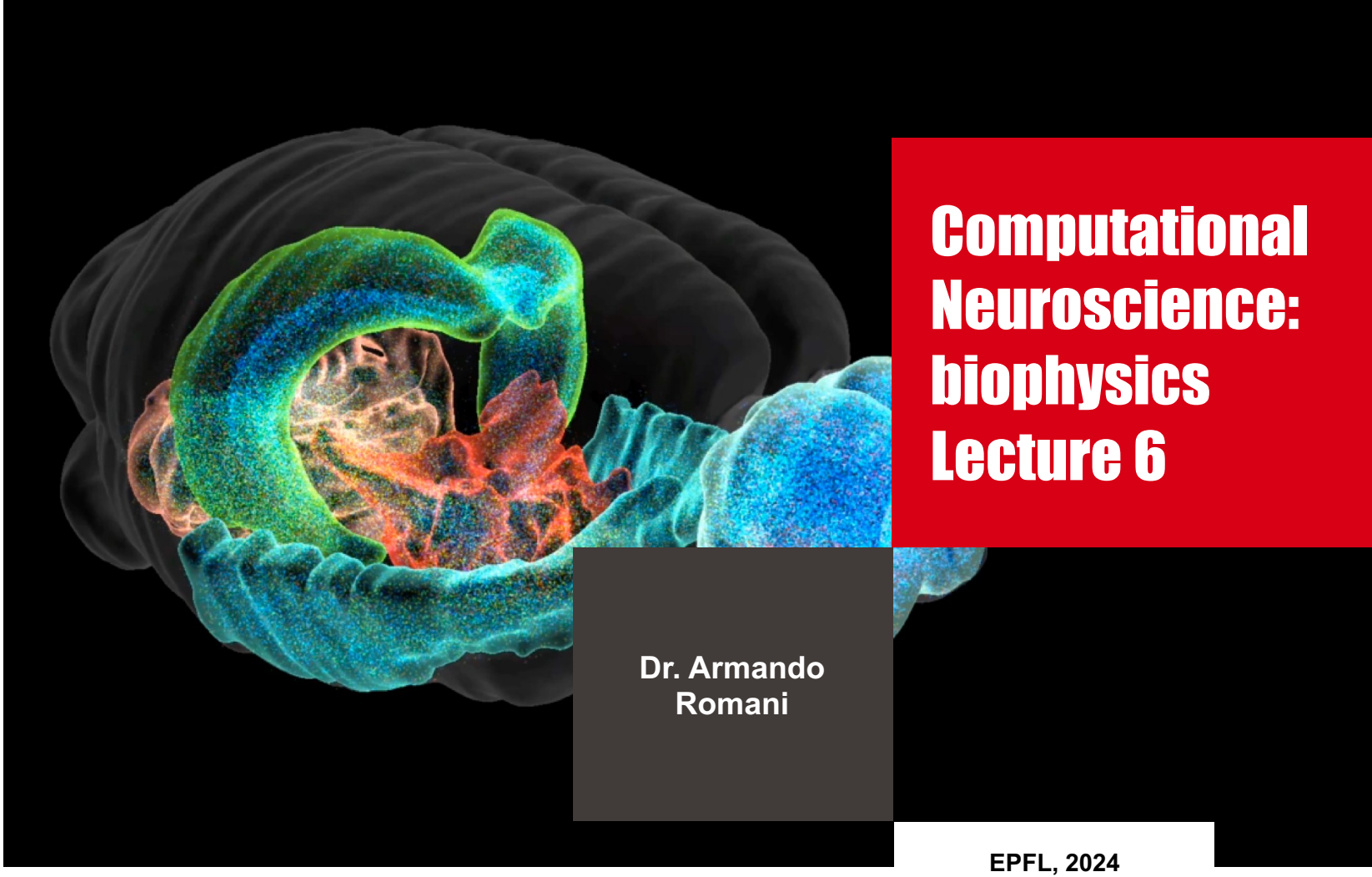


Computational Neuroscience: biophysics Lecture 6

Dr. Armando Romani

EPFL, 2024

Electrical Behavior of Neurons II

Lecture Overview

- Scope
- Approaches
- Applications

Lecture Overview

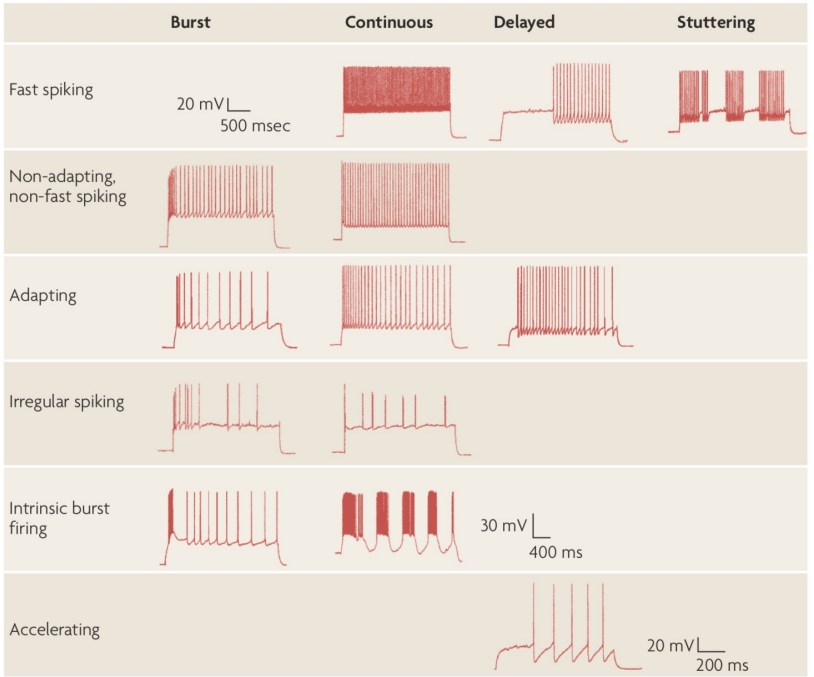
- **Scope**
- Approaches
- Applications

Variety of electrical behaviours

Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex

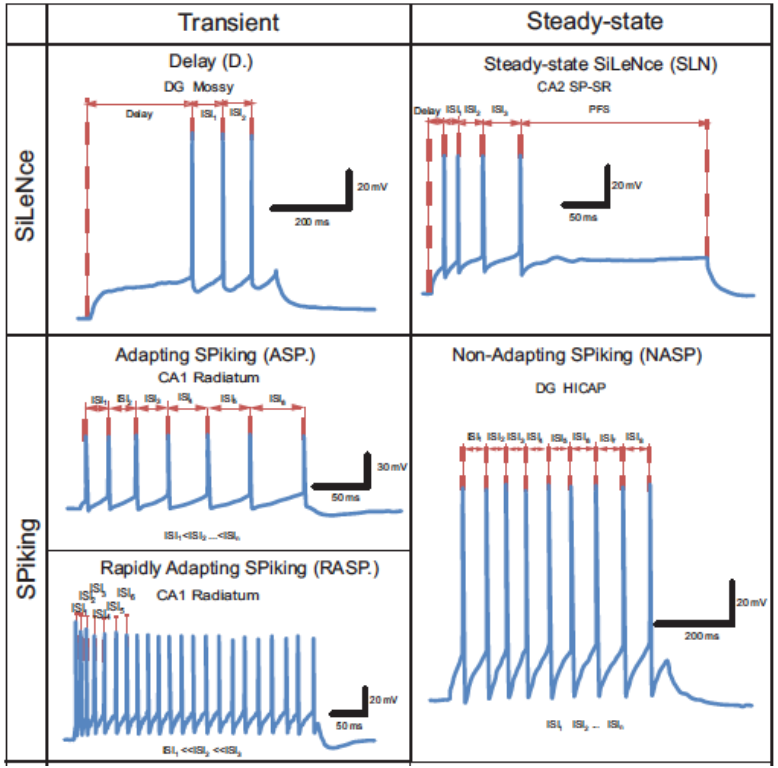
*The Petilla Interneuron Nomenclature Group (PING)**

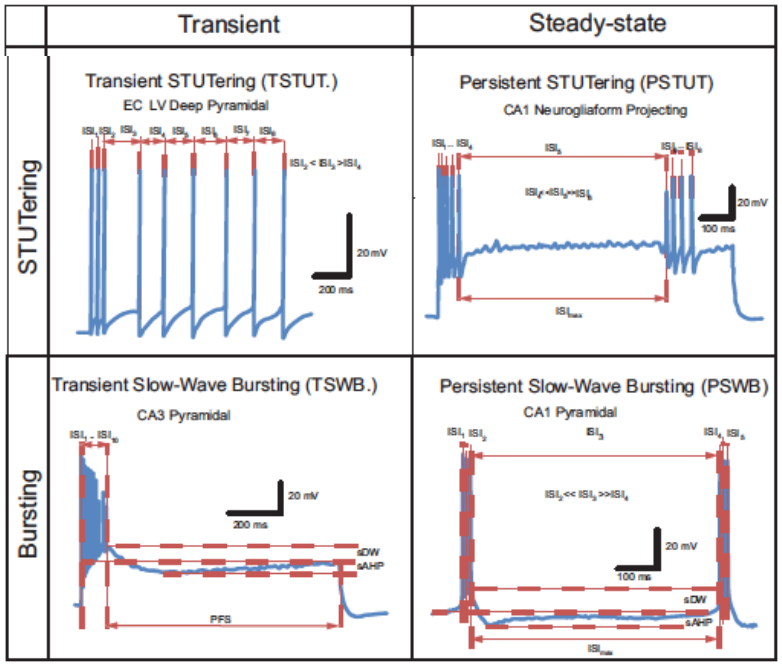
Abstract | Neuroscience produces a vast amount of data from an enormous diversity of neurons. A neuronal classification system is essential to organize such data and the knowledge that is derived from them. Classification depends on the unequivocal identification of the features that distinguish one type of neuron from another. The problems inherent in this are particularly acute when studying cortical interneurons. To tackle this, we convened a representative group of researchers to agree on a set of terms to describe the anatomical, physiological and molecular features of GABAergic interneurons of the cerebral cortex. The resulting terminology might provide a stepping stone towards a future classification of these complex and heterogeneous cells. Consistent adoption will be important for the success of such an initiative, and we also encourage the active involvement of the broader scientific community in the dynamic evolution of this project.



Ascoli et al., 2008

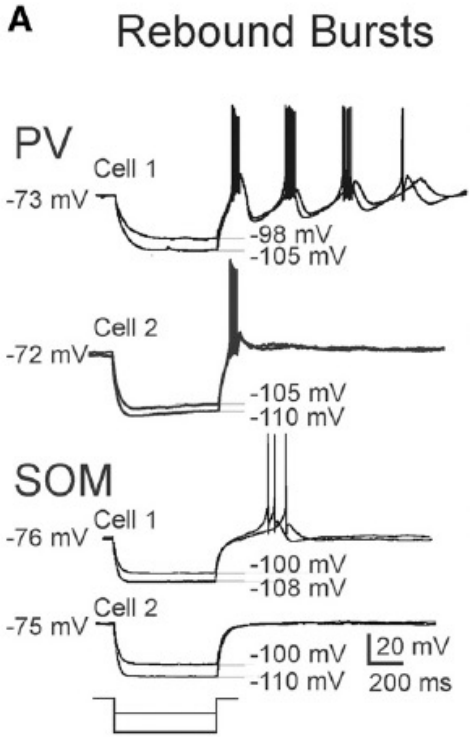
Variety of electrical behaviours

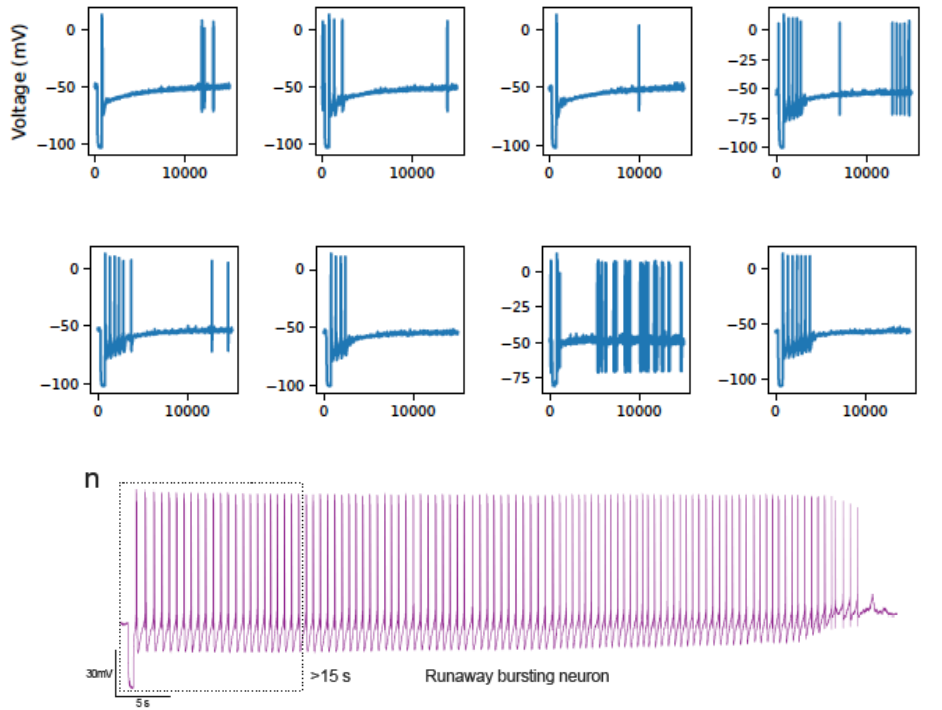




Komendantov et al., 2019

Variety of electrical behaviours





Clemente-Perez et al., 2017

RT cells, unpublished

Variety of electrical behaviours

MS non-cholinergic neurons

MS cholinergic neurons

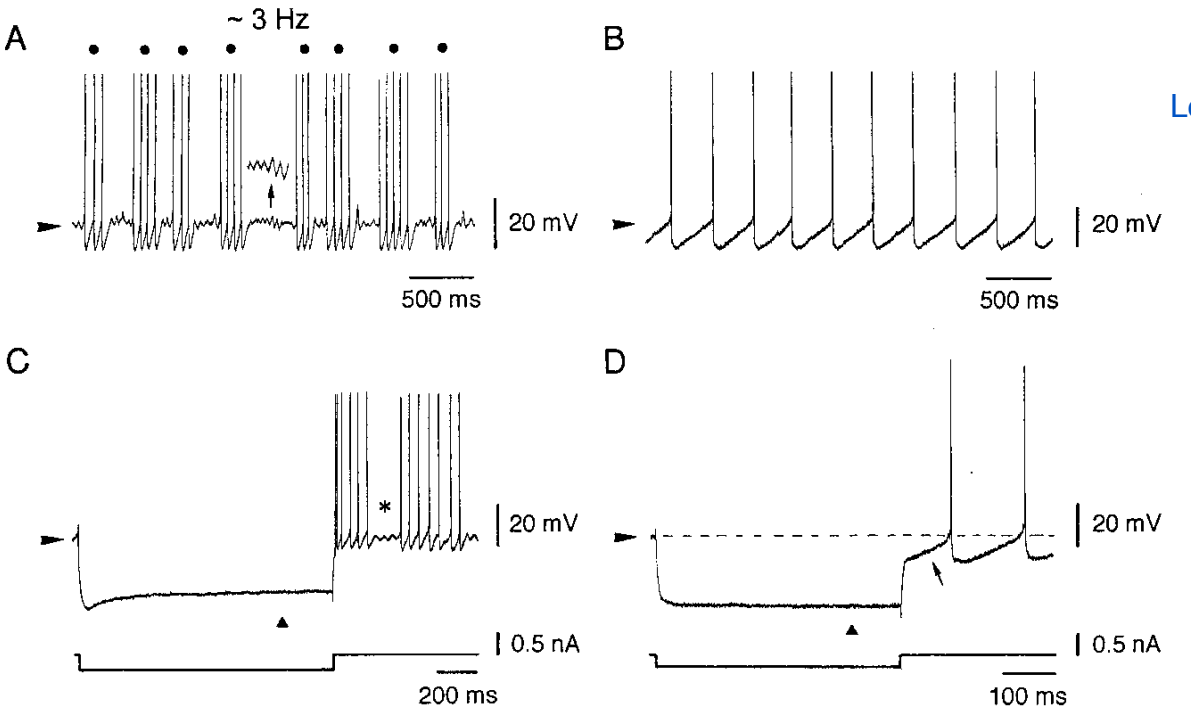
Spontaneous firing

Rhythmic firing

Sub-threshold oscillations

Low-rate firing

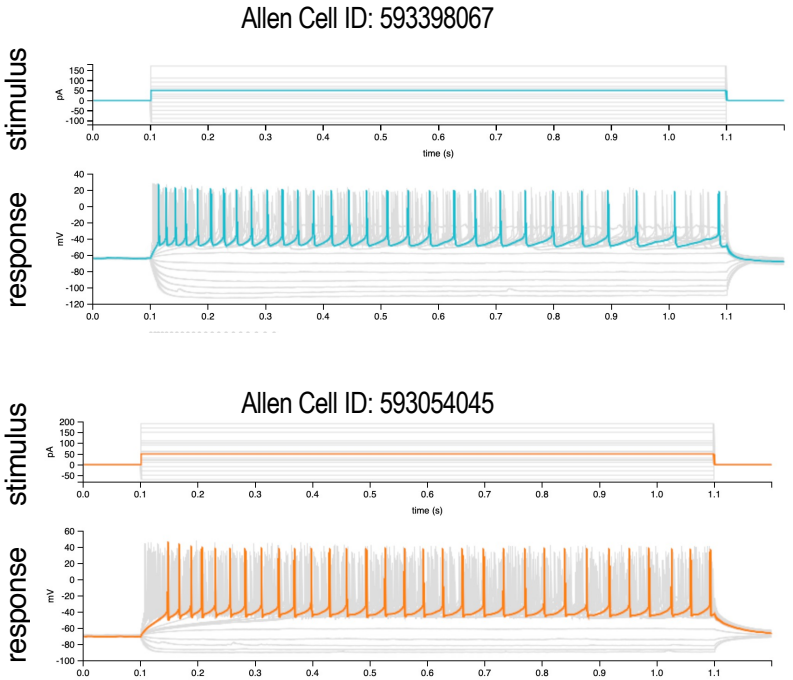
Hyperpolarizing current



Serafin et al., 1996

Neuron Class vs. Individual Neuron

- We have seen examples of classification (more or less quantitative/automatic) of electrical properties
- Taken singularly, each neuron shows electrical properties potentially unique as in the case of the morphologies
- This is relevant because we can decide to model individual neurons or neuron classes



Go beyond HH model

- There are a variety of ion channels (lecture 4) that can create the different electrical behaviors.
- Despite we started to characterize genetically-defined ion channels, it is still convenient to consider classes of channels
- The different ion channels can be modelled using HH formalism (lecture 4).
- Dendrites can influence somatic behavior (lecture 5), and soma can influence the dendritic behavior (see below).
- Even if synaptic inputs end up in the soma, dendrites combine (integrate) them in a complex manner (dendritic computation) (see below).

EPFL

Different ion channels contribute to different electrical features

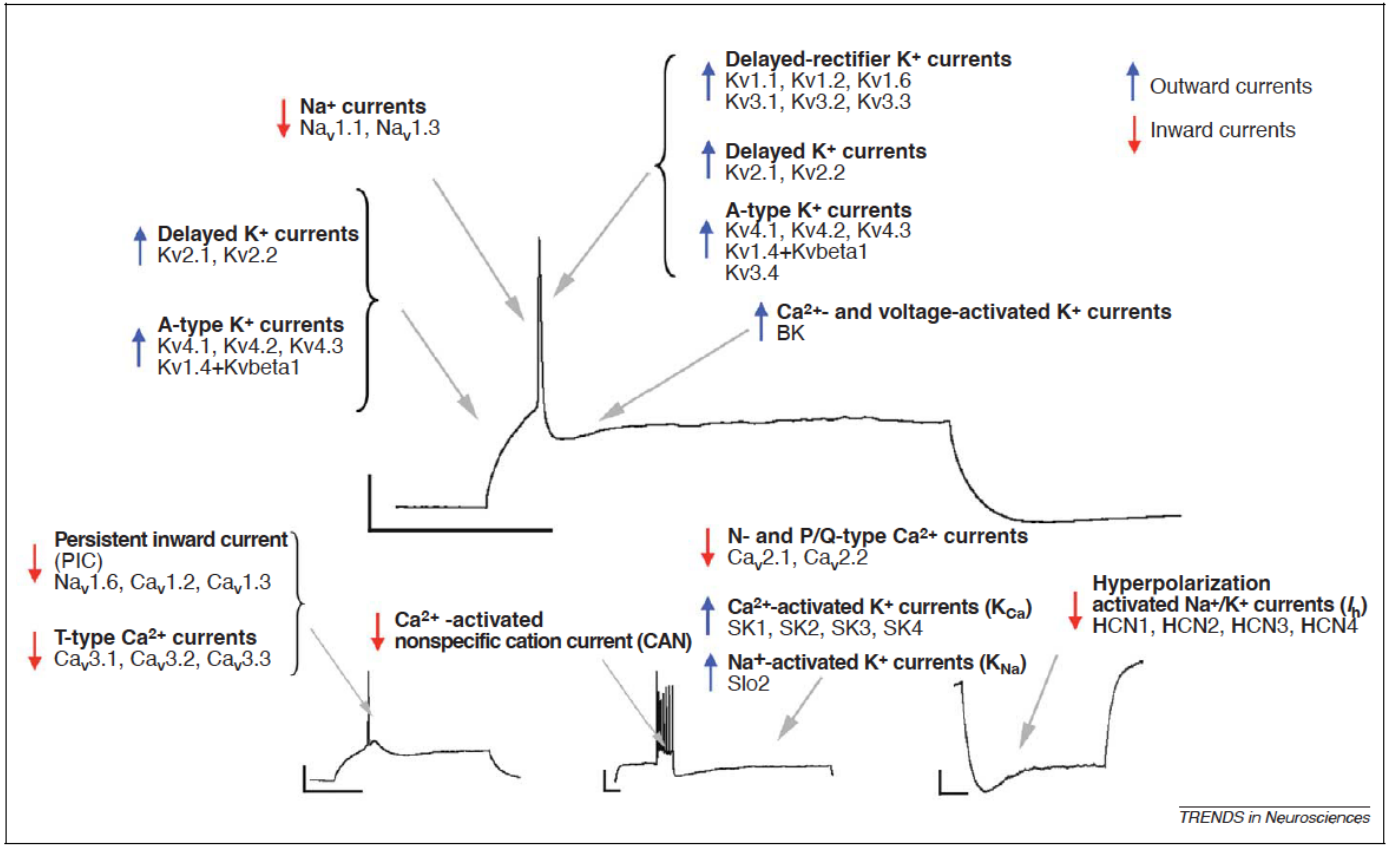
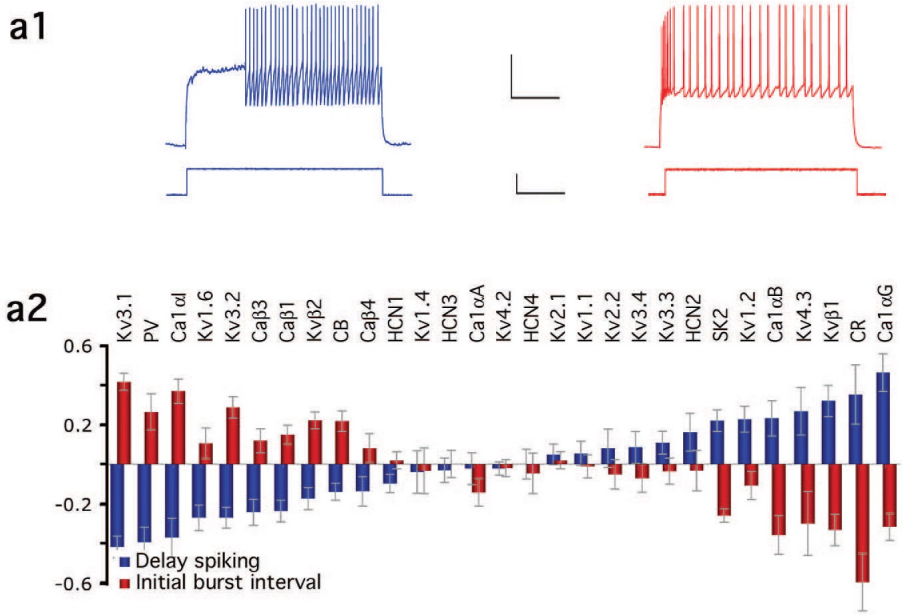


Figure 1. Different inward and outward currents and the ion channels that underlie each current [14–20,22–27,69,70,28–32]. Scale bars, 20 mV and 200 ms.

Toledo-Rodriguez, 2005

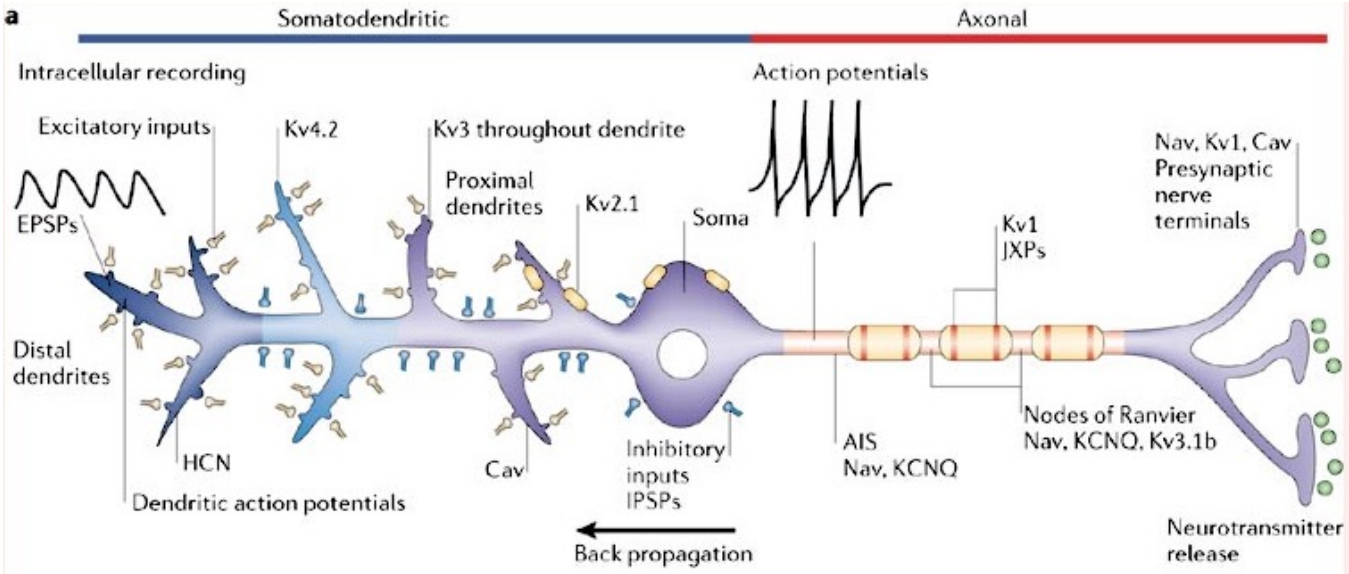
Different neurons have different ion channels

- Channel composition can be studied with electrophysiology (pharmacology), anti-body staining, single cell multiplex RT-PCR, single cell transcriptome.
- In the image, relative correlation of the different ion channel and calcium binding protein genes with electrical phenotypes (regression coefficient).



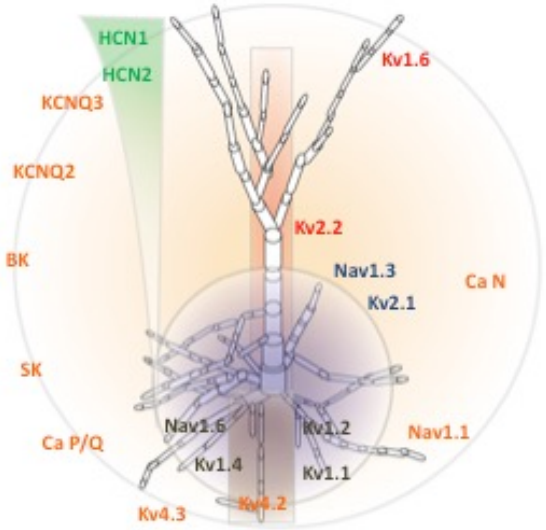
Toledo et al., 2004

Different compartments have different ion channels

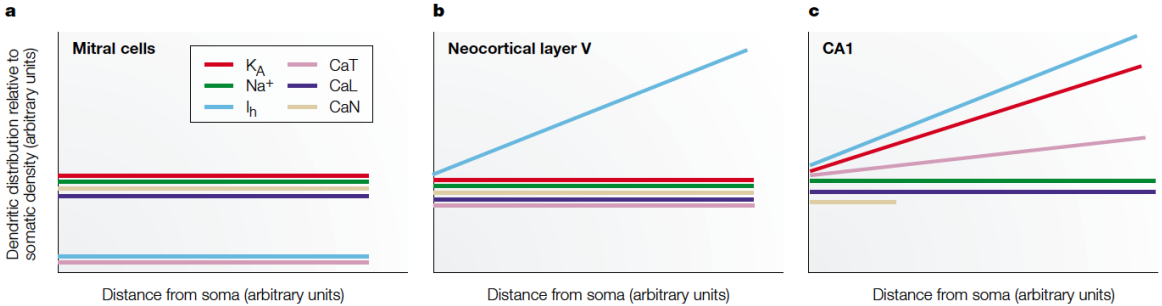


Lay and Jan, 2006

Ion channels show different distributions



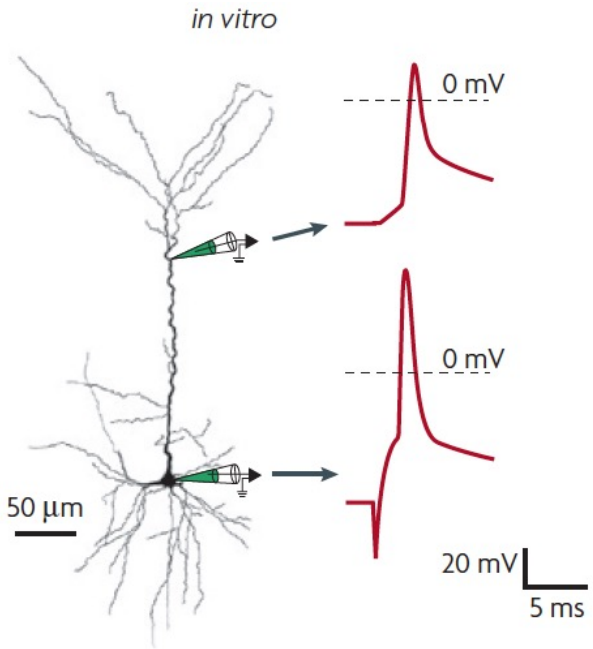
Rajnish, PhD thesis



Migliore and Shepherd, 2002

Backpropagating action potential (BPAP)

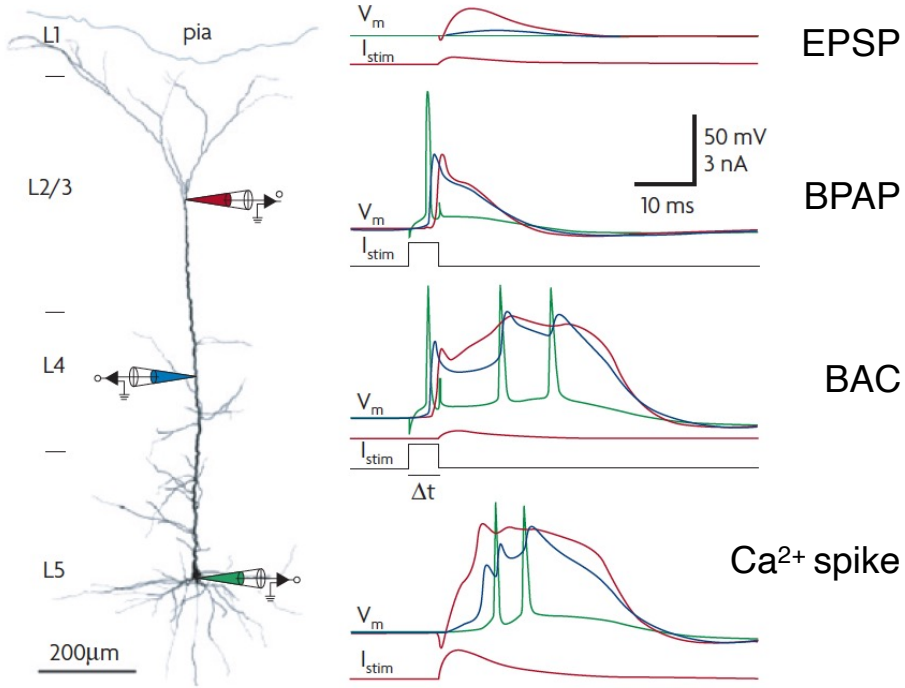
- An action potential (generated in soma or AIS) propagates backwards in the dendrites
- Ion channel composition affects how reliable the BPAP is transmitted
- BPAP can release the Mg^{2+}-block necessary to induce an NMDAR-mediated plasticity



Spruston, 2008

Dendritic spikes

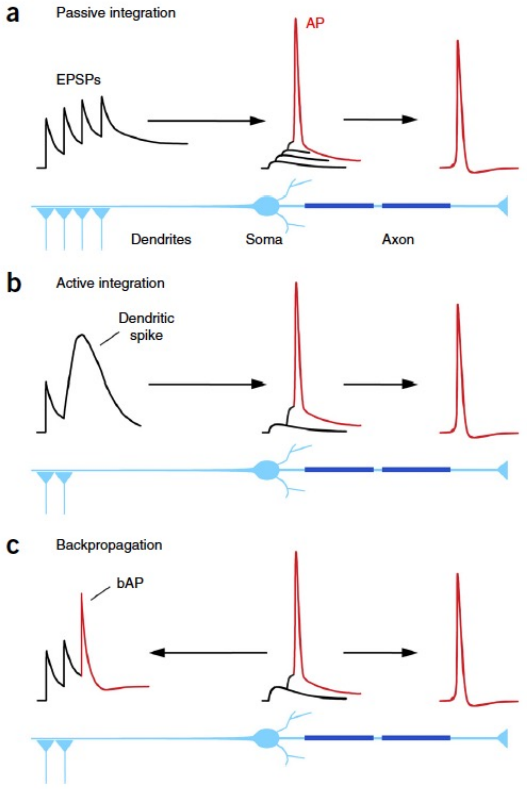
- Na^{+} spikes. Brief events
- Ca^{2+} spikes. Larger and broader events
- NMDA spikes. Due to release of Mg^{2+}. They remain where glutamate release occurs
- Backpropagation-activated Ca^{2+} spike (BAC spike). Synaptic stimulation + BPAP (figure)
- Dendritically initiated spikes are required for LTP or LTD induction in response to strong synaptic stimulation or during pairing of EPSPs with postsynaptic bursts



Larkum et al., 1999

Synaptic inputs, dendritic spikes and AP interact

- Dendrites are not very excitable and electrical events in the dendrites remains local
- However, depolarization of the dendrite can travel to the soma with more or less attenuation depending on the morphology and the ion channel composition
- Synaptic inputs, dendritic spikes and BPAP (or bAP) can interact if they are in the right space and time
- This interaction influences AP generation, plasticity, sensory tuning, feature selection…
- Few events can perform simple operation (see next). However, considering the complexity of the dendritic arbor we can imagine a sophisticated dendritic computation



Stuart and Spruston, 2015

Dendritic computation: AND, OR, AND-NOT operation

AND

- Synaptic inputs occurs in a dendrite in short sequence and generate a dendritic spike (coincidence detector)
- Proximal input enhances distal dendritic spike
- BPAP interacts with distal input and generates dendritic spike
- Several dendritic spikes trigger an AP

OR

- Two sets of inputs in two different branches can generate AP alone

AND-NOT

- On-path inhibition can block distal excitatory input
- Off-path inhibition can reduce dendritic spike

EPFL

Challenges

- What to model: individual cells vs classes
  - There seem to be classes of prototypical electrical behavior, but all cells are also unique (due their unique morphology and ion channels)
  - No final consensus on what classes exist (and on which modalities to establish the classes, i.e. morphology, electrical, genetic)
- Multimodal datasets/pooling of data
  - For detailed models, we need multiple modalities (e.g. morphology, ephys, transcriptome, others) but such data sets are rare
  - The alternative is to pool data across experiments & labs, which has the challenge of standardized protocols so that data can be pooled
- Difficult to obtain data: some parameters are much more inaccessible than others (e.g. channel localization and maximum conductances)

EPFL

Summary 1

- Building “realistic” models of specific neurons requires many parameters (morphology, ion channels, ion channel kinetics, ion channel distributions)
- One way to make models meaningful and achievable is to derive as many parameters as possible from experimental data
- Multi-compartment Hodgkin-Huxley is a well suited formalism for the body of data that is available today
- A data-driven model is a “data-ready” model, i.e. it is easy to absorb new data as it comes about
- Some parameters are much more inaccessible than others (e.g. channel localization and maximum conductances) and will need to be constrained differently

EPFL

Lecture Overview

- Scope
- **Approaches**
- Applications

Underconstrained parameters

- We normally have the characterization of the cell in terms of electrical behavior
- In most of the cases, we have *in vitro* somatic recordings of the cell responding to a limited number of artificial stimuli
- The composition and distribution of ion channels is generally unknown. We can make assumptions based on similar cells described in literature
- The number of unknown parameters normally exceeds the known ones

What to do with underconstrained parameters?

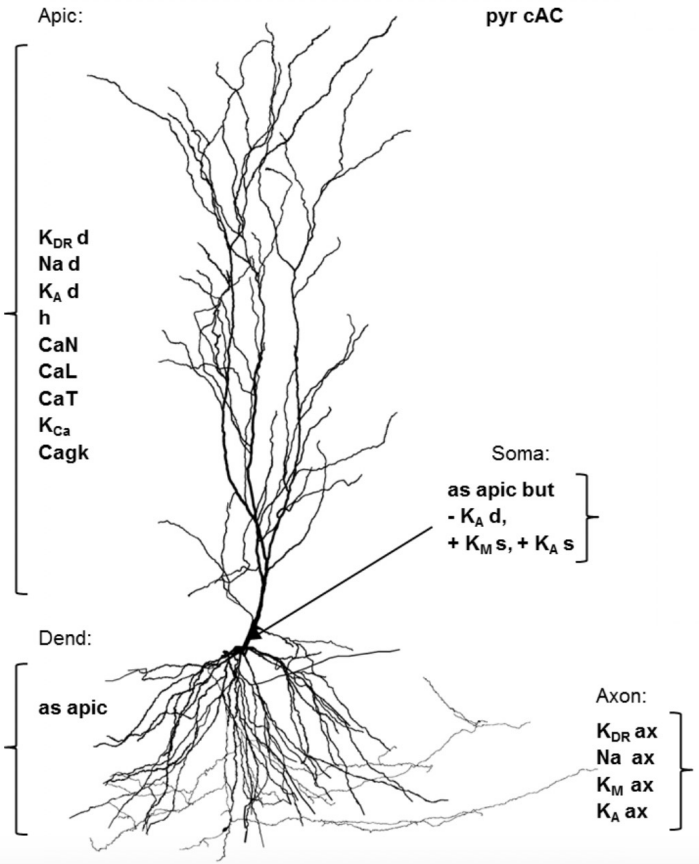
- More (novel) experiments
- Guess parameters and hand-tune
- Systematic grid search
- Regularize parameters
- Infer parameters from other (related) and more easily measurable properties
- Parameter Optimization

A typical example

- The morphology is given
- Ion channel models are given (already constrained)
- Different ion channels are present in different compartments
- For simplicity, assume that they are uniformly distributed within the compartment
- Ion channel models follow HH formalism

$$I_x = g_x(V_m - E_x)$$

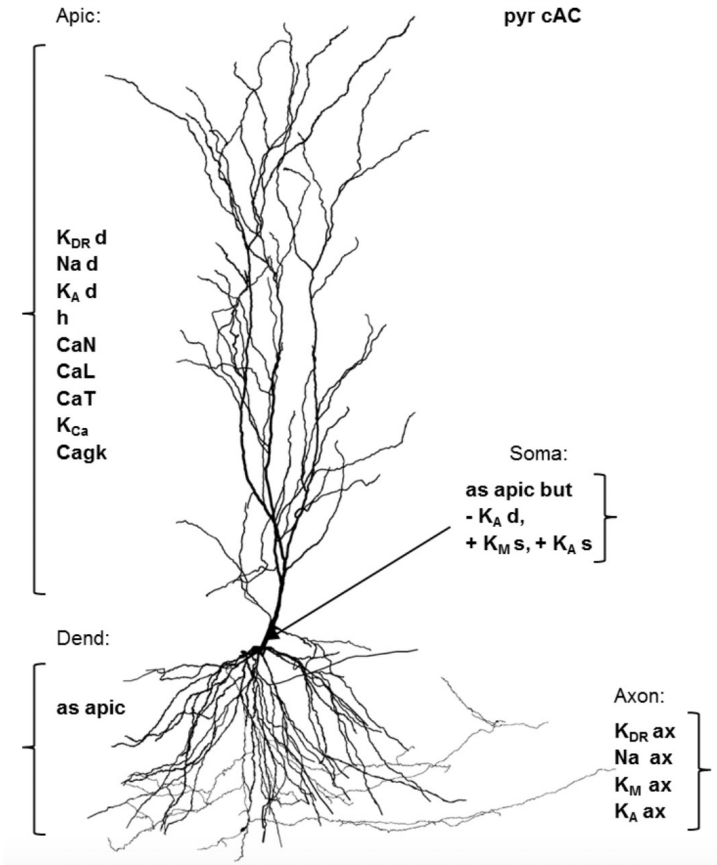
$$g_x = \bar{g}_x m^a h^b$$



Migliore et al., 2018

A typical example

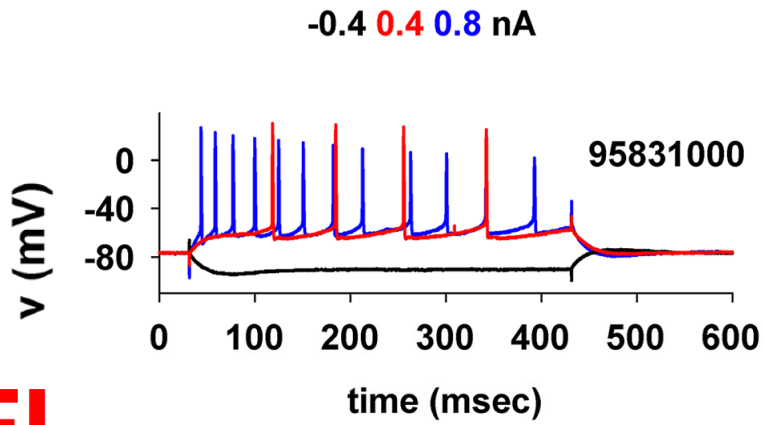
- $\bar{g}_x$ is the maximum conductance and represents the channel density
- Unknown parameters to be optimized include: $\bar{g}_x$ for the different currents in different compartments, passive properties (R_m, g_{leak}, E_{leak})
- Other parameters are known (E_x, C_m, R_a)

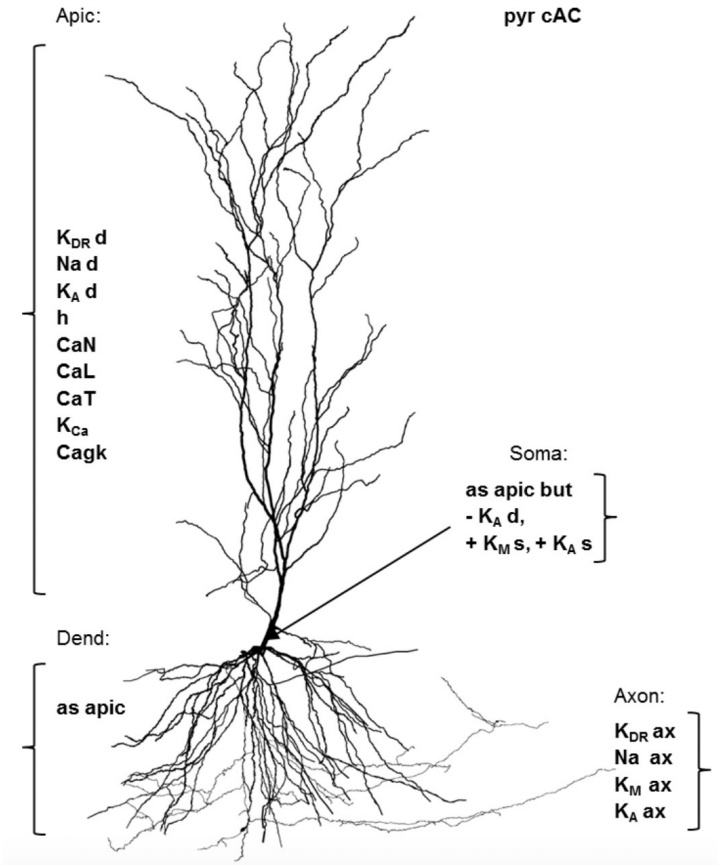


Migliore et al., 2018

A typical example

Define the unknown parameters ($\bar{g}_x$, R_m, g_{leak}, E_{leak}) so the neuron model reproduces the electrical behavior recorded experimentally (from the same cell of the reconstruction or from another cell). Let's consider only 1 trace per input





Migliore et al., 2018

A typical example

We can keep fixed the known parameters $\overrightarrow{P_f}$ vary the unknown parameters $\overrightarrow{P_v}$

We can express our model as $M(\overrightarrow{P_f}, \overrightarrow