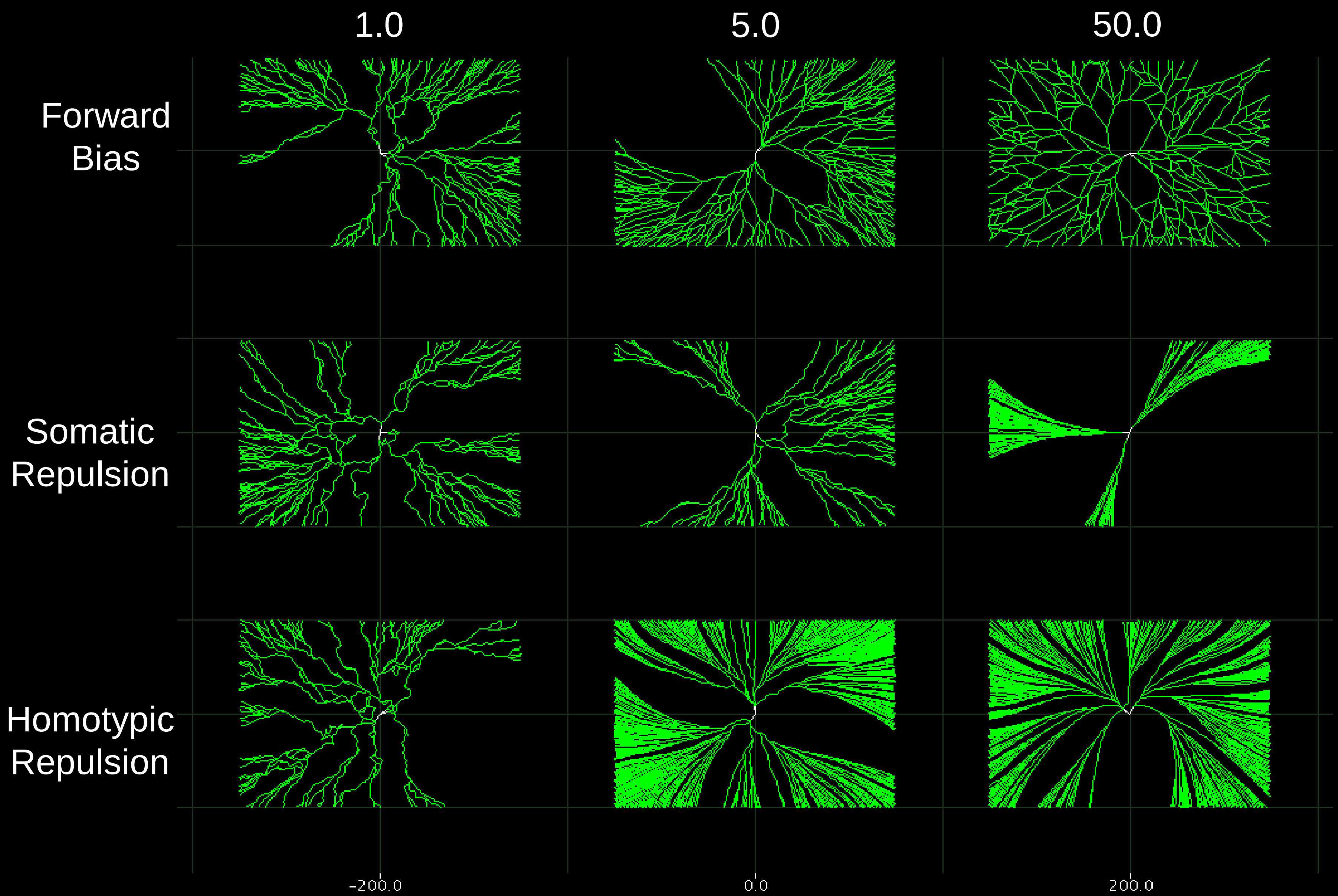


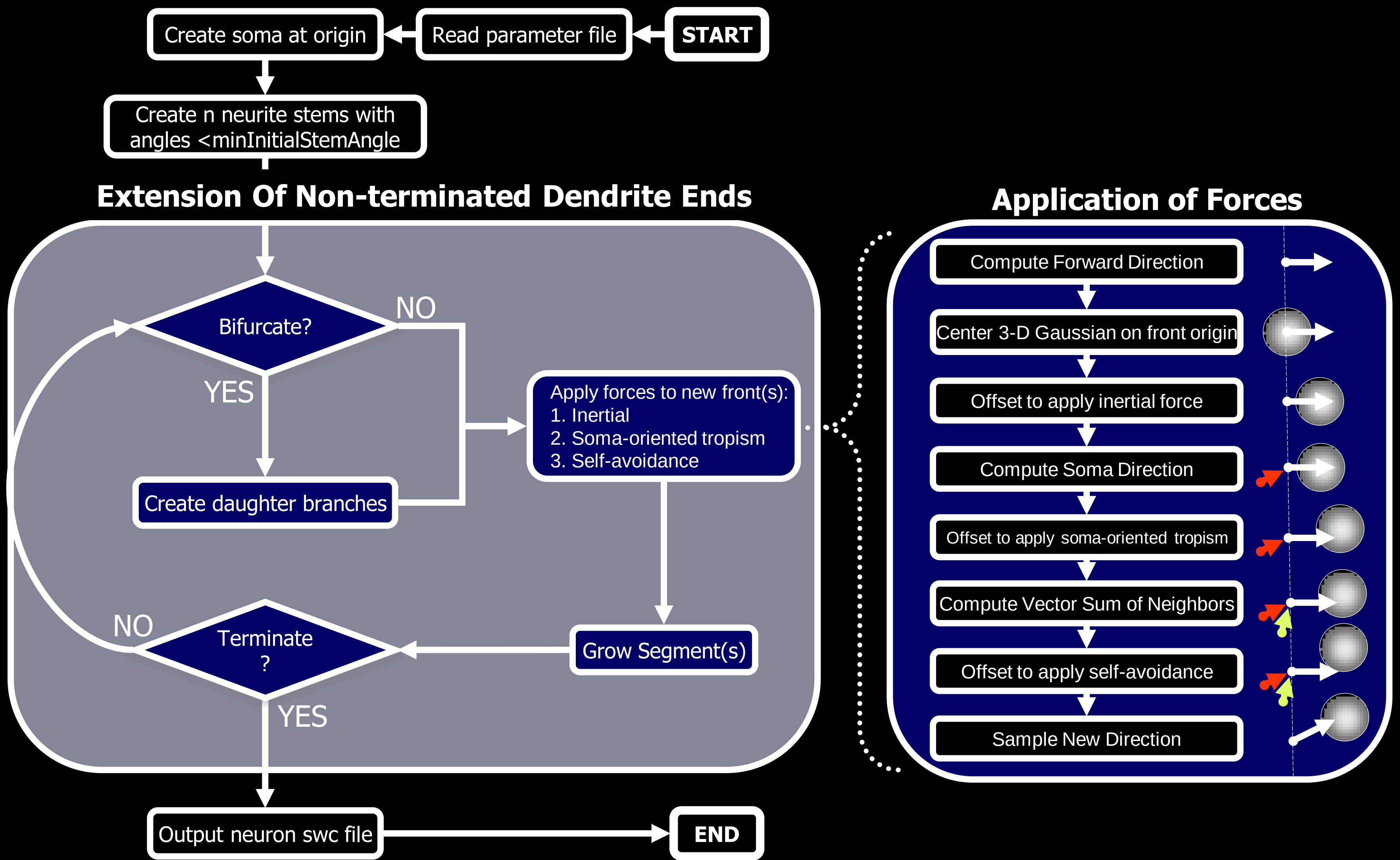
Modified Diffusion Model

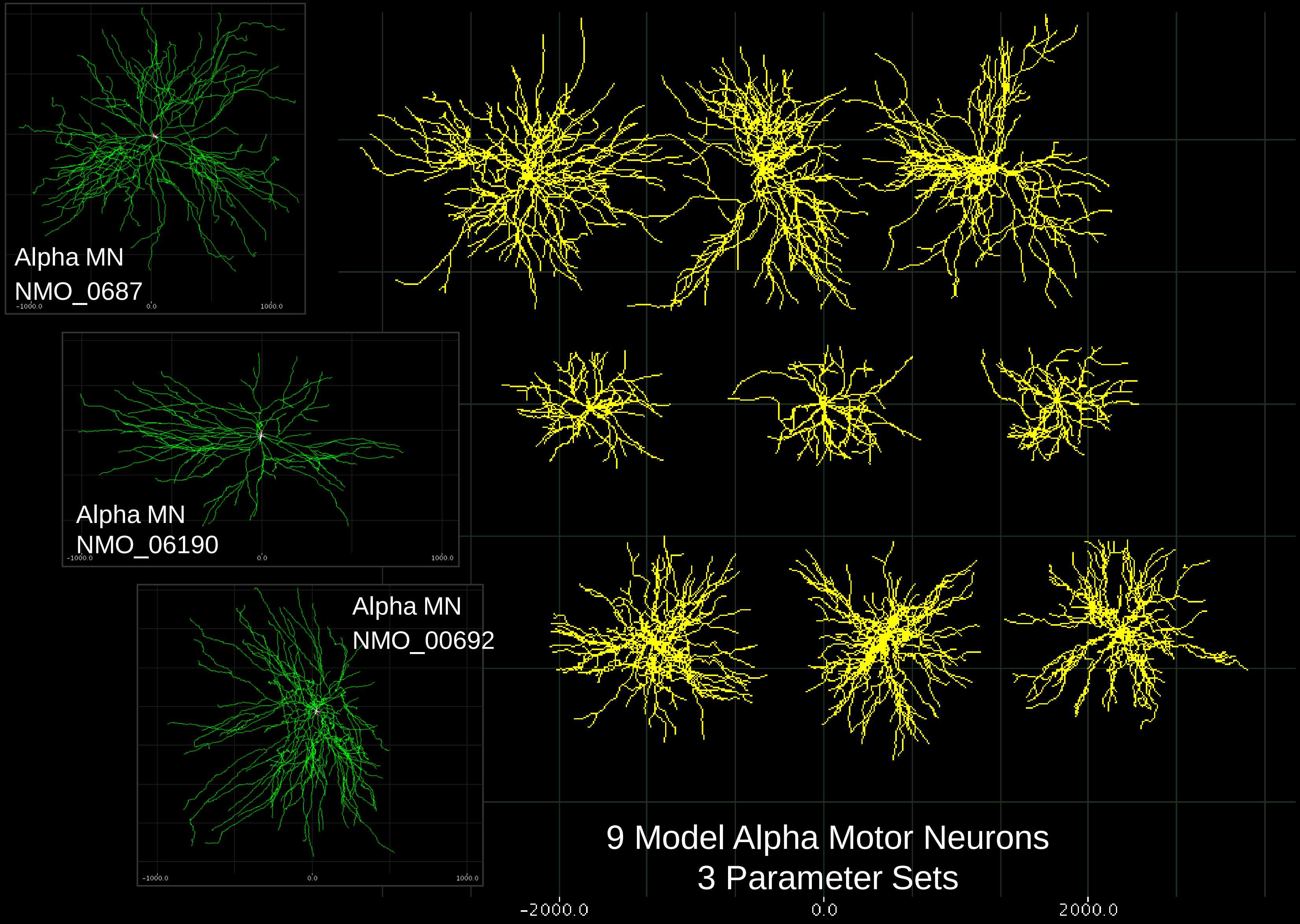
- Points sampled from 3-D Gaussian distribution centered on previous point
- Branching is represented as particle division
- Termination is represented by collision between particles and past trajectories as annihilation

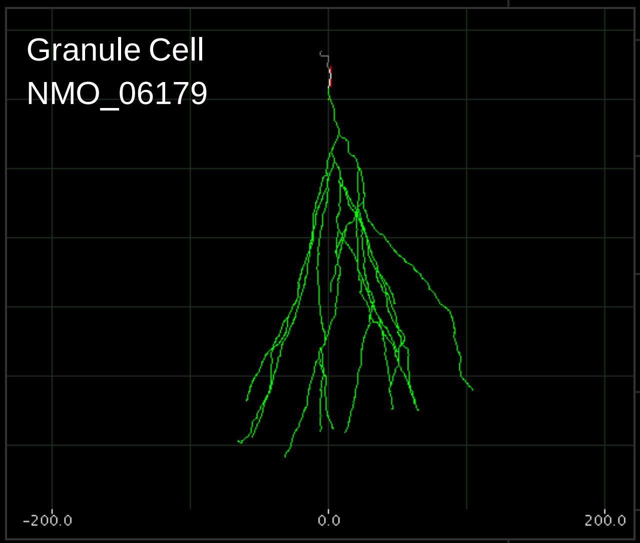
COMPLETELY RANDOM

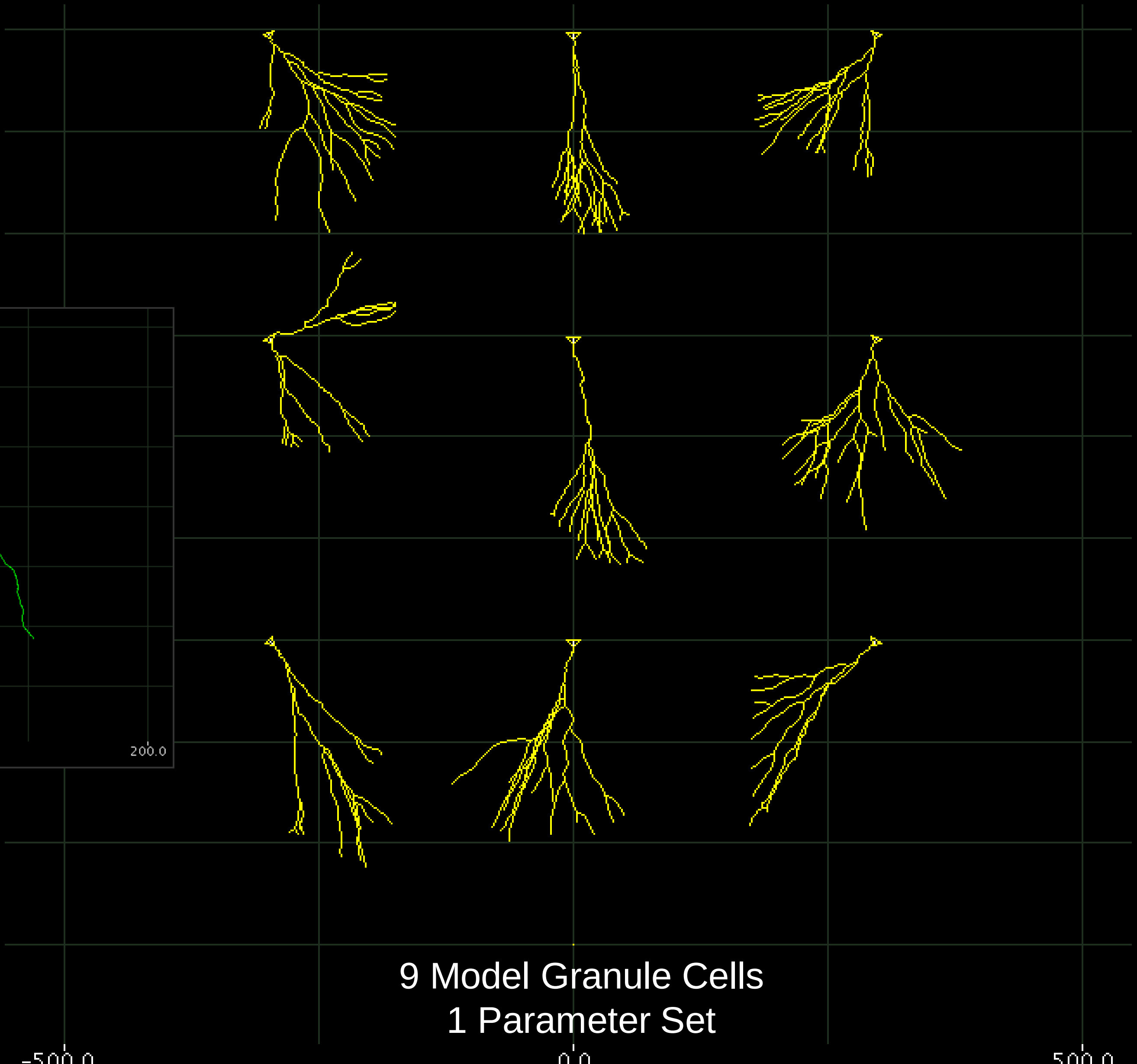



1.0
5.0
50.0
Forward Bias
Somatic Repulsion
Homotypic Repulsion
-200.0
0.0
200.0


Create soma at origin
Read parameter file
START
Create n neurite stems with angles <minInitialStemAngle
Extension Of Non-terminated Dendrite Ends
Bifurcate?
NO
YES
Create daughter branches
Apply forces to new front(s):
1. Inertial
2. Soma-oriented tropism
3. Self-avoidance
Grow Segment(s)
Terminate ?
NO
YES
Output neuron swc file
END
Application of Forces
Compute Forward Direction
Center 3-D Gaussian on front origin
Offset to apply inertial force
Compute Soma Direction
Offset to apply soma-oriented tropism
Compute Vector Sum of Neighbors
Offset to apply self-avoidance
Sample New Direction

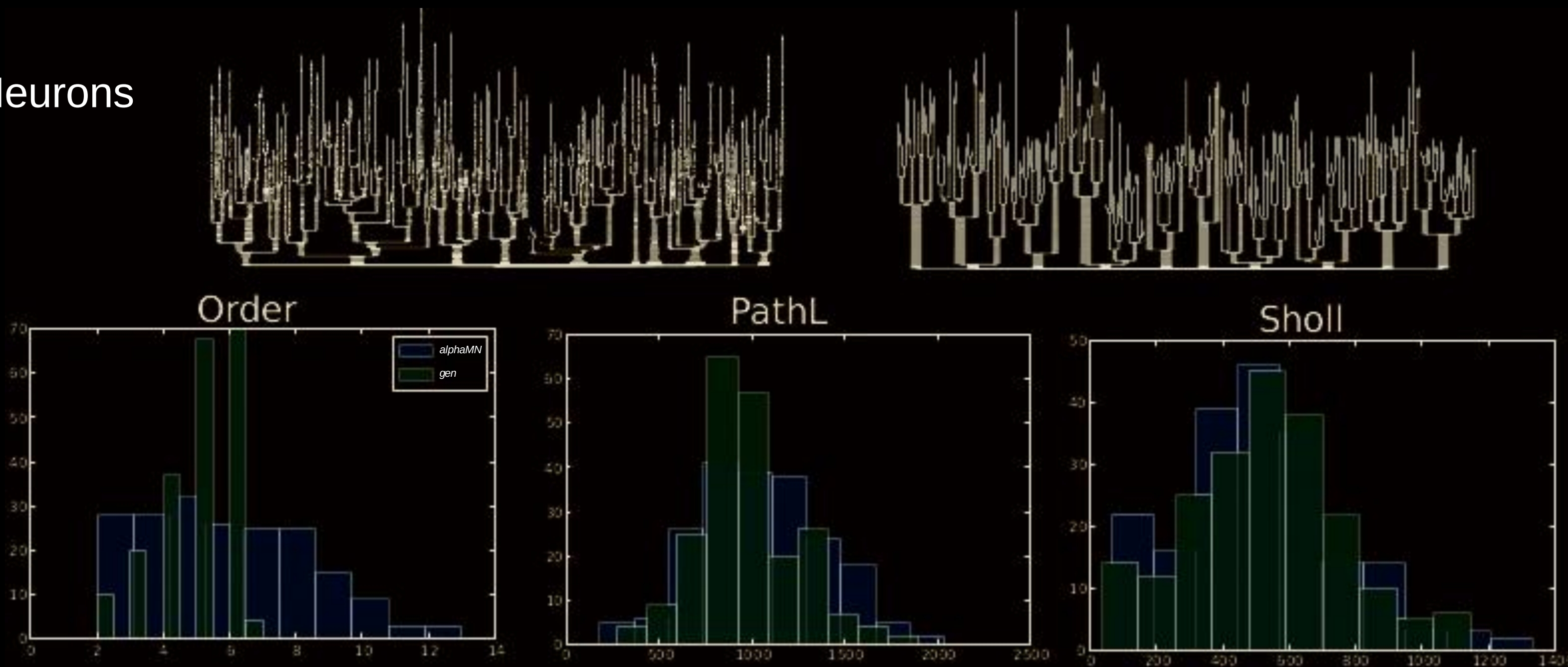

Alpha MN
NMO_0687
-1000.0
0.0
1000.0
Alpha MN
NMO_06190
-1000.0
0.0
1000.0
Alpha MN
NMO_00692
-1000.0
0.0
1000.0
9 Model Alpha Motor Neurons
3 Parameter Sets
-2000.0
0.0
2000.0



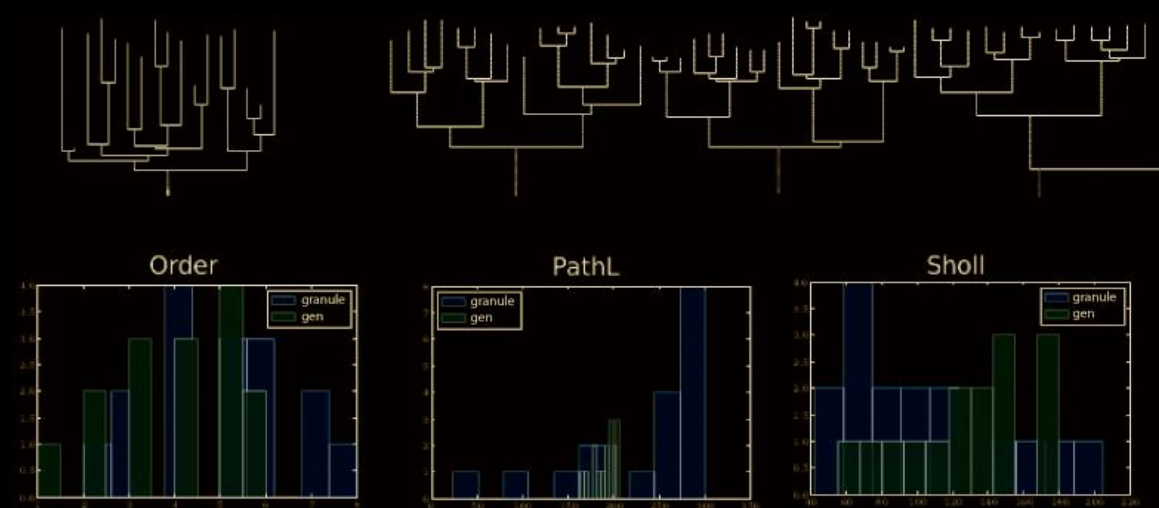


9 Model Granule Cells
1 Parameter Set

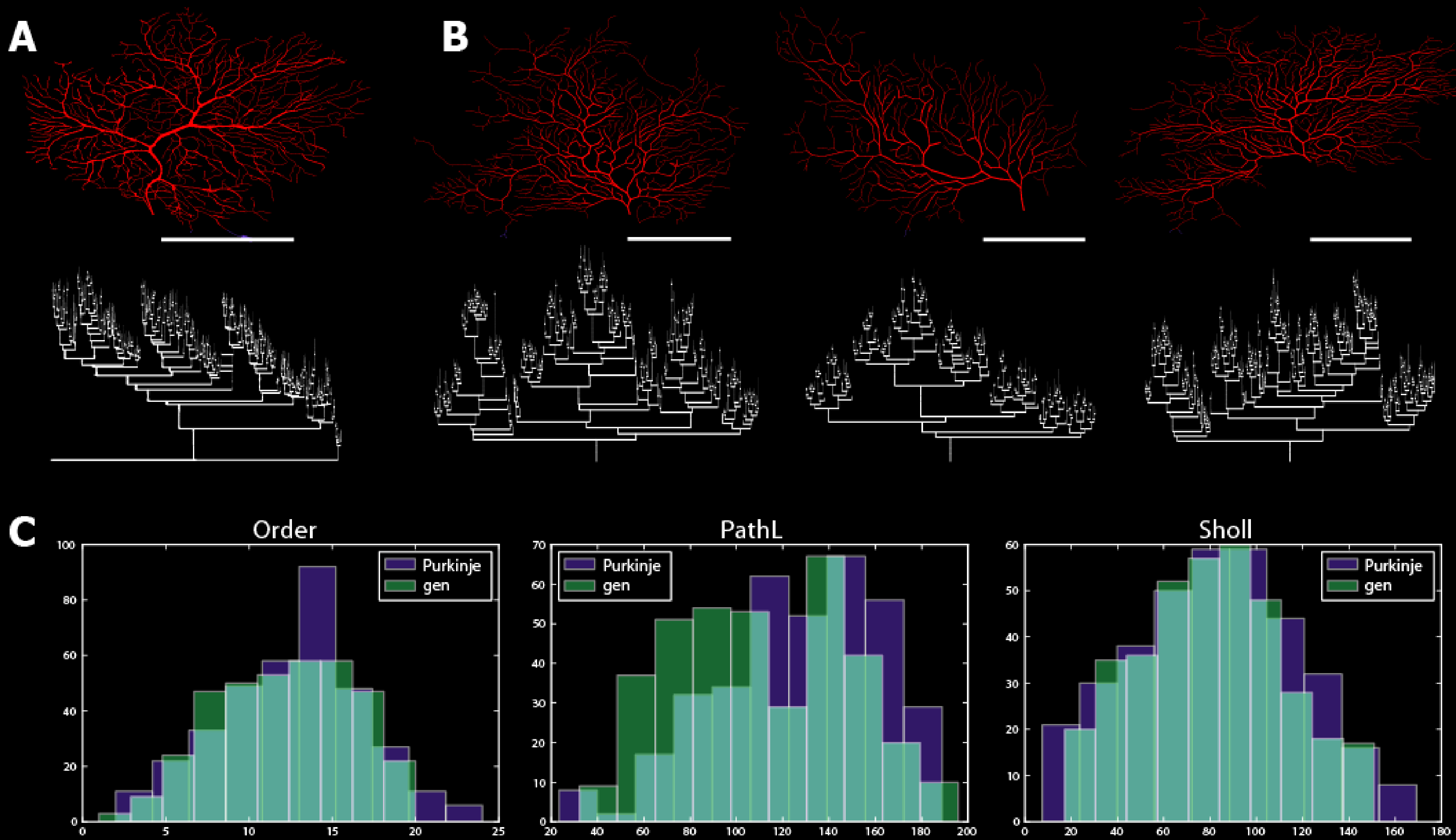
Alpha Motor Neurons



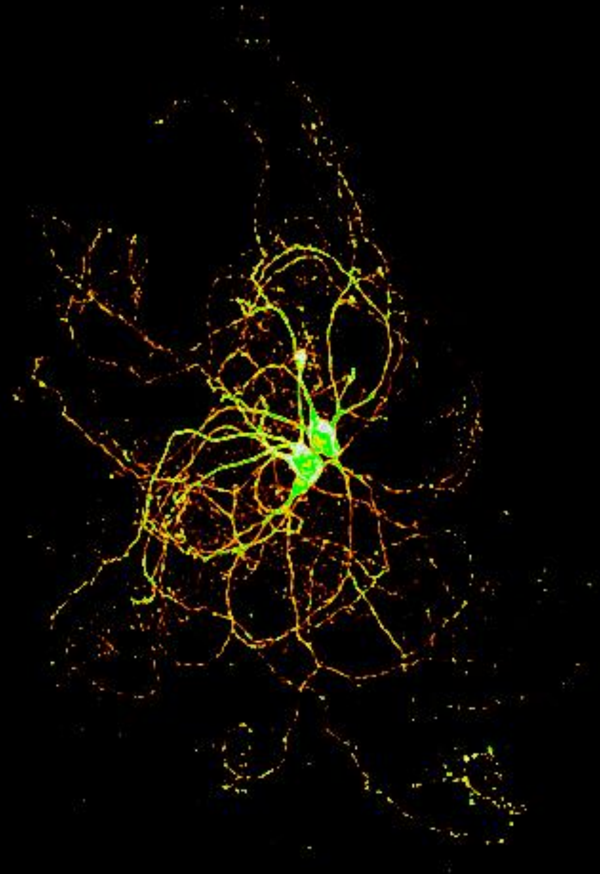
Granule Cells

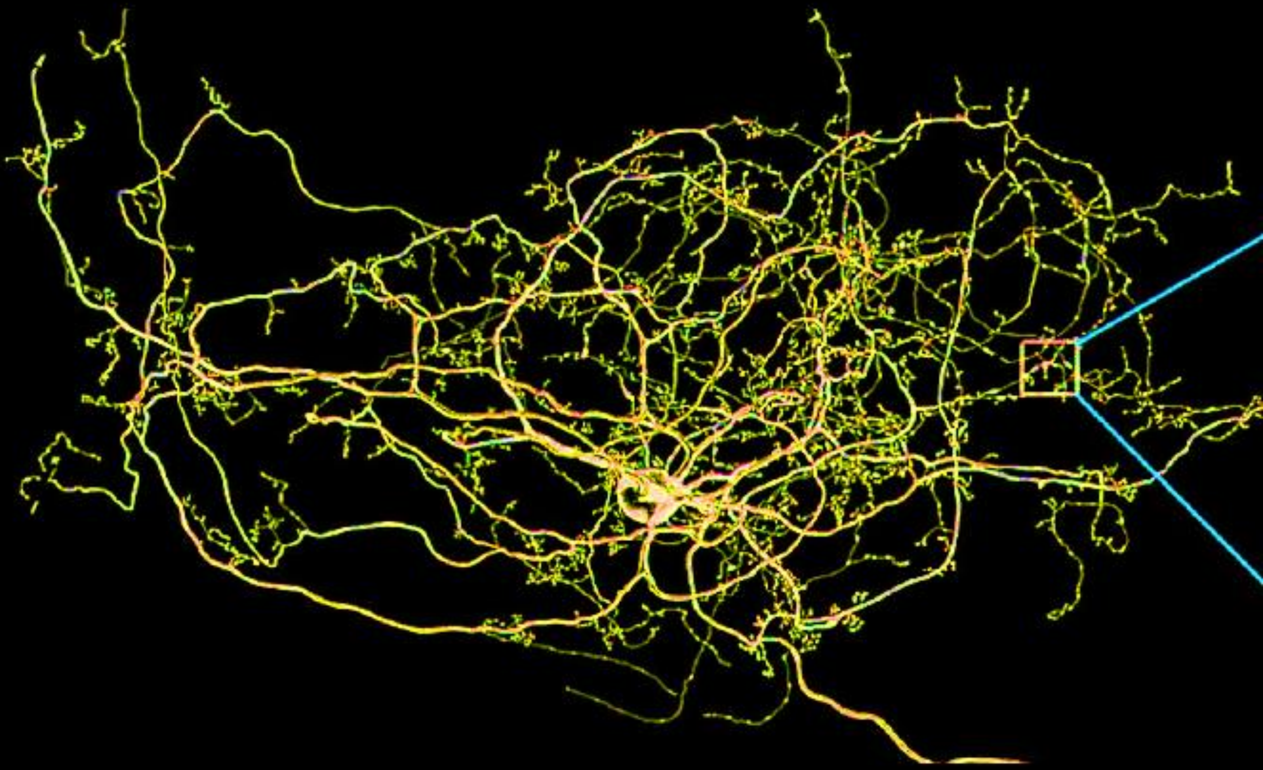


Cerebellum Modeling: Cortex

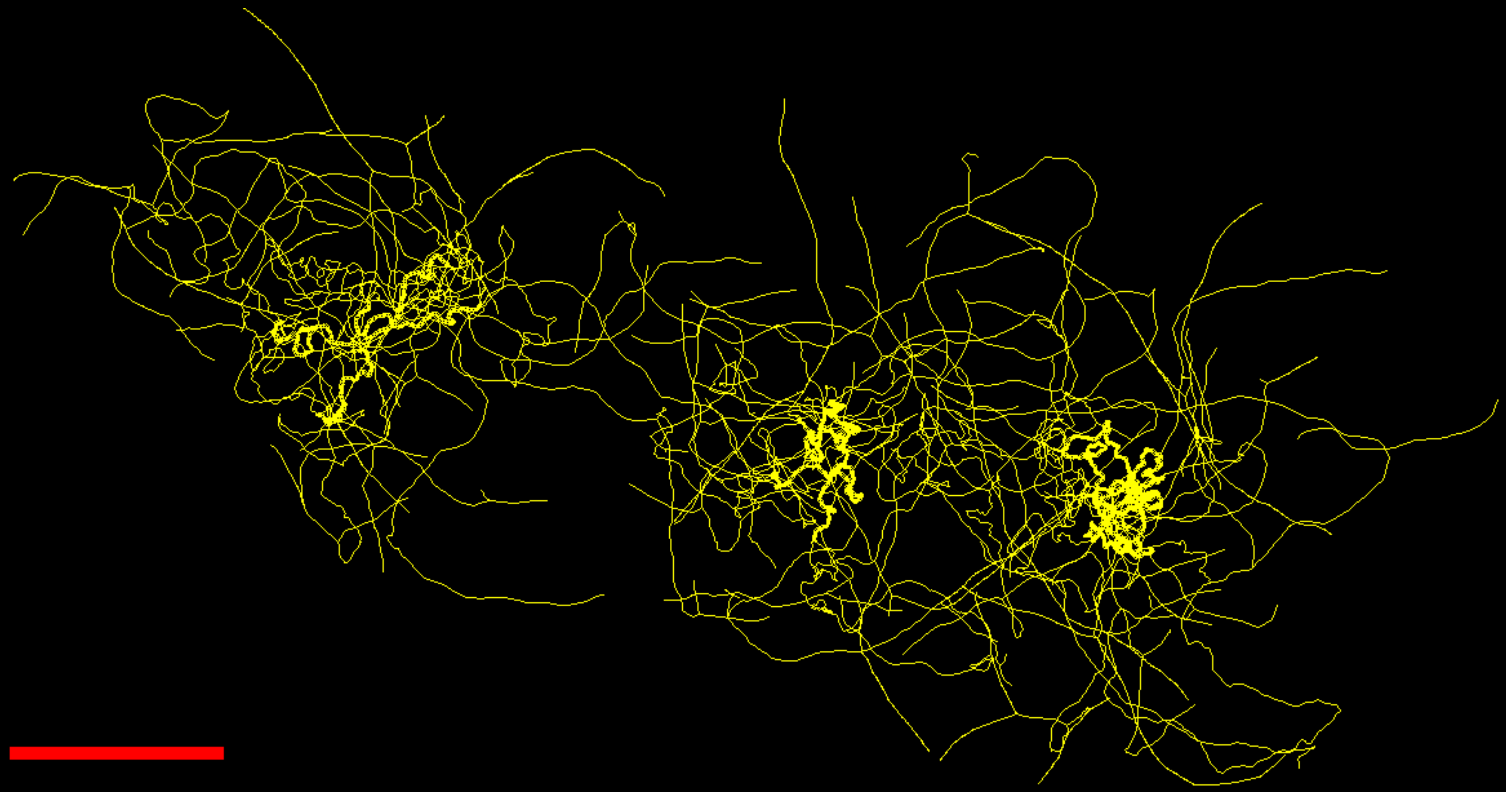


Cerebellum Modeling: Inferior Olive

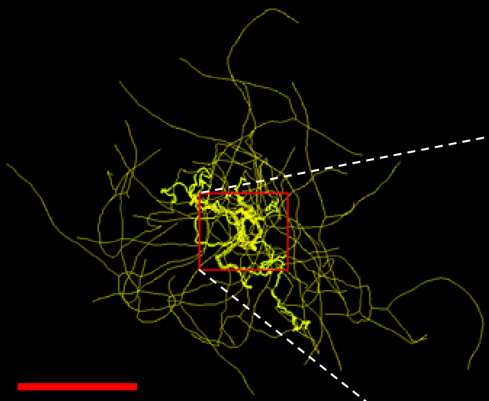


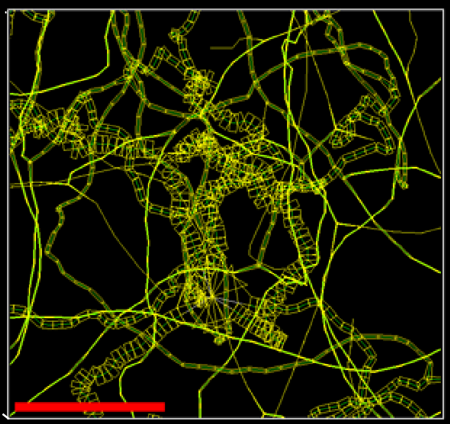


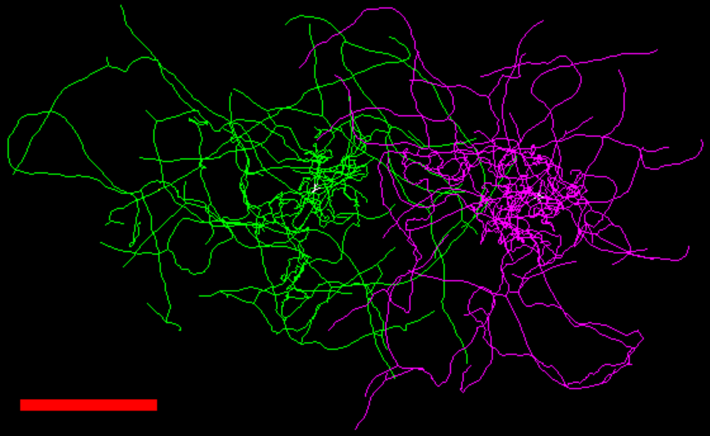
Cerebellum Modeling: Inferior Olive

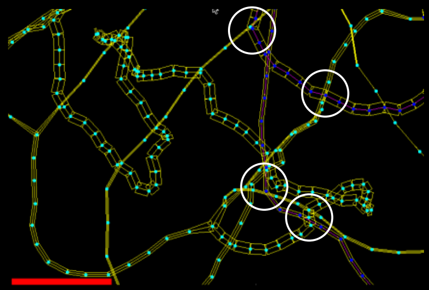


Cerebellum Modeling: Inferior Olive

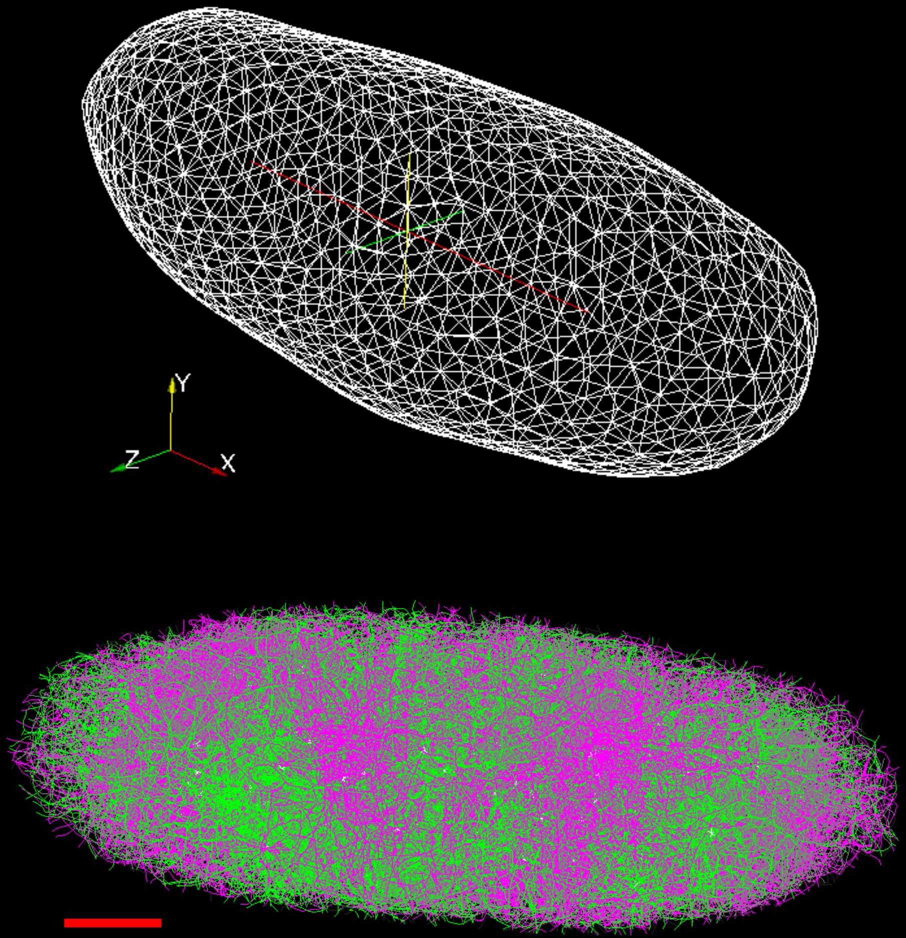


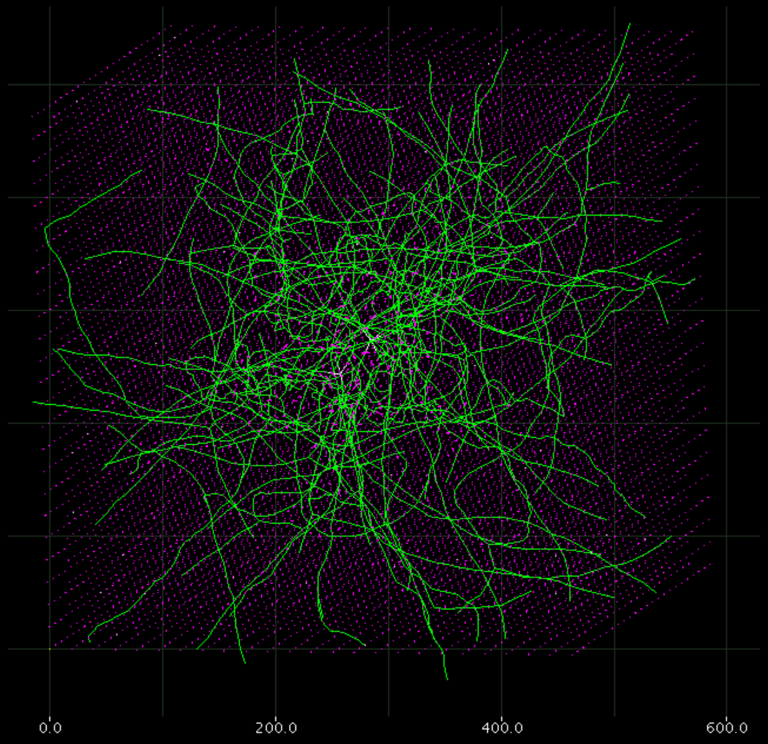






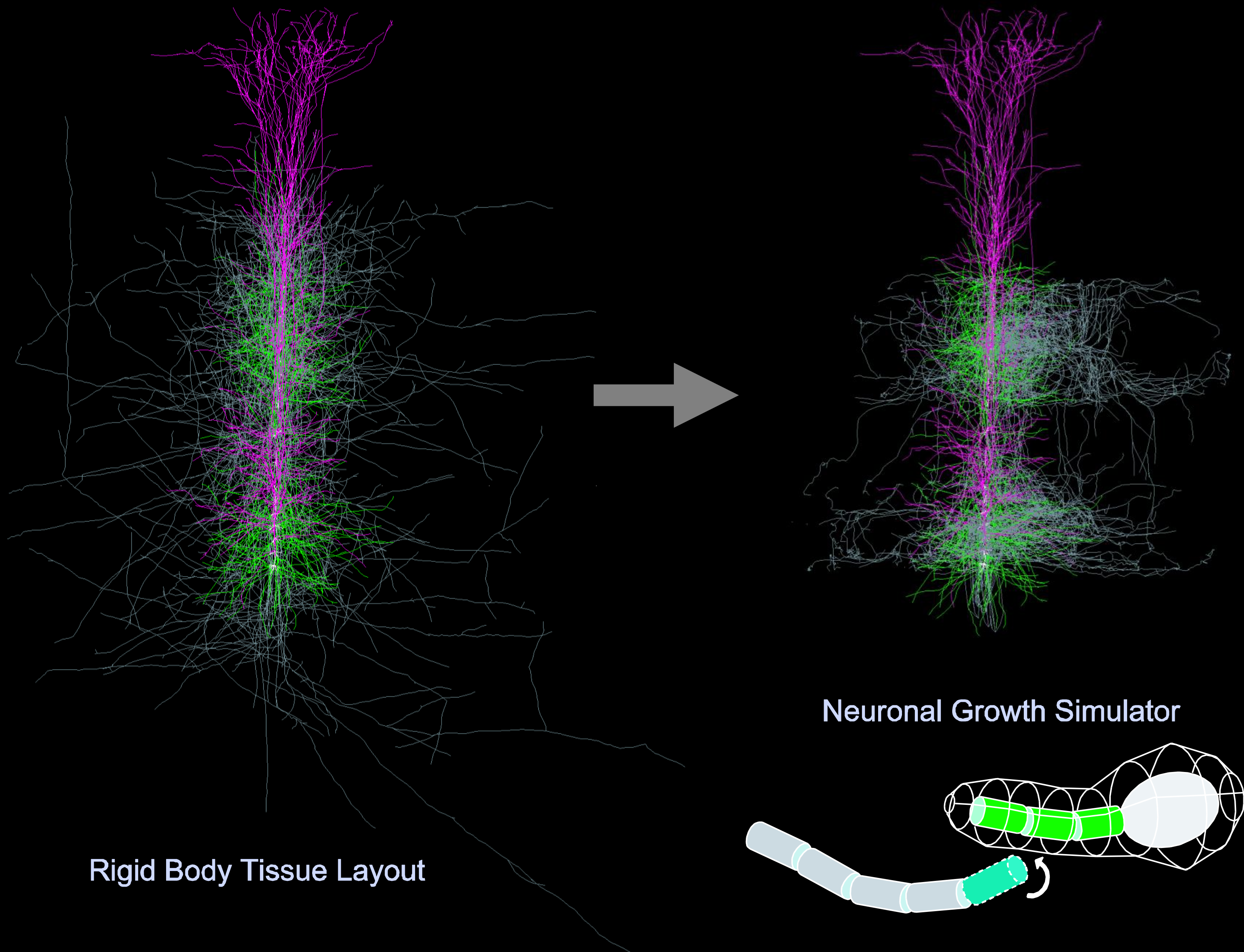
Cerebellum Modeling: Inferior Olive



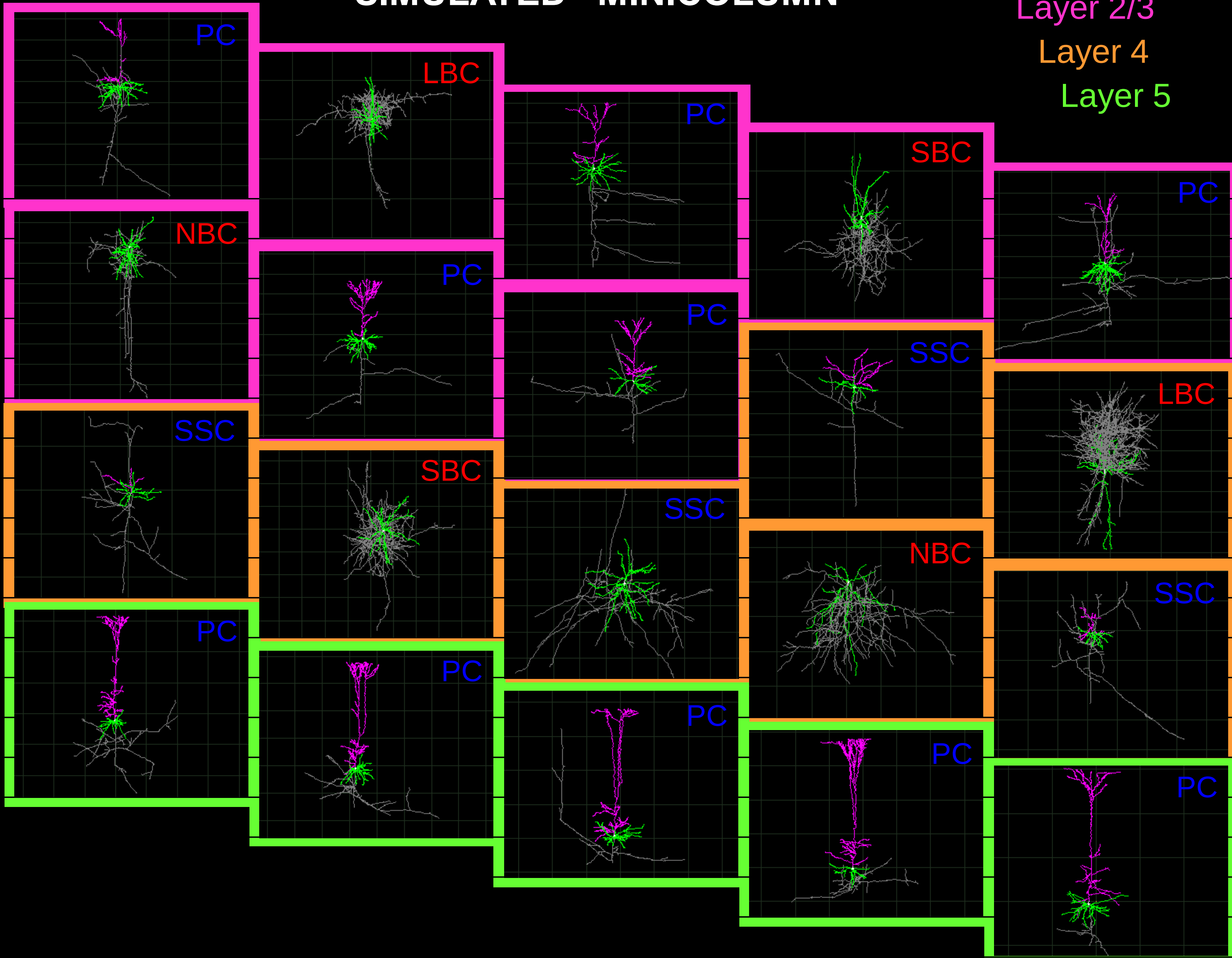


SIMULATED "MINICOLUMN" DEVELOPMENT

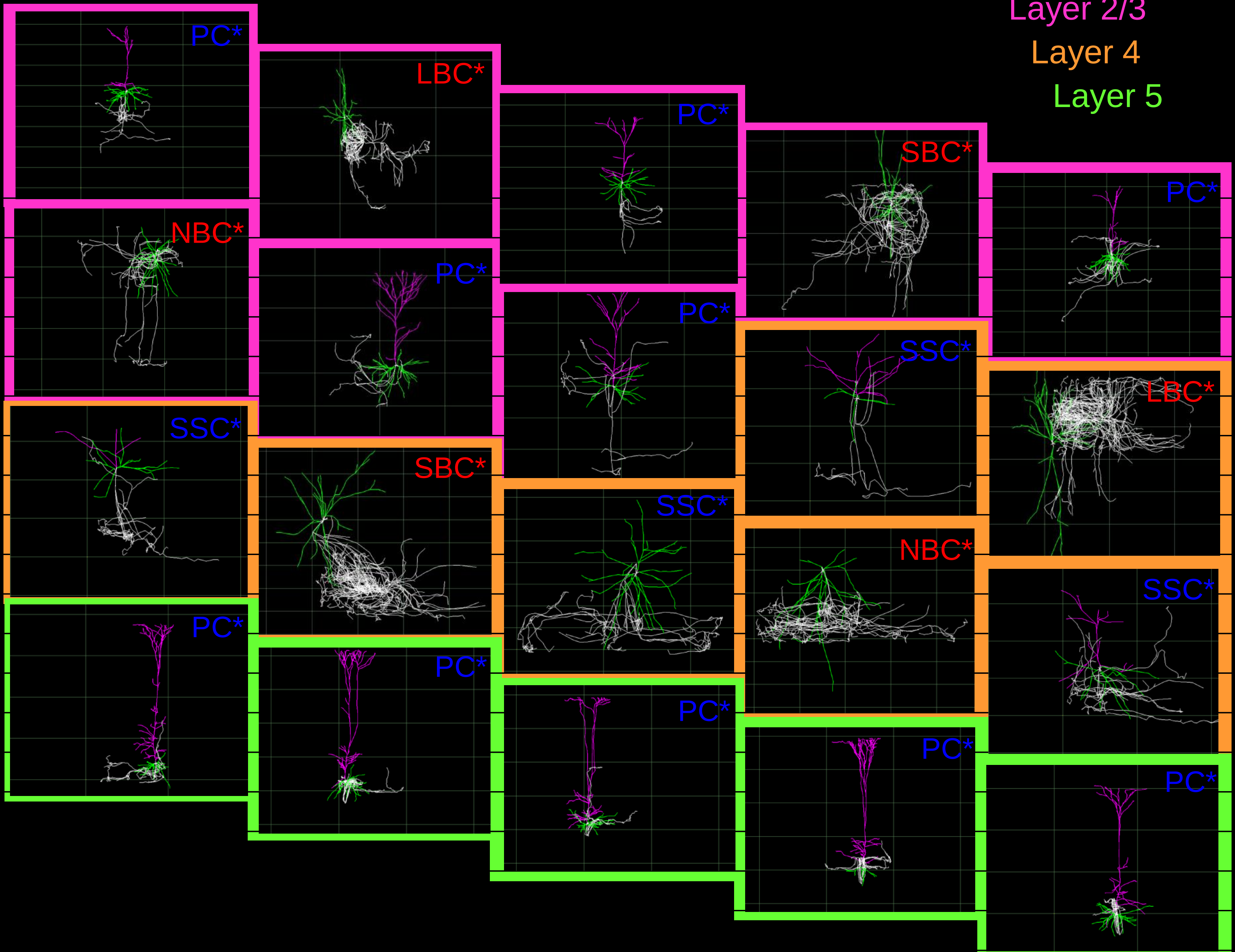
In collaboration with Mike Pitman, Protein Science & Molecular Dynamics



SIMULATED “MINICOLUMN”



SIMULATED “MINICOLUMN” DEVELOPMENT



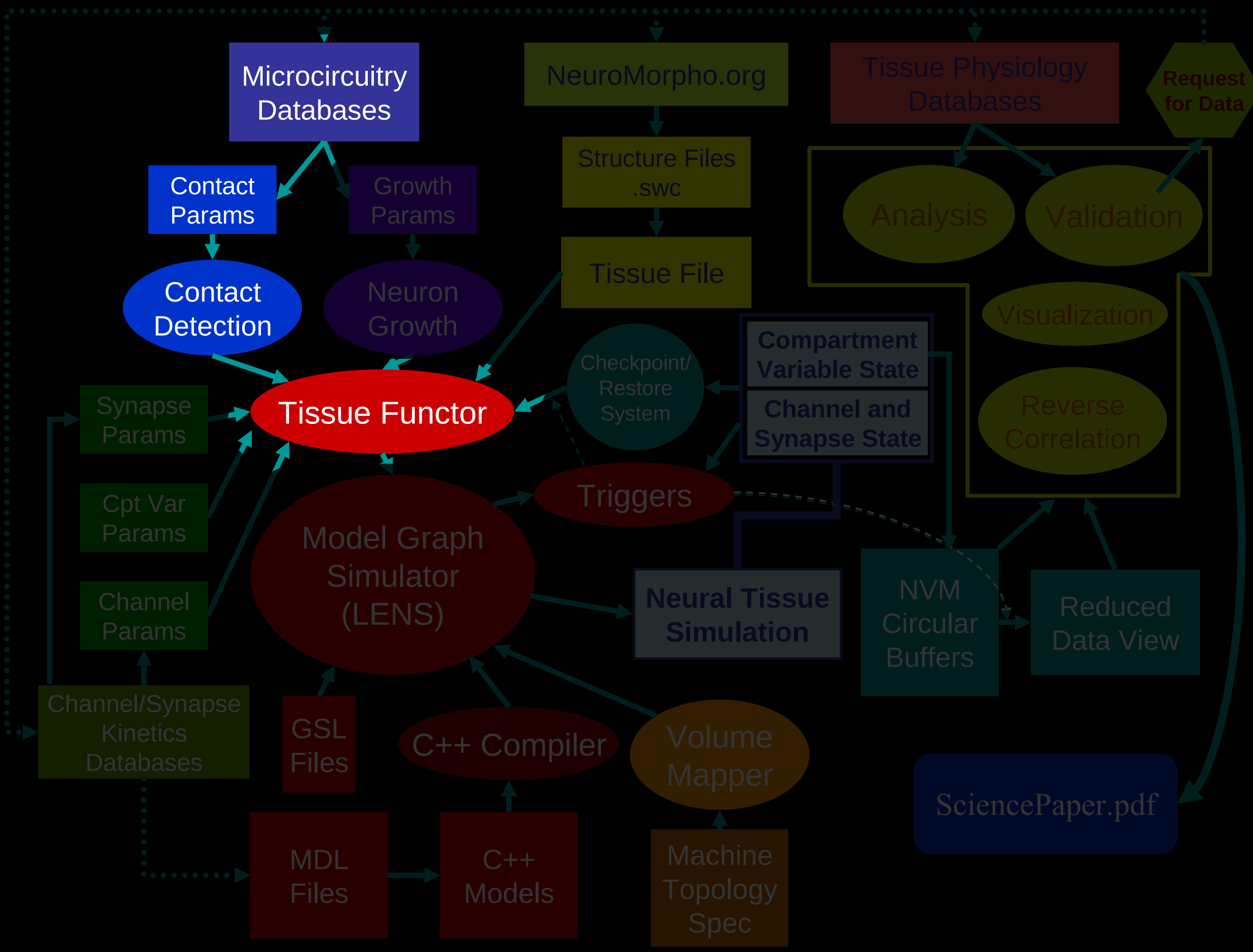
“Patient-specific models of deep brain stimulation: Influence of field model complexity on neural activation predictions

A. Chaturvedi, C. R. Butson, S. F. Lempka, S. E. Cooper, C. C. McIntyre,

Neural Tissue Simulator Approach

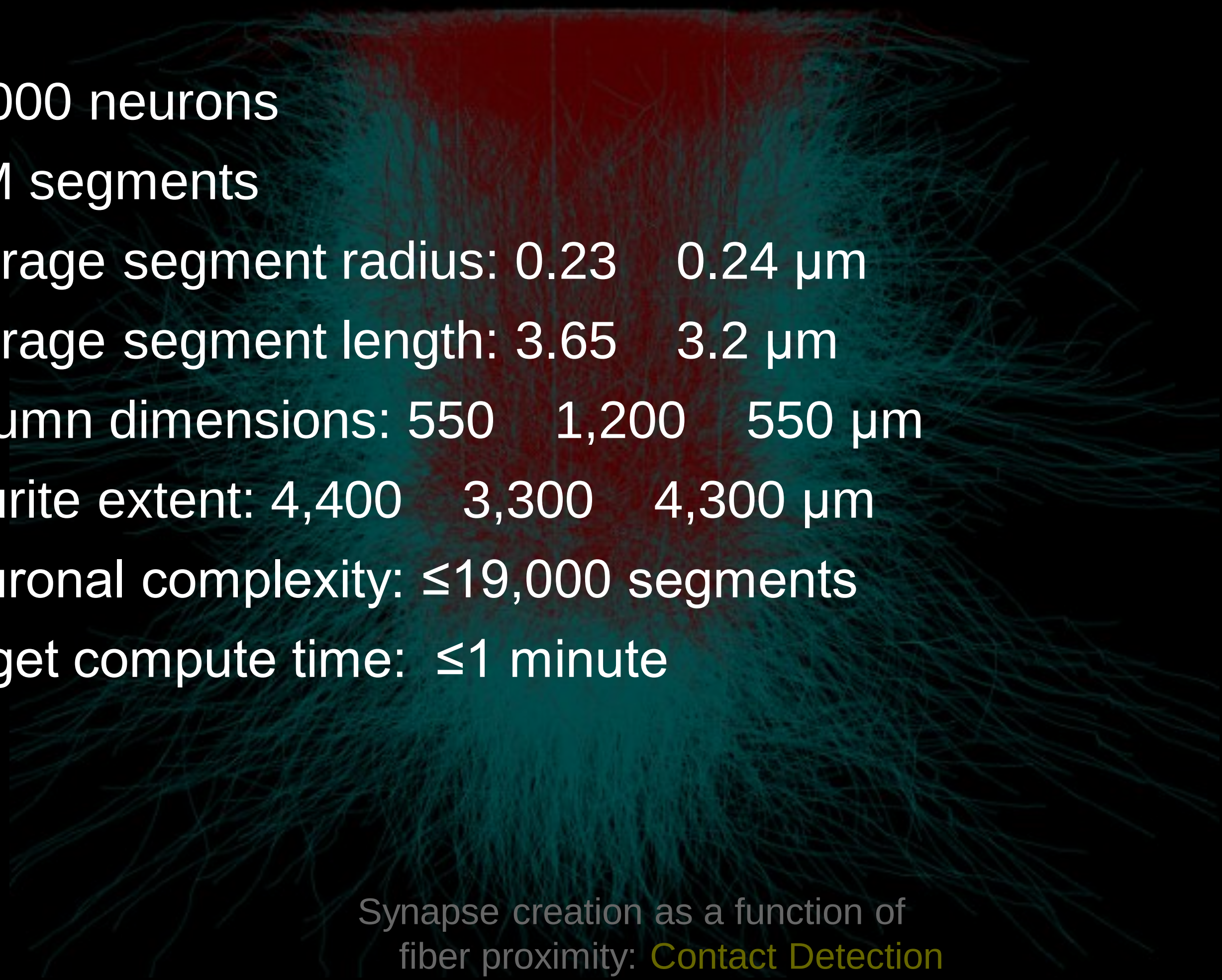
- Develop axonal tracts differentially using Neuron Growth Simulator, constrained by patient data
- Establish extracellular mesh of finite elements using new models expressed in MDL
- Connect finite elements to neural tissue elements using GSL
- Model effects of DBS on axonal tracts, generating action potentials in multi-compartment axonal models

Neural Tissue Simulation Workflow



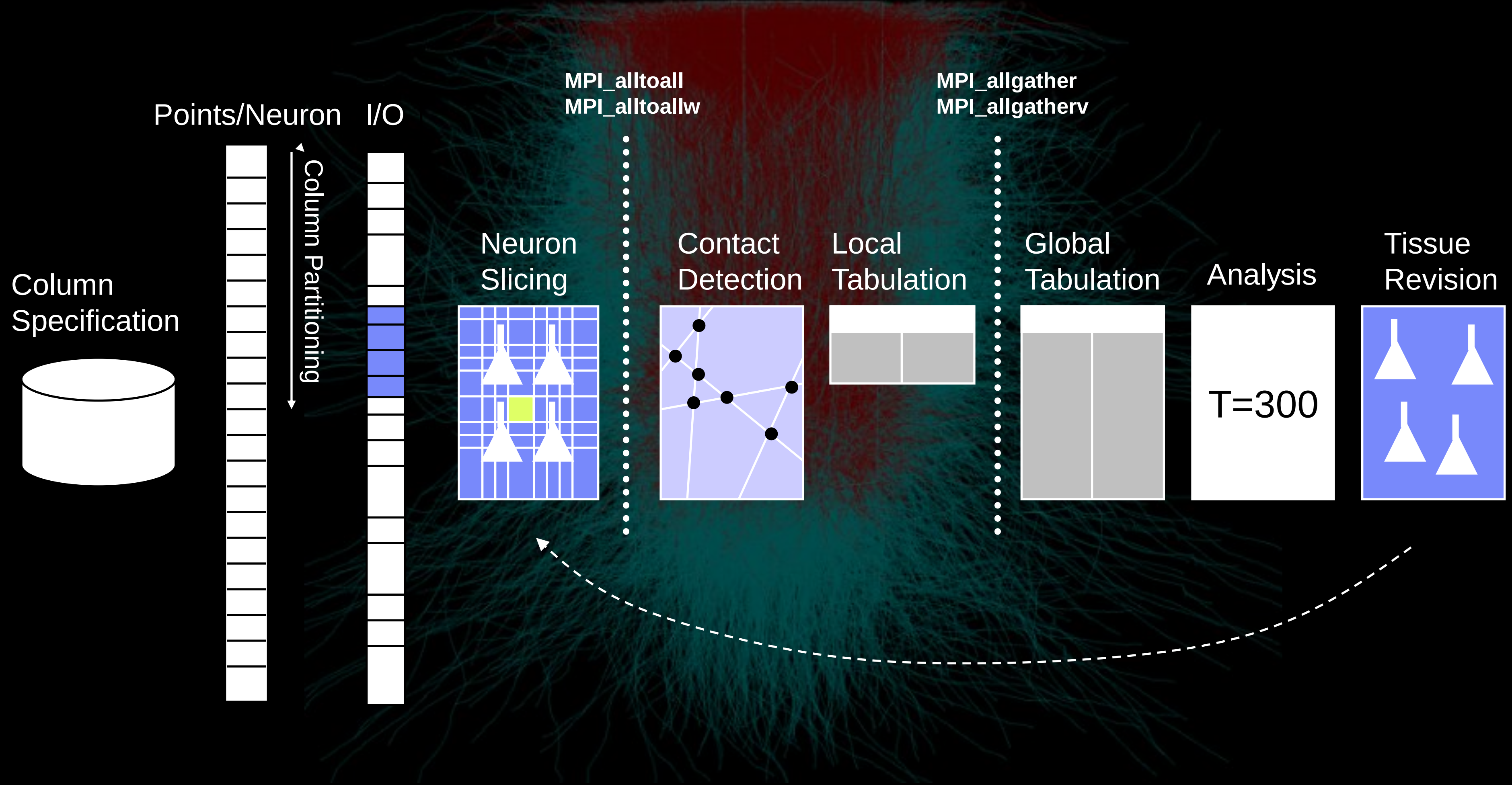
Contact Detection: The Problem

- 10,000 neurons
- 40M segments
- Average segment radius: 0.23   0.24 µm
- Average segment length: 3.65   3.2 µm
- Column dimensions: 550   1,200   550 µm
- Neurite extent: 4,400   3,300   4,300 µm
- Neuronal complexity: ≤19,000 segments
- Target compute time: ≤1 minute

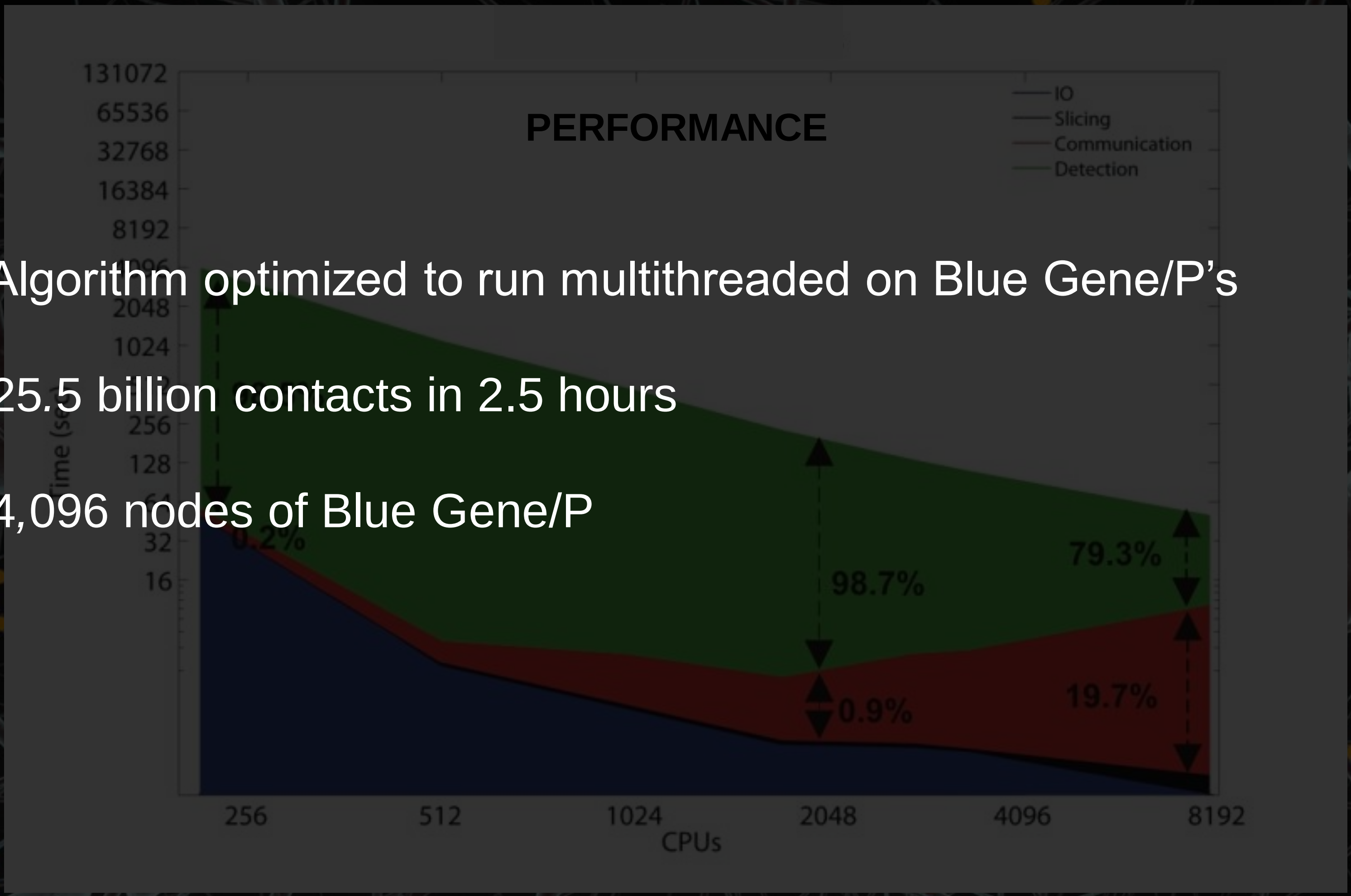


Synapse creation as a function of fiber proximity: Contact Detection

Contact Detection: Architectural Overview



Contact Detection

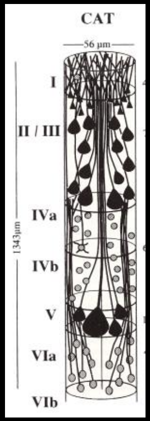


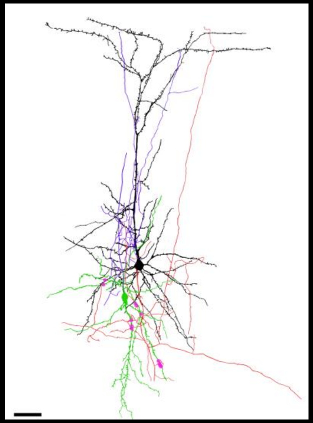
- Algorithm optimized to run multithreaded on Blue Gene/P’s
- 25.5 billion contacts in 2.5 hours
- 4,096 nodes of Blue Gene/P

Simulation Workflow In a Single Executable

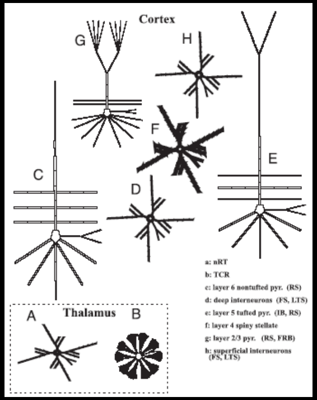
- GSL parser initializes tissue with specified grid dimensions
- Volume mapper assigns volumes to nodes of Blue Gene/P
- Tissue Functor reads tissue spec, computes consistent scheme for dividing neuron work
- On each node, Tissue Functor loads different neurons from .swc files
- Tissue Functor resamples neurons
- Communication enumerates all tissue points, constructs global point histogram, equalizes
- Neuron segments communicated to all volumes they traverse, according to histogram
- Tissue Functor executes neuron growth algorithm (optional)
- Tissue Functor executes touch detection algorithm
- Tissue Functor aggregates computational costs associated with each neuron segment
- Global cost histogram equalized in three dimensions to create a second volume slicing
- Tissue Functor communicates touches, segments to nodes responsible for models or proxies
- GSL parser creates all tissue models locally, including branches, channels, and synapses
- Probability for creating synapse models of a specific type from a set of valid touches applied
- GSL parser initializes other models such as stimulation and recording electrodes
- GSL parser interprets the specified phase structure of the simulation
- GSL parser initiates the simulation, in which each iteration comprises a sequence of phases:
  - Solve ion channel and synapse states
  - Predict branch junction states
  - Forward eliminate branches of appropriate compute orders
  - Back-substitute branches of appropriate compute orders
  - Correct branch junction states
- Between phases, state is marshalled, communicated (MPI_alltoallv), demarshalled into proxies
- Simulation terminates when the end criterion is satisfied
- All models and simulation data are destructed on all nodes of Blue Gene/P

Neural Tissue Simulation Scalability

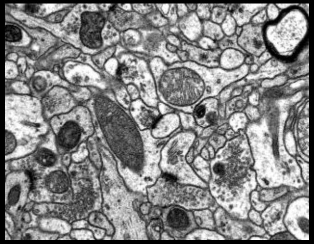




≠

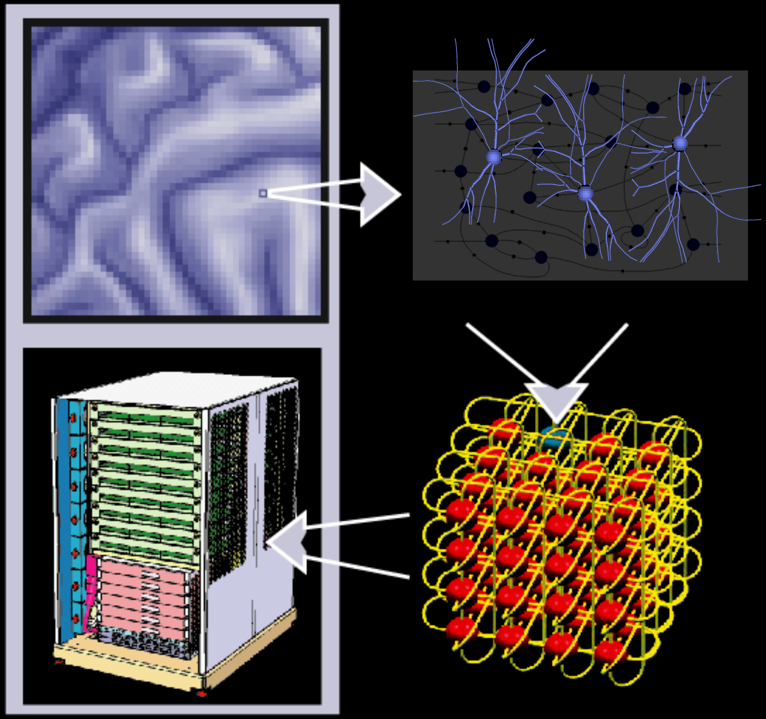


Traub et al., *J Neurophysiol* 93: 2194 2232, 2005



Neural Tissue Simulation Scalability

The question of axons…

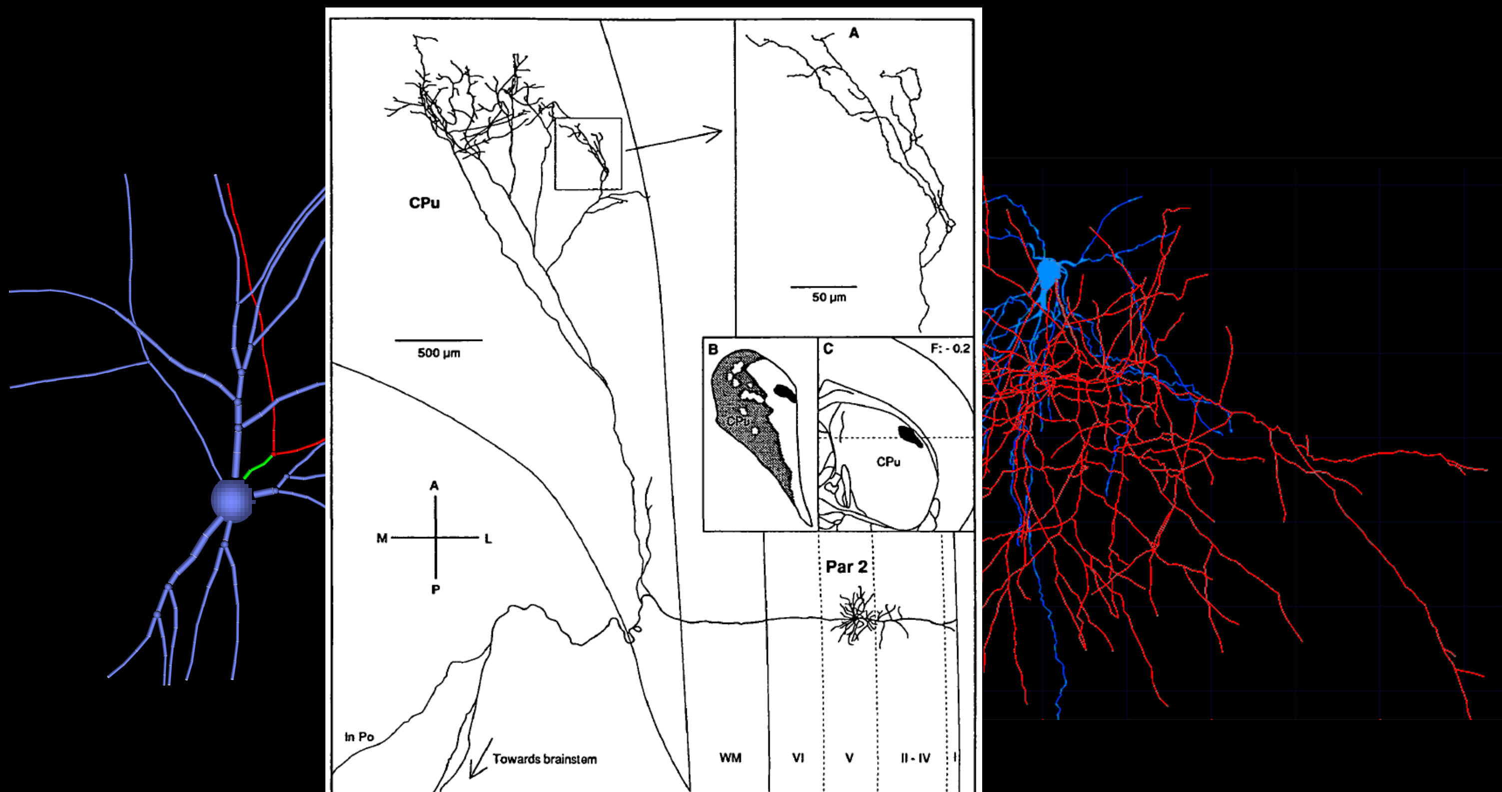


"…processors act like neurons and connections between processors act as axons…"

H. Markram. The Blue Brain Project. *Nat Rev Neurosci*, 7(2):153–160, Feb 2006.

Neural Tissue Simulation Scalability

The question of axons…



M. Lévesque, S. Gagnon, A. Parent, and M. Deschênes, "Axonal Arborizations of Corticostriatal and Corticothalamic Fibers Arising from the Second Somatosensory Area in the Rat

Neural Tissue Simulation Scalability

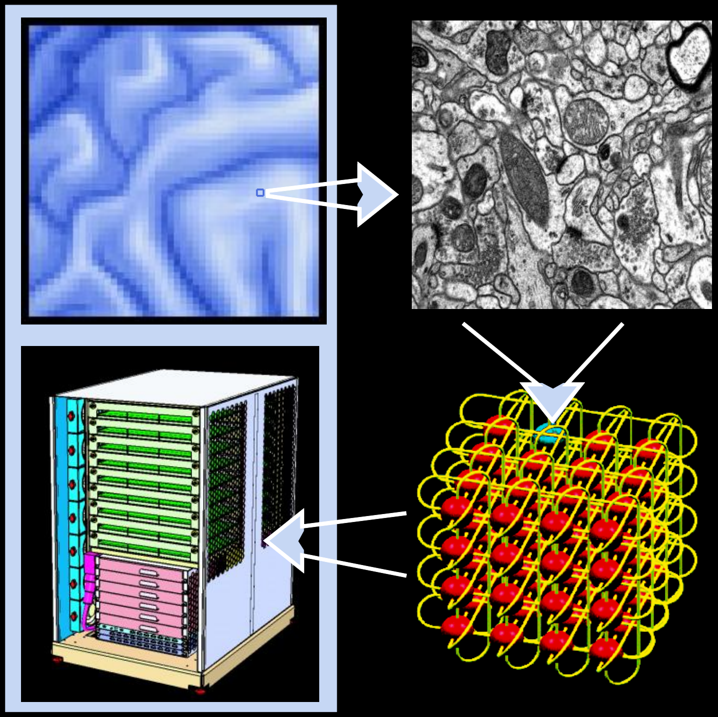
The question of axons...

- Failures of action potential propagation can occur at certain points along an axon, introducing uncertainty surrounding the signaling role of action potentials transmitted through otherwise reliable axons [1]
- Electrical synapses between axons can initiate action potentials without first depolarizing the axon initial segment [2]
- Action potentials may be generated by a mechanism that depends on the length of the axon (e.g., bursts of action potentials of a particular duration may be generated when a calcium spike from the cell body depolarizes an axon of a particular length [1])

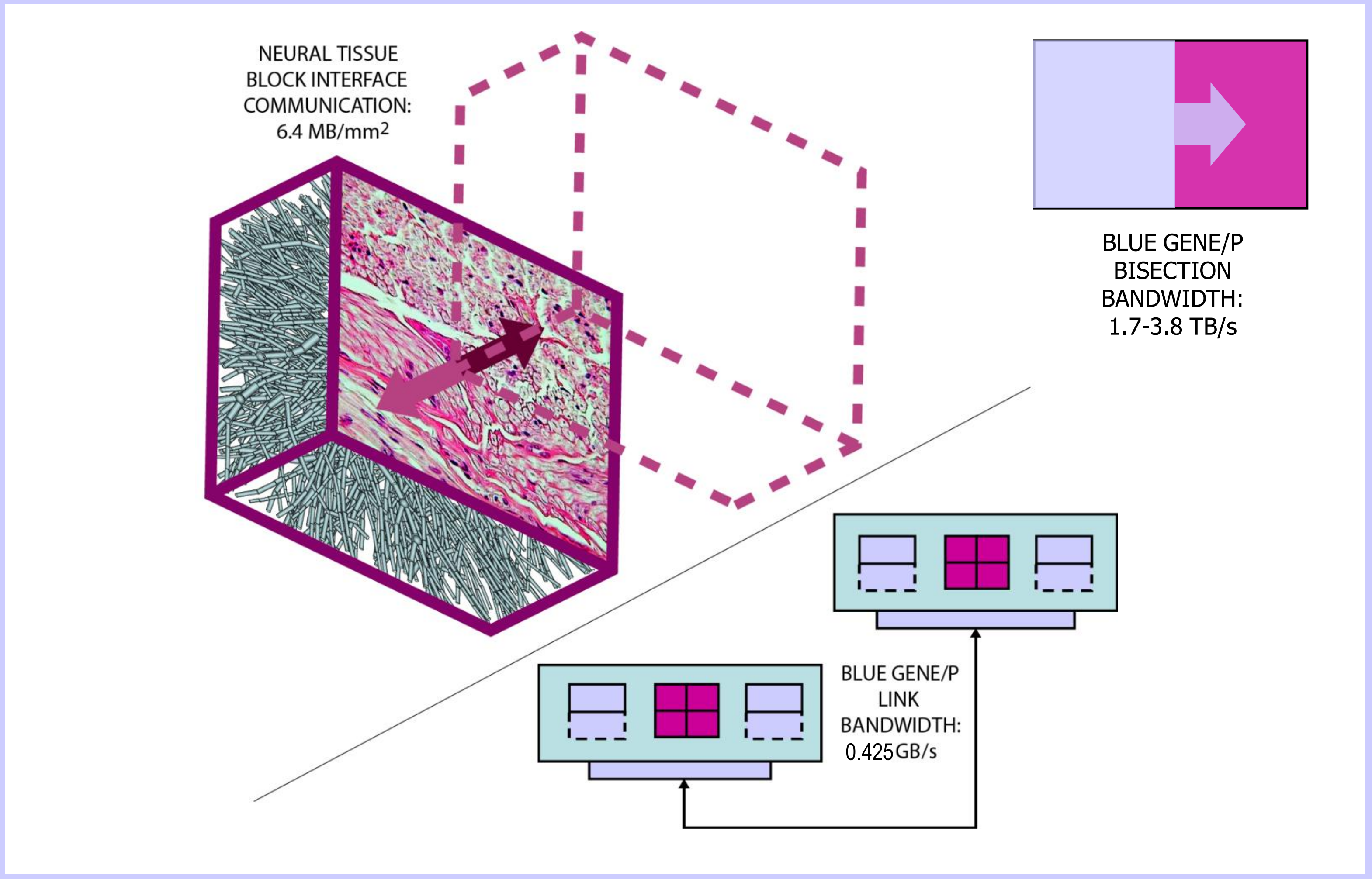
[1] A. Mathy, S. S. N. Ho, J. T. Davie, I. C. Duguid, B. A. Clark, and M. Husser. Encoding of oscillations by axonal bursts in inferior olive neurons. *Neuron*, 62(3):388–399, May 2009.

[2] D. Schmitz, S. Schuchmann, A. Fisahn, A. Draguhn, E. H. Buhl, E. Petrasch-Parwez, R. Dermietzel, U. Heinemann, and R. D. Traub. Axo-axonal coupling. a novel mechanism for ultrafast neuronal communication. *Neuron*, 31(5):831–840, Sep 2001.

Neural Tissue Simulation Scalability



Neural Tissue Simulation: Network Bandwidth



frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: xx August 2011
doi: 10.3389/fninf.2011.00015

An ultrascalable solution to large-scale neural tissue simulation

James Kozloski[1]* and John Wagner[2]

[1] *Biometaphorical Computing, Computational Biology Center, IBM Research Division, IBM T. J. Watson Research Center, Yorktown Heights, NY, USA*
[2] *Victorian Life Sciences Computation Initiative, IBM Research Collaboratory for Life Sciences – Melbourne, Carlton, VIC, Australia*

Edited by:
Markus Diesmann, RIKEN Brain Science Institute, Japan

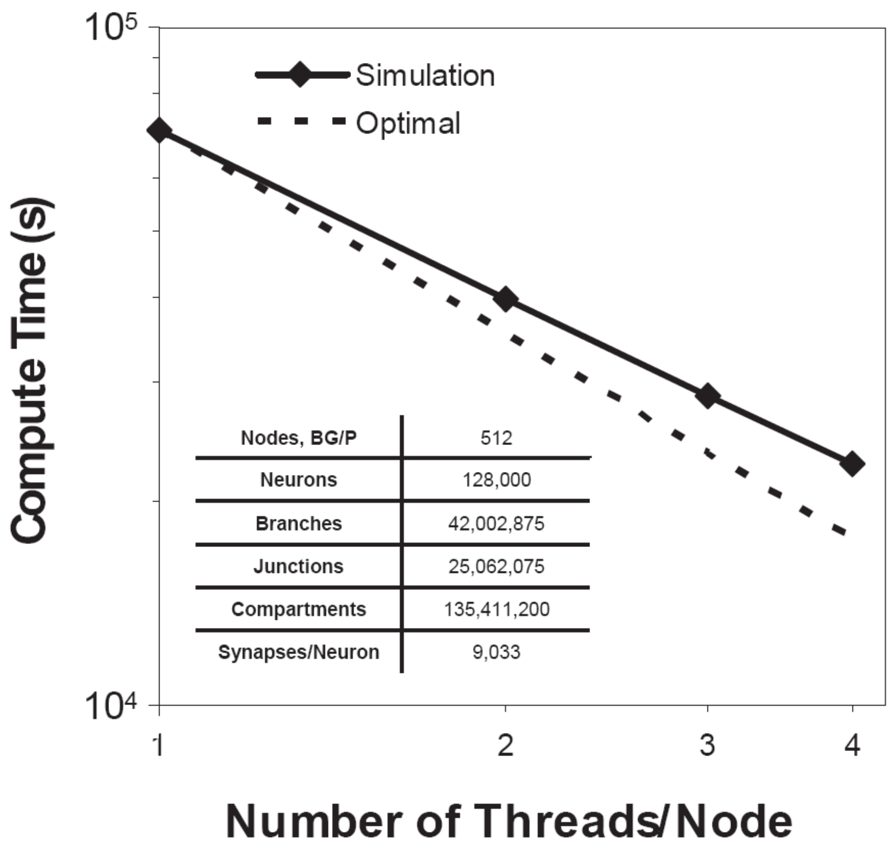
Reviewed by:
Michael Hines, Yale University, USA
Abigail Morrison, Bernstein Center Freiburg, Germany

***Correspondence:**
James Kozloski, Biometaphorical Computing, Computational Biology Center, IBM Research Division, IBM T. J. Watson Research Center, 1101 Kitchawan Road, Room 05-144, Yo Heights, NY, USA.
e-mail: kozloski@us.ibm.com

Neural tissue simulation extends requirements and constraints of previous neuronal and neural circuit simulation methods, creating a tissue coordinate system. We have developed a novel tissue volume decomposition, and a hybrid branched cable equation solver. The decomposition divides the simulation into regular tissue blocks and distributes them on a parallel multithreaded machine. The solver computes neurons that have been divided arbitrarily across blocks. We demonstrate thread, strong, and weak scaling of our approach on a machine with more than 4000 nodes and up to four threads per node. Scaling synapses to physiological numbers had little effect on performance, since our decomposition approach generates synapses that are almost always computed locally. The largest simulation included in our scaling results comprised 1 million neurons, 1 billion compartments, and 10 billion conductance-based synapses and gap junctions. We discuss the implications of our ultrascalable Neural Tissue Simulator, and with our results estimate requirements for a simulation at the scale of a human brain.

Keywords: neural tissue, simulation, parallel computing, distributed computing, Hodgkin–Huxley, numerical methods, ultrascalable, whole-brain

Thread Scaling

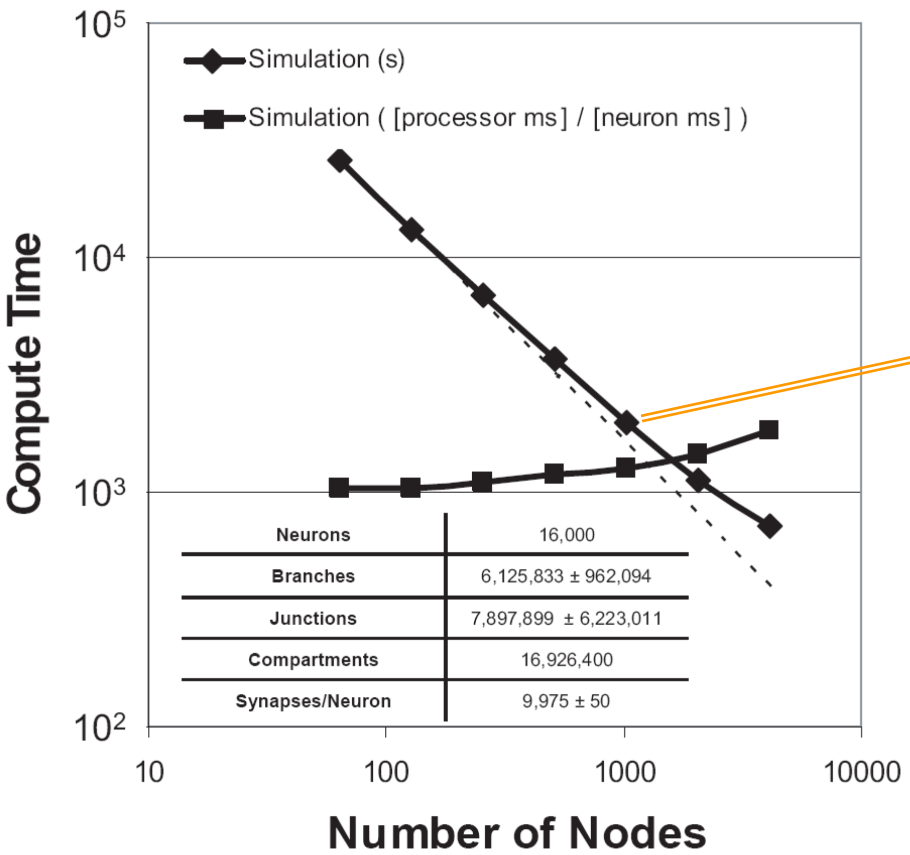


Nodes, BG/P	512
Neurons	128,000
Branches	42,002,875
Junctions	25,062,075
Compartments	135,411,200
Synapses/Neuron	9,033

Strong Scaling

- 10 µs time step
- 100 ms simulated physiology
- 160 million synapses

- 50 µs time step
- 1000 ms simulated physiology
- 13 million synapses



Neurons	16,000
Branches	6,125,833 ± 962,094
Junctions	7,897,899 ± 6,223,011
Compartments	16,926,400
Synapses/Neuron	9,975 ± 50

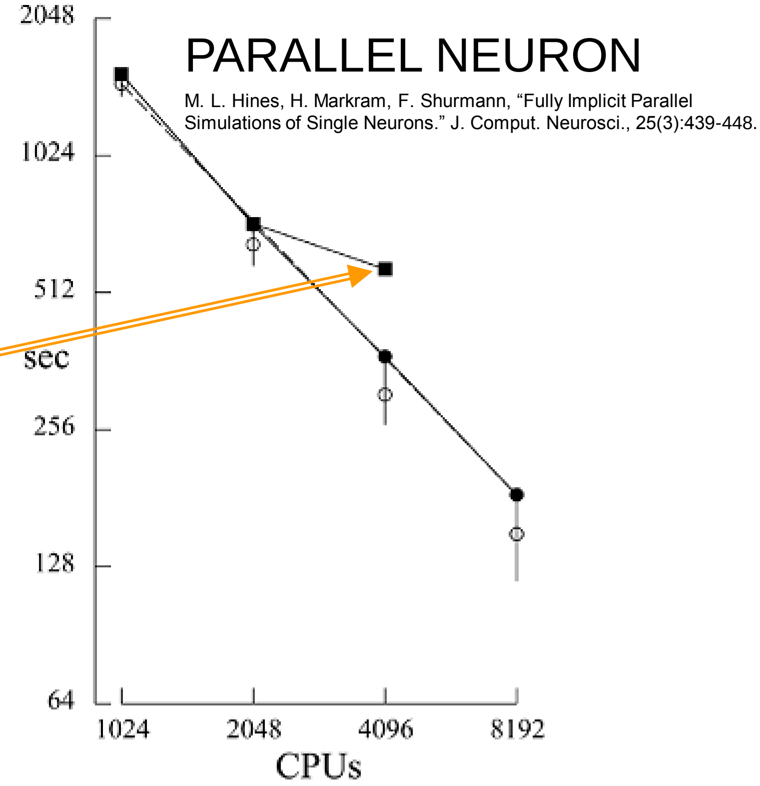
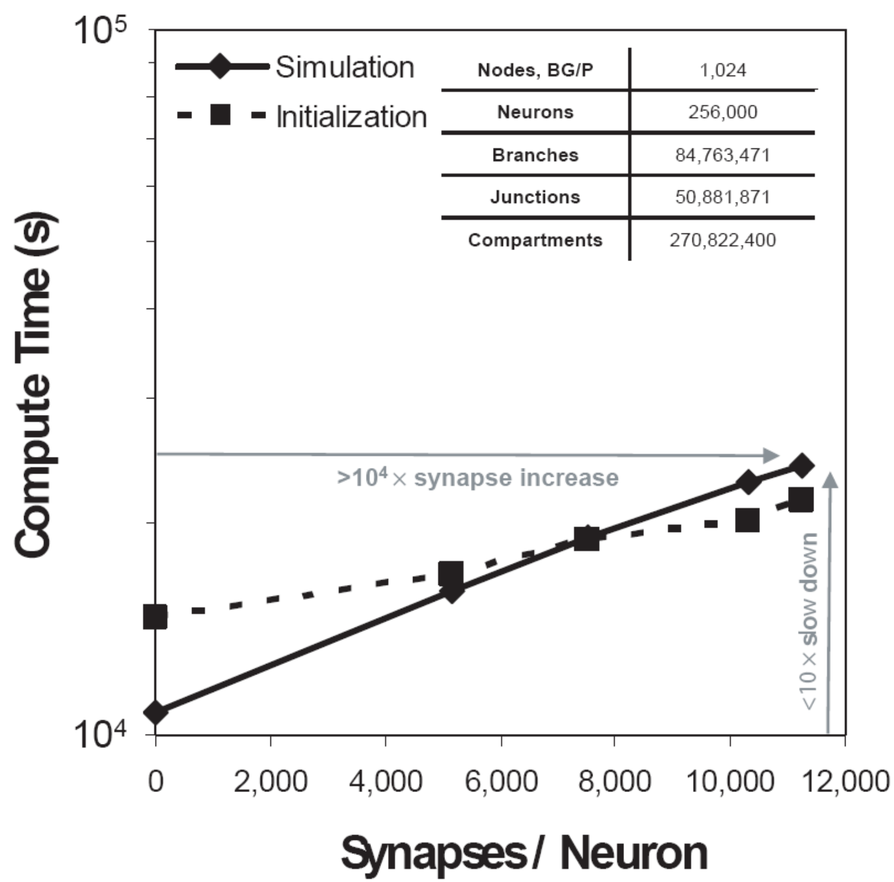


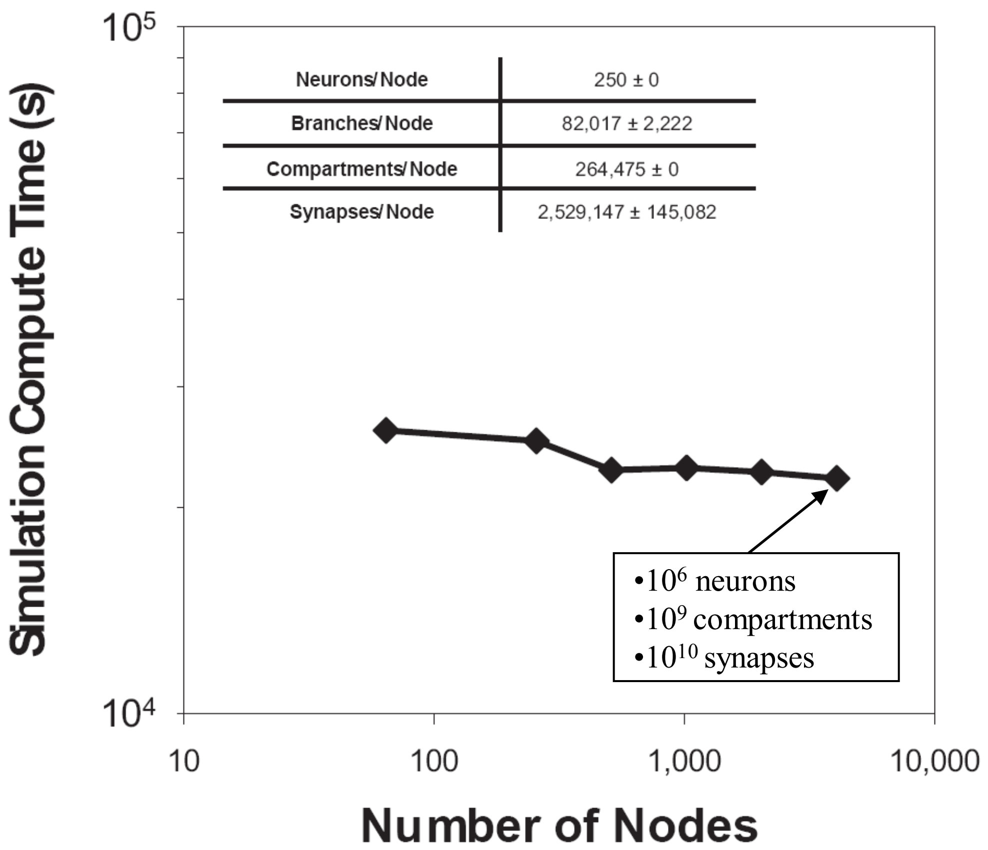
Figure 7.
Performance as a function of number of processors for the 10000 cell model.

Synapse Scaling

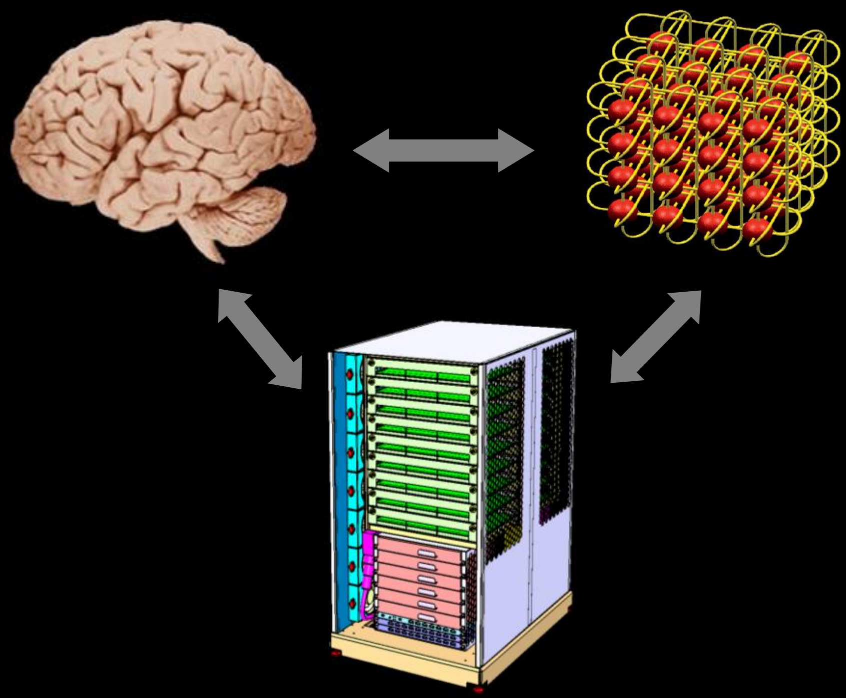


Nodes, BG/P	1,024
Neurons	256,000
Branches	84,763,471
Junctions	50,881,871
Compartments	270,822,400

Weak Scaling



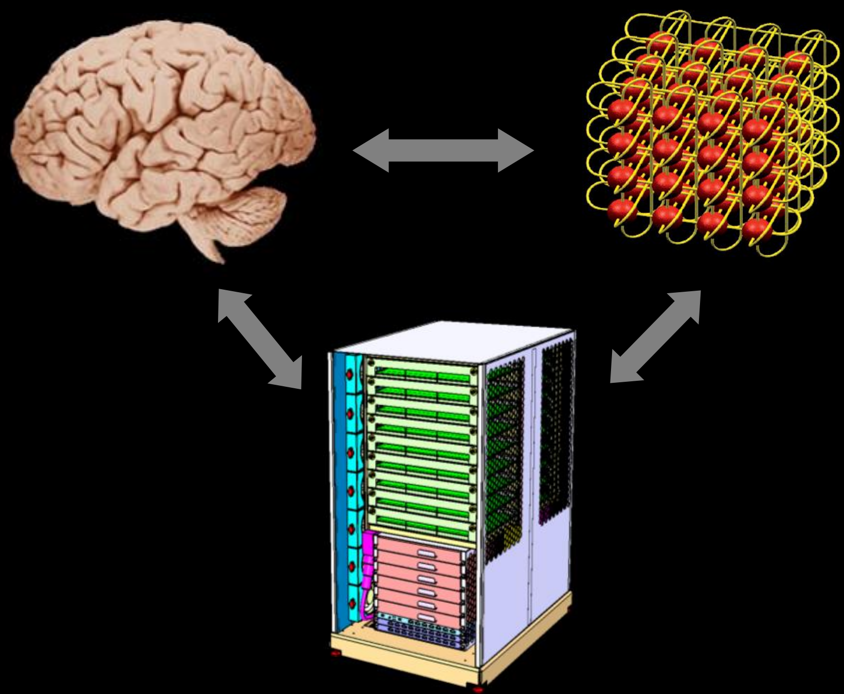
Feasibility of Human Brain-Scale Calculations



1 Liter of Neural Tissue In 10^7 Volumes

- 10^8 μm^3 / volume
- 10^6 compartments / volume, assuming 100 μm^3 / compartment
- 10^{10} flop / 50 μs simulation timestep / volume, assuming 10^4 flop / timestep / compartment (includes Hodgkin Huxley cable solution, plus 10 ion channels or synapses / compartment)
- 64kB communicated / volume face / timestep, assuming 32 bytes communicated / spanning compartment in each direction, and 10 μm^2 / compartment cross section
- 2.25 GB of memory / volume, including 250 MB of simulation overhead, plus 1.60 kB / compartment and 64 bytes / channel or synapse

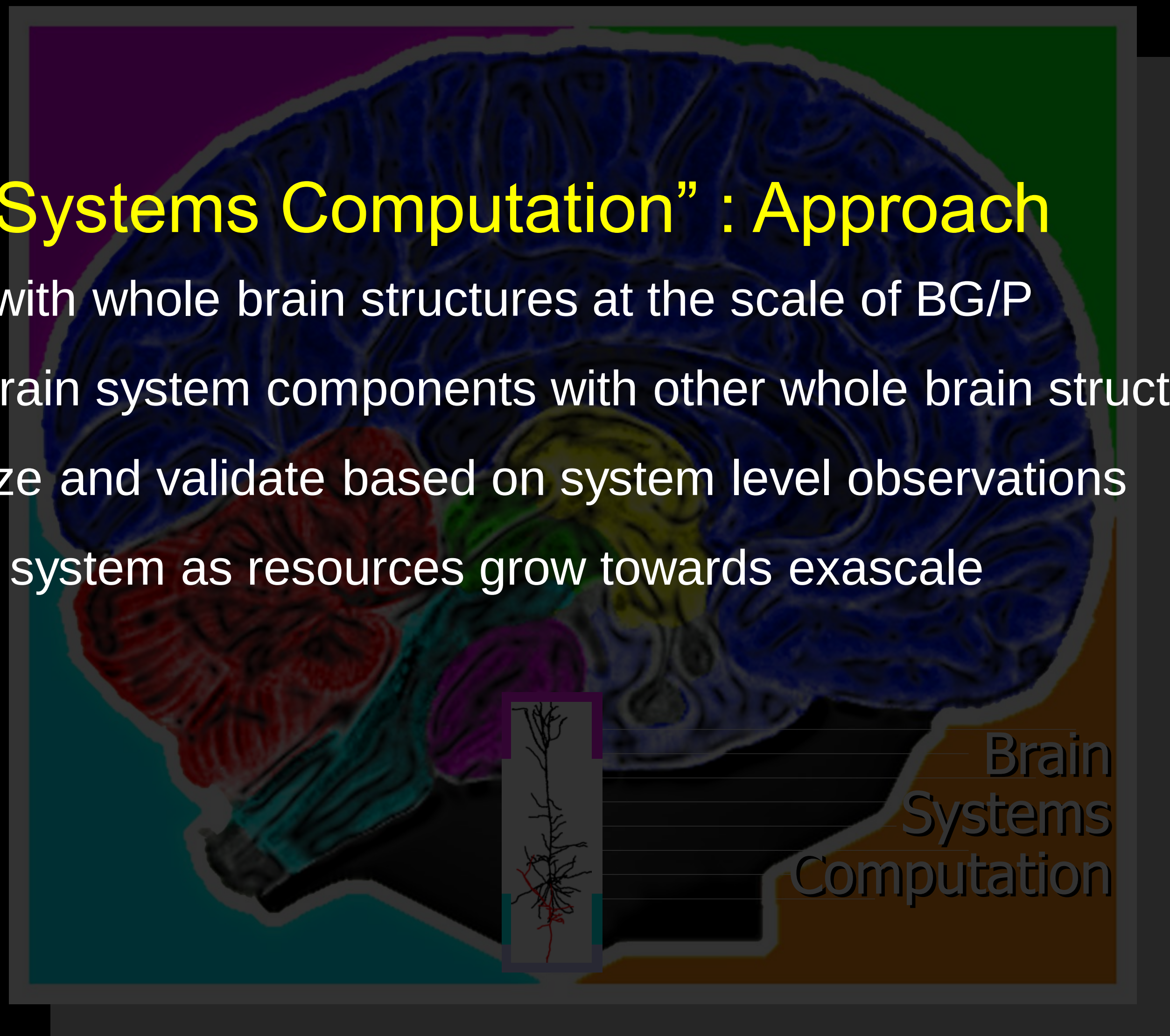
Feasibility of Human Brain-Scale Calculations



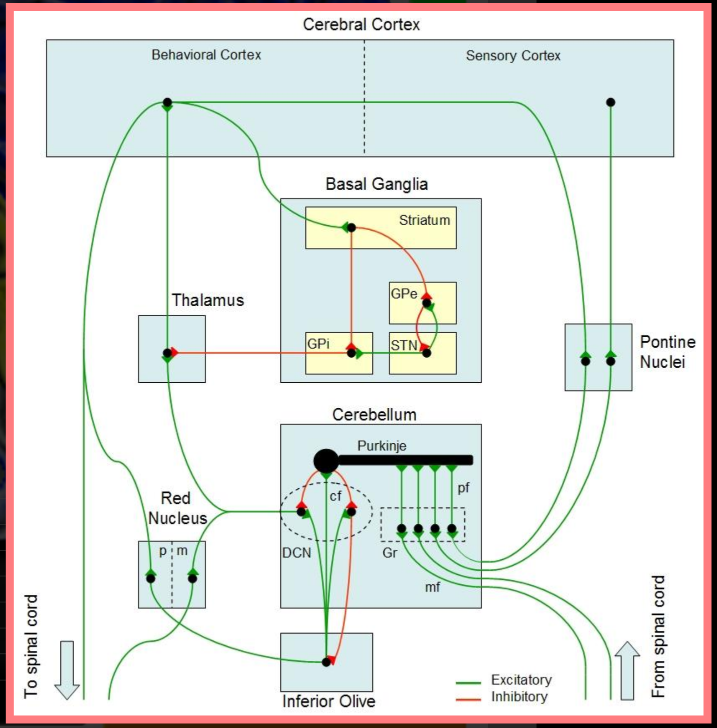
10^7 volumes mapped to a hypothetical Blue Gene with 10^7 computational nodes

- 10^9 flops / node (Blue Gene/P scale)
- 6.4 kB/s of link bandwidth (in each direction, to accommodate packet overhead, well within Blue Gene/P scale)
- 3 GB of memory / node (Blue Gene/P scale); 30 PB of total machine memory
- 6 GB/s memory bandwith, assuming all simulation state must be traversed 3 times in each simulation time step (Blue Gene/P scale)
- Simulation time to real time factor of 2 10^5 yielding a simulated time duration of 400 ms per day of computation, or 3 s per week of computation

- “Brain Systems Computation” : Approach
  - Start with whole brain structures at the scale of BG/P
  - Add brain system components with other whole brain structures
  - Analyze and validate based on system level observations
  - Scale system as resources grow towards exascale

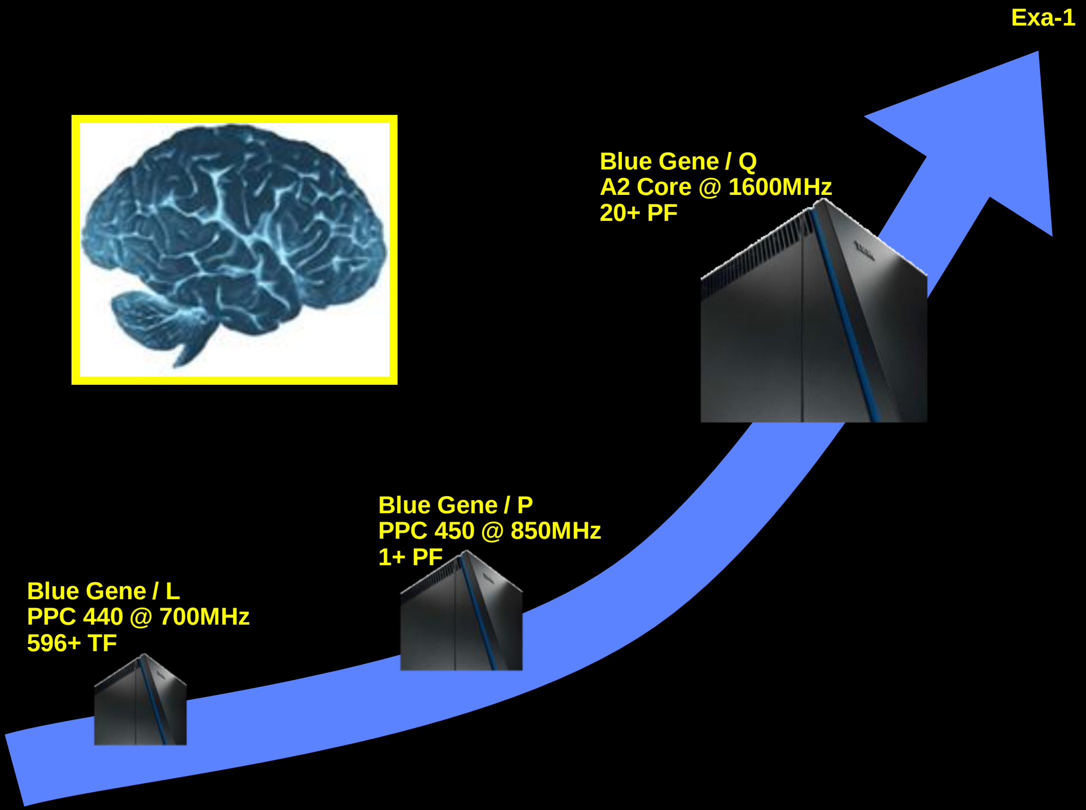


- **“Brain Systems Computation” : Example**
  - Inferior Olive
  - Deep Cerebellar Nuclei
  - Cerebellar Cortex
  - Thalamocortical
  - Pontine Nuclei
  - Rubrospinal
  - Basal Ganglia

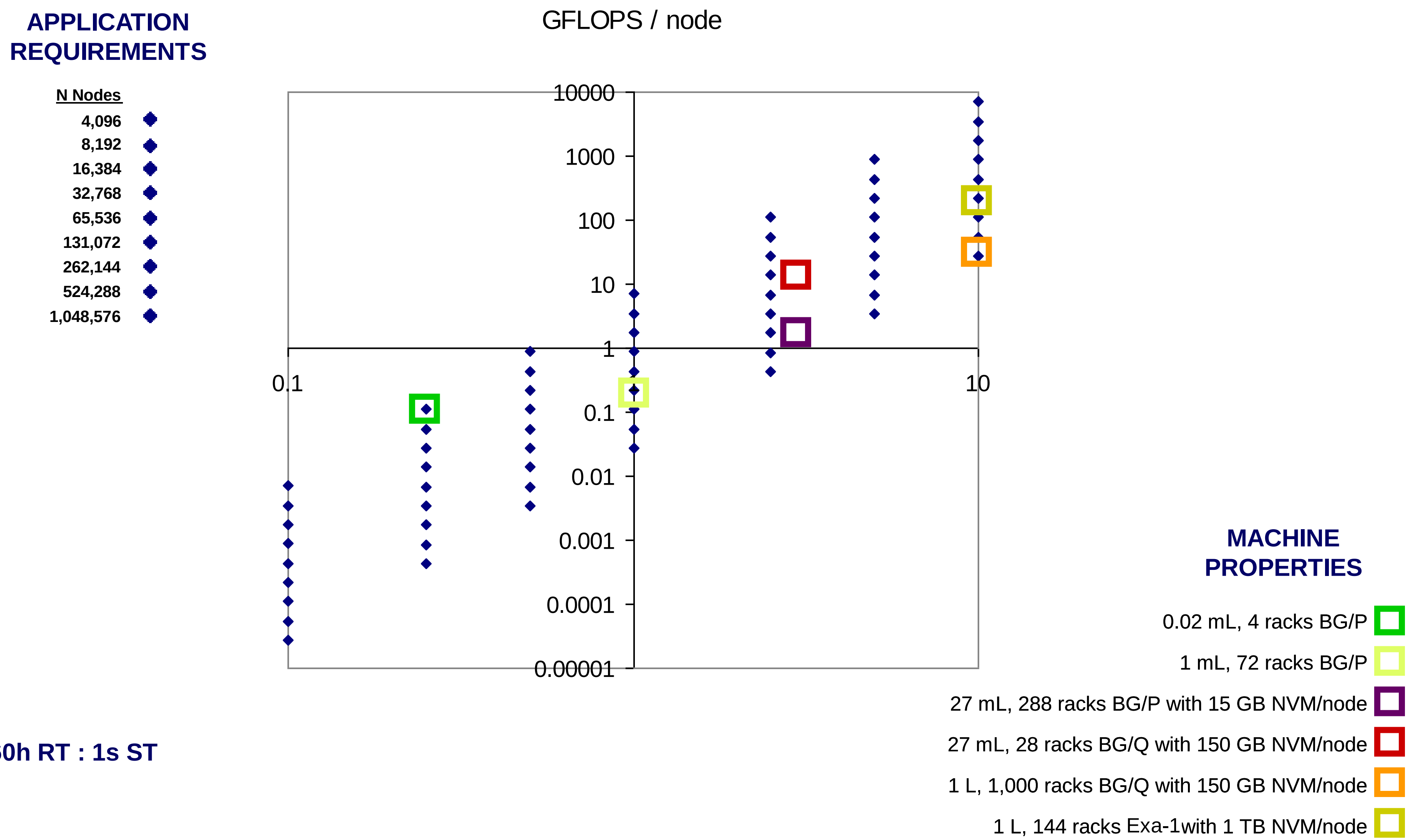


Mapping To Exascale

	GFLOPS	GFLOPS (App)	mem/node (GB)	LINK BW (GB/s)	mem BW (GB/s)
BG/L	5.6	1.12	0.5	0.175	5.6
BG/P	13.6	2.72	4	0.425	13.6
BG/Q	204.8	40.96	16	2.0	43
Exa-1 (Required)	1000	200	16	6.8	300



Speed Requirements By Tissue Size



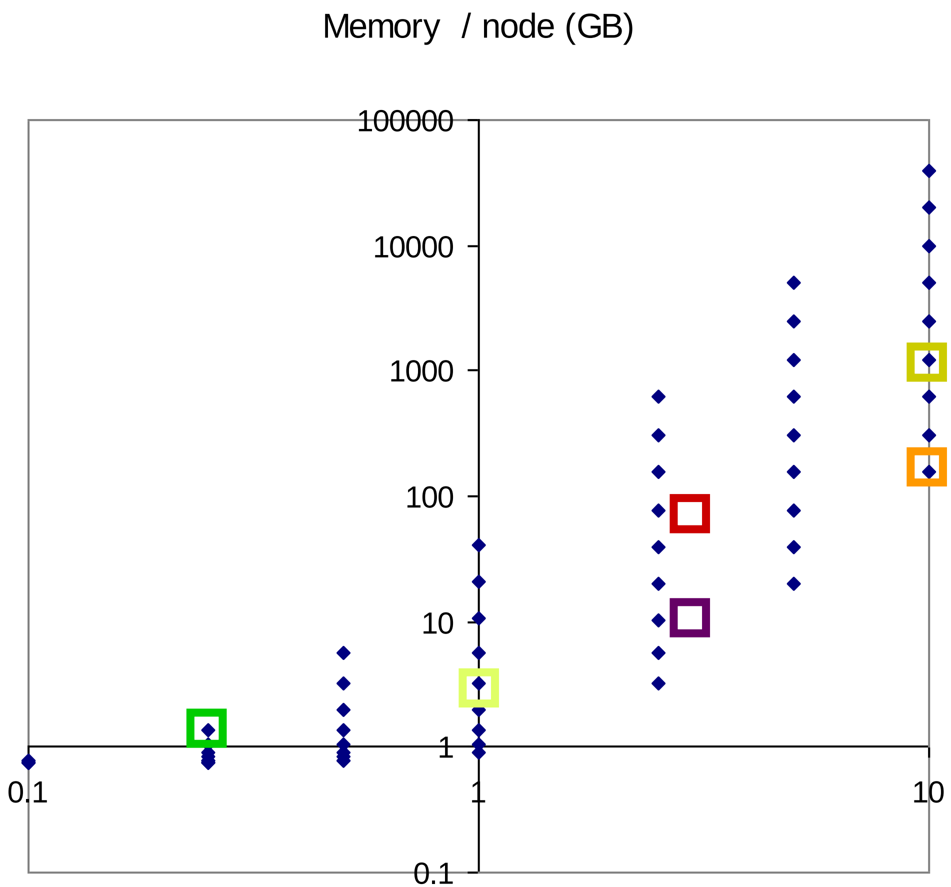
60h RT : 1s ST

Memory Requirements By Tissue Size (mL^{-3})

APPLICATION REQUIREMENTS

N Nodes

- 4,096
- 8,192
- 16,384
- 32,768
- 65,536
- 131,072
- 262,144
- 524,288
- 1,048,576



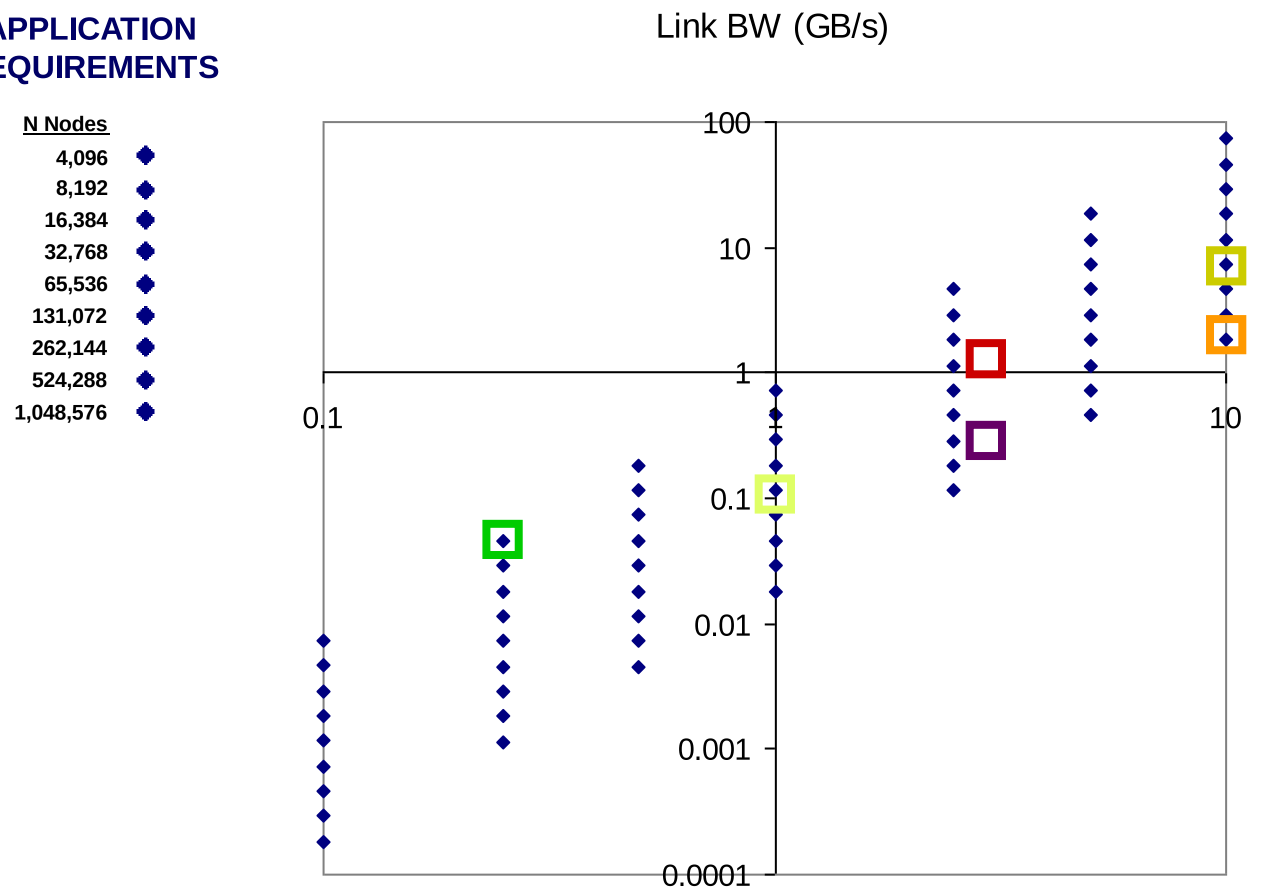
MACHINE PROPERTIES

- 0.02 mL, 4 racks BG/P
- 1 mL, 72 racks BG/P
- 27 mL, 288 racks BG/P with 15 GB NVM/node
- 27 mL, 28 racks BG/Q with 150 GB NVM/node
- 1 L, 1,000 racks BG/Q with 150 GB NVM/node
- 1 L, 144 racks Exa-1with 1 TB NVM/node

60h RT : 1s ST

Link BW Requirements By Tissue Size (mL^{-3})

APPLICATION REQUIREMENTS



MACHINE PROPERTIES

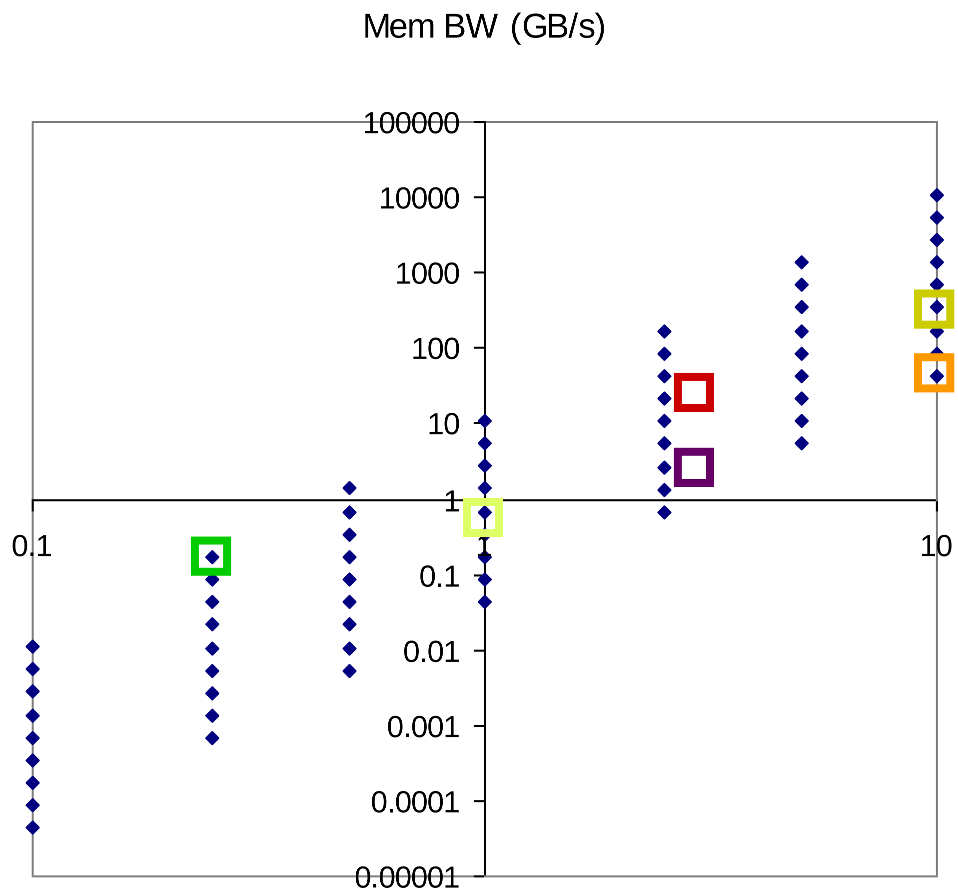
- 0.02 mL, 4 racks BG/P
- 1 mL, 72 racks BG/P
- 27 mL, 288 racks BG/P with 15 GB NVM/node
- 27 mL, 28 racks BG/Q with 150 GB NVM/node
- 1 L, 1,000 racks BG/Q with 150 GB NVM/node
- 1 L, 144 racks Exa-1with 1 TB NVM/node

60h RT : 1s ST

Memory BW Requirements By Tissue Size (mL^{-3})

APPLICATION REQUIREMENTS

N Nodes
- 4,096
- 8,192
- 16,384
- 32,768
- 65,536
- 131,072
- 262,144
- 524,288
- 1,048,576

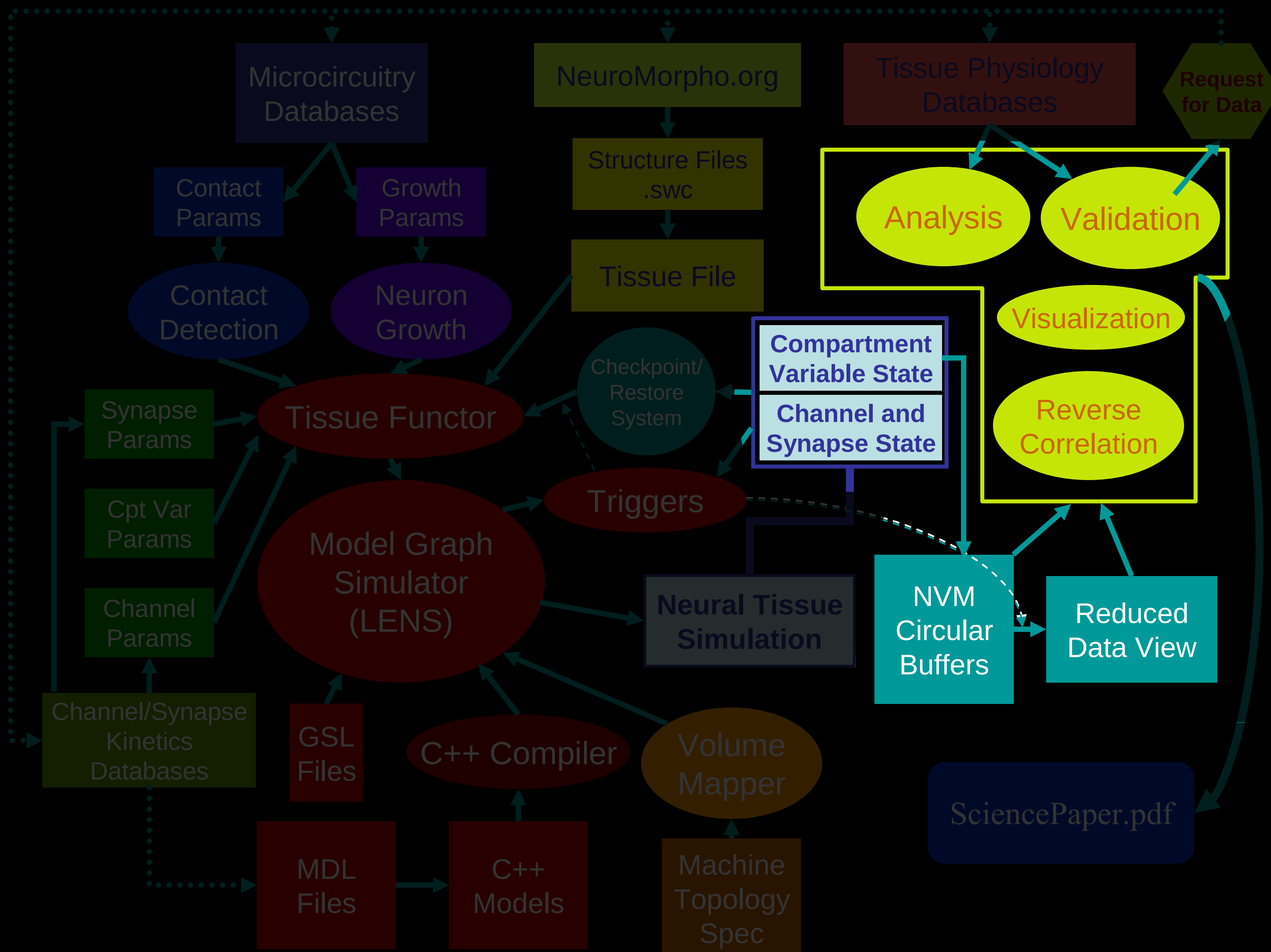


MACHINE PROPERTIES

- 0.02 mL, 4 racks BG/P
- 1 mL, 72 racks BG/P
- 27 mL, 288 racks BG/P with 15 GB NVM/node
- 27 mL, 28 racks BG/Q with 150 GB NVM/node
- 1 L, 1,000 racks BG/Q with 150 GB NVM/node
- 1 L, 144 racks Exa-1with 1 TB NVM/node

60h RT : 1s ST

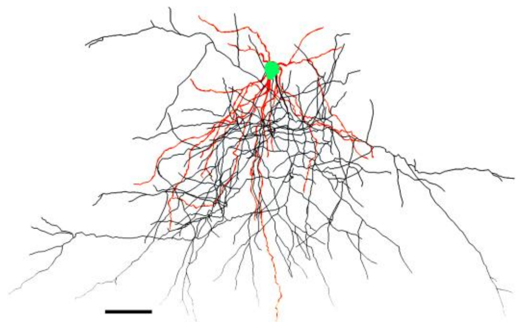
Neural Tissue Simulation Workflow



Neural Tissue Simulator: Current Storage Model

- Model Graph Simulator creates special data collection graph elements, specified in MDL, declared in GSL
- Data collectors’ connections to specific neural tissue elements declared in GSL according to key specification
- At runtime, the existence of elements on each compute node matching key is established, and data collectors created
- Data collectors write to own files upon satisfaction of predicates of GSL-specified triggers (e.g., iteration)

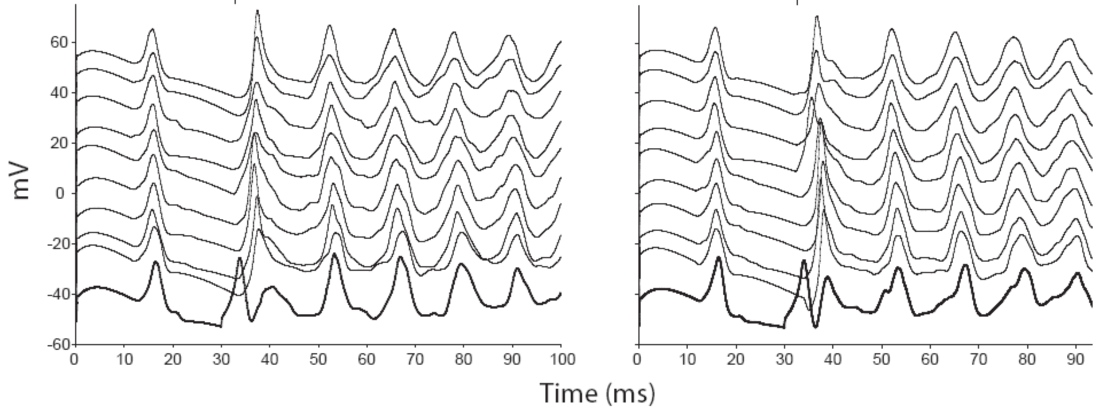
Physiological Results: “Neocortical” Simulation

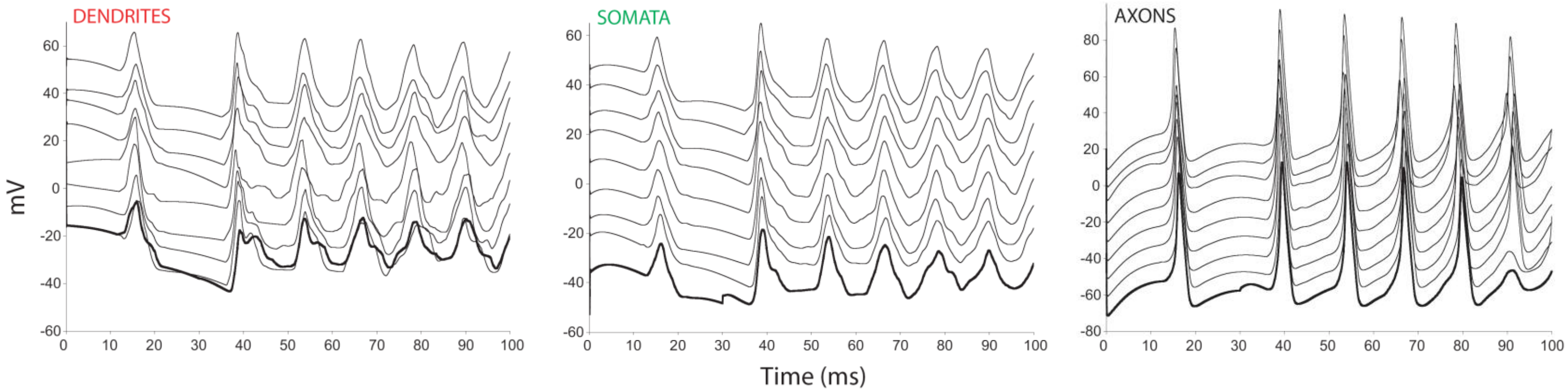


Neurons	16,000
AMPA	115,647,522
GABAA	44,352,919
Connexons	137,338
Synapses/Neuron	10,004

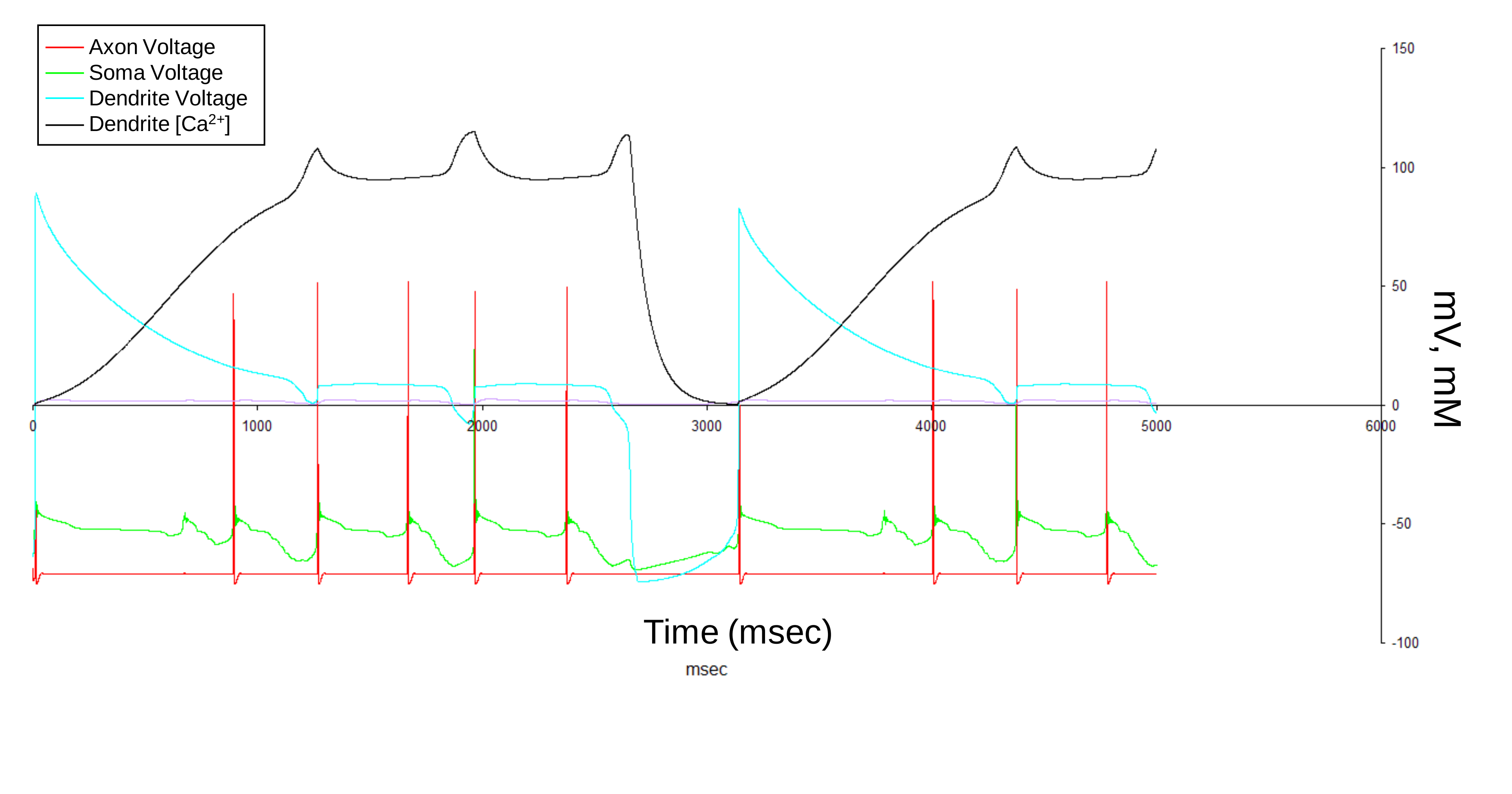
Neurons	1,024,000
AMPA	7,777,621,545
GABAA	2,142,318,617
Connexons	7,244,700
Synapses/Neuron	9,691

Neurons	1,024,000
AMPA	8,186,972,360
GABAA	2,255,068,948
Connexons	7,626,124
Synapses/Neuron	10,201

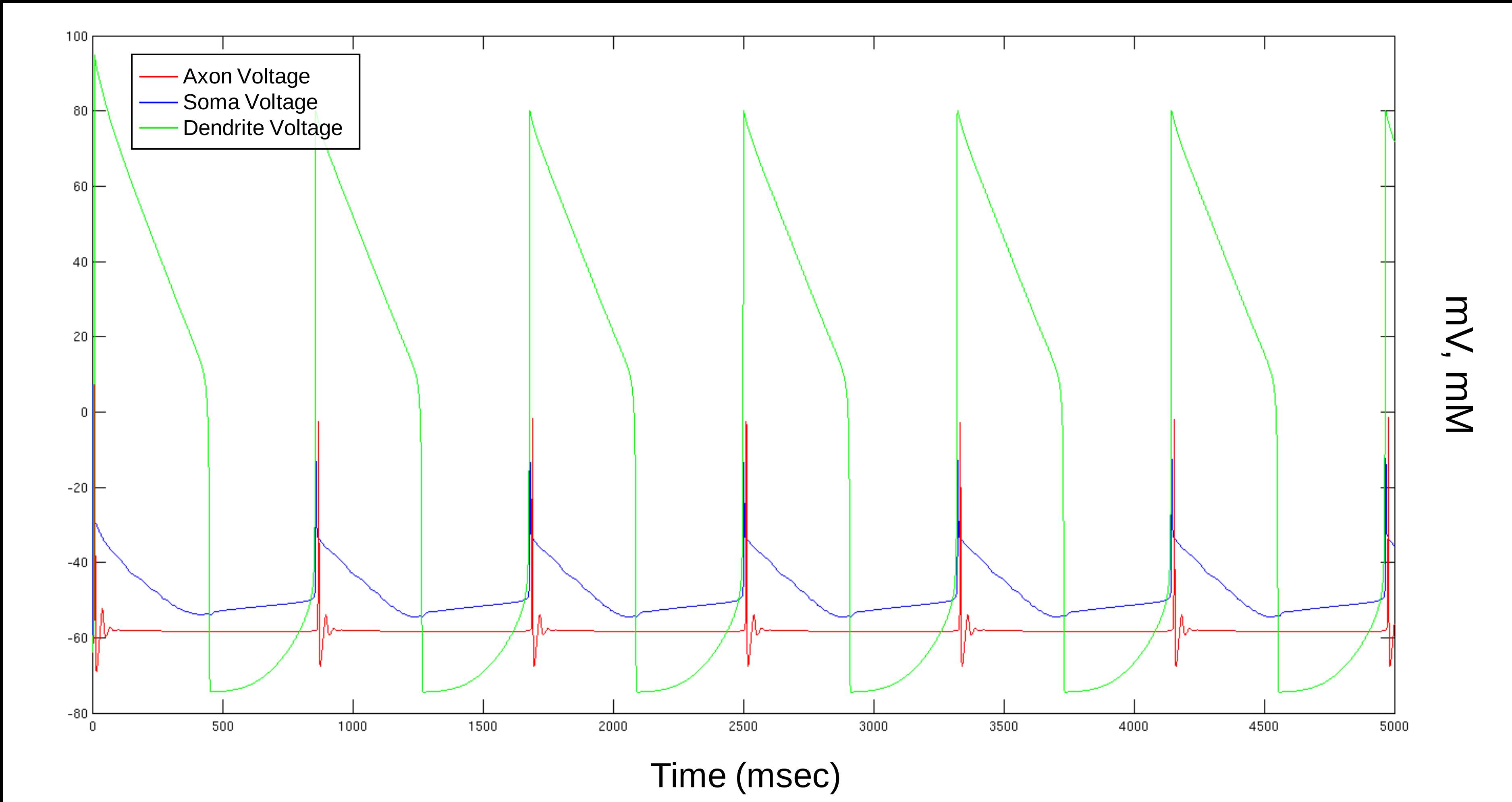




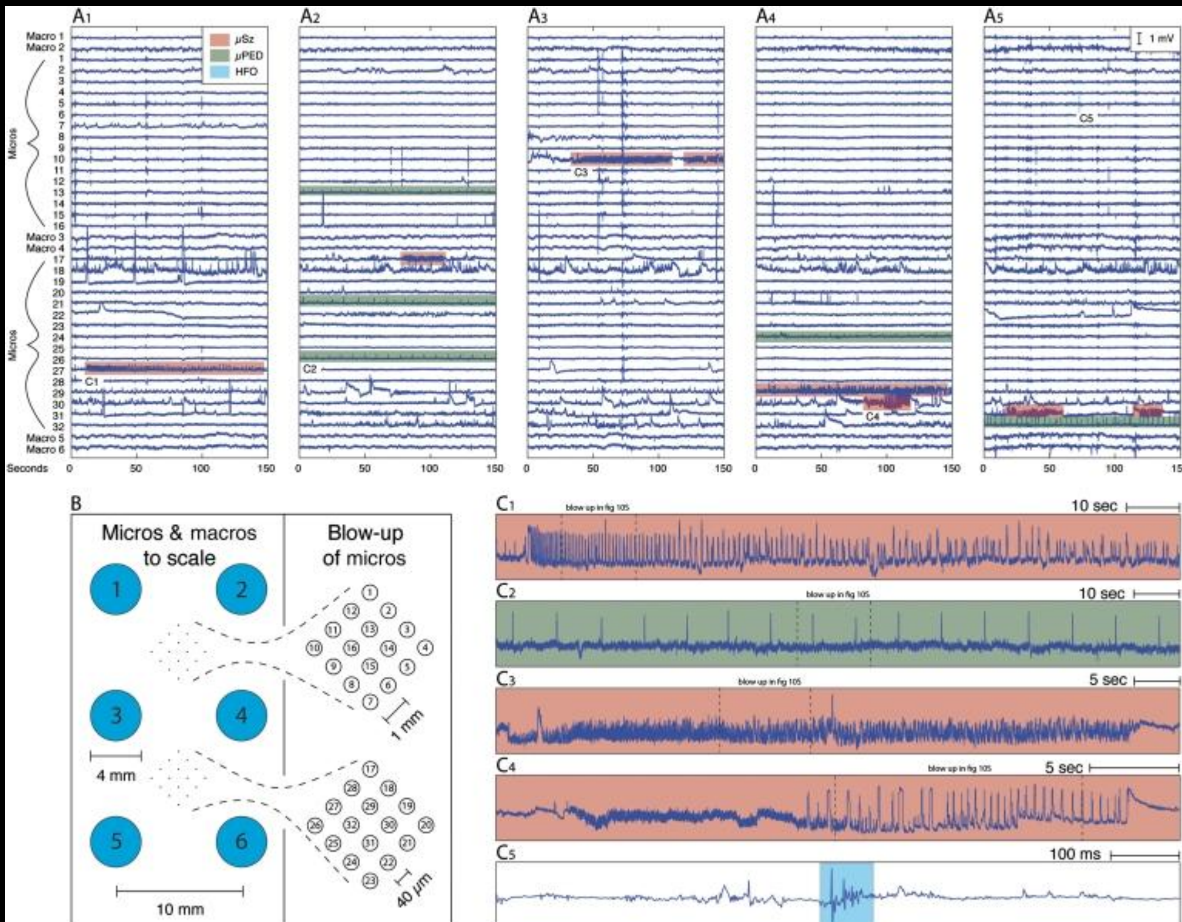
Physiological Results: Inferior Olive Simulation



Physiological Results: Inferior Olive Simulation



Neural Current Analyzer: I/O Bound Application

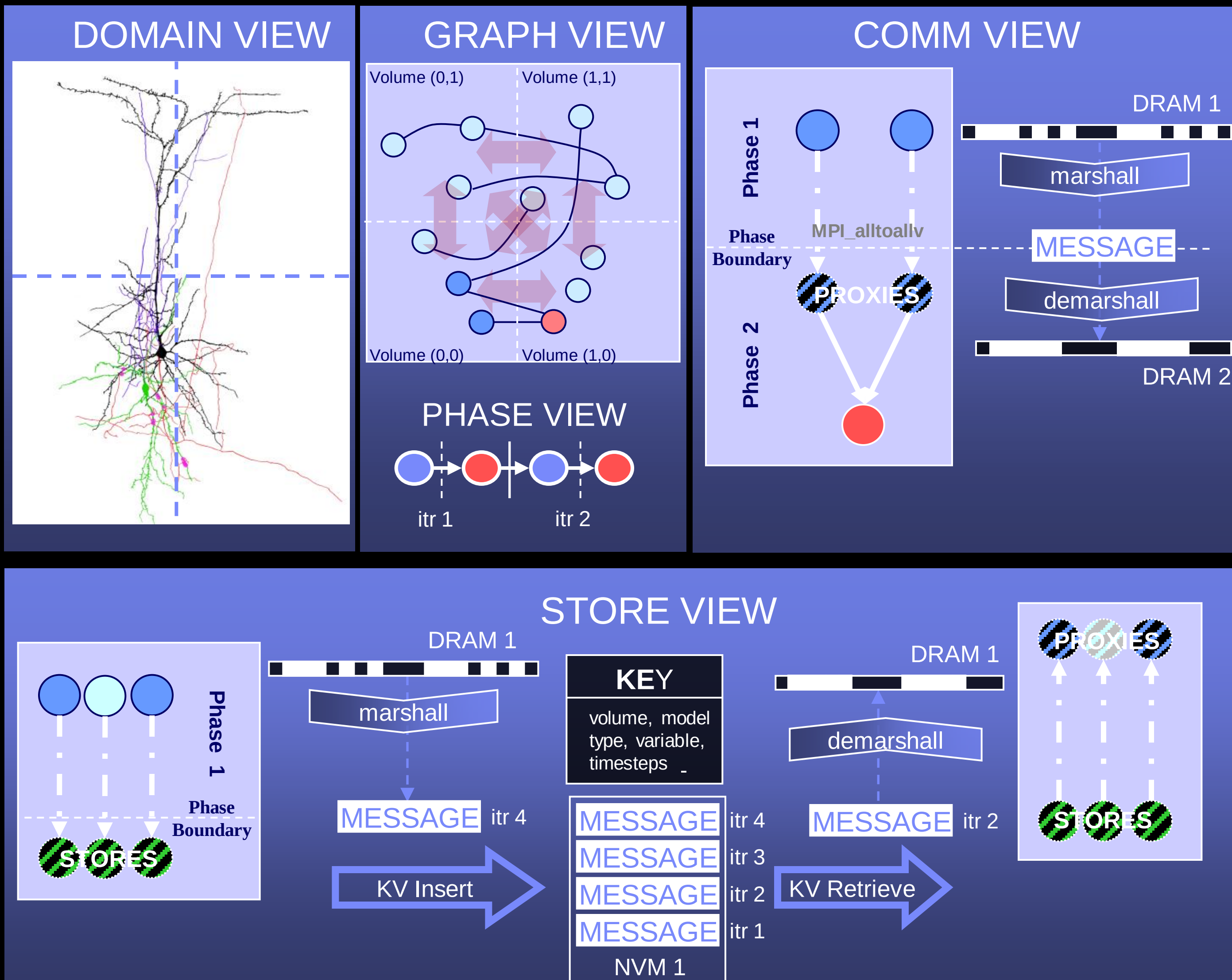


Microseizures and the spatiotemporal scales of human partial epilepsy

Matt Stead,[1] Mark Bower,[1] Benjamin H. Brinkmann,[1] Kendall Lee,[2] W. Richard Marsh,[2] Fredric B. Meyer,[2] Brian Litt,[3,4] Jamie Van Gompel,[2] and Greg A. Worrell[1]

- Standard neural analysis methods lack the ability to store massive data quickly, perform post-simulation analyses to discover causation
- Standard demonstration-driven approaches record exemplars to test hypotheses inherent in simulation design
- A discovery-driven approach and data analysis framework would allow the quantitative identification of causes of network phenomena in neural tissue

Neural Current Analyzer: Proposed Storage Model



- Simulation framework creates messages at phase boundaries to be sent across node boundaries to initialize proxies for next phase
- Storage model uses framework generated messages to store state for
  - Checkpoint rollback
  - Triggered analysis
  - Causality analysis
  - Visualization
- Framework marshalling generates key/value inserts saving graph-node state
- Framework uses demarshalling infrastructure to generate key/value/offset retrieve to initialize graph-node proxy state as required by analysis algorithm

Neural Current Analyzer: Application Areas

- Epilepsy
  - Quantitative analysis of current motifs responsible for epileptogenesis
  - Identification of specific mechanisms of observed channelopathies
  - Drug candidate identification
  - Brain stimulation protocols and parameters

- Predictive EEG modeling
  - Solve EEG in FEM model of extracellular fields for each current motif identified
  - Use EEG “motifs” as building blocks (components) for a forward model of real EEG recordings
  - Relate EEG to behavior to predict functional brain (current) states

Acknowledgements

- Blake Fitch, Volume Decomposition Design, Neural Current Analyzer on BGAS
- John Wagner, Distributed Branched Hodgkin-Huxley on Blue Gene
- Charles Peck, Model Graph Simulator Architect, Manager of Biometaphorical Computing, IBM
- Benjamin Torben-Nielsen (EPFL) and Heraldo Memelli (Stony Brook), Constrained Diffusion Model of Neuron Morphogenesis
- Mike Pitman, Molecular Dynamics abstractions of neural growth and development
- Maria Eleftheriou, Model Graph Simulator Testing

Brain Systems Computation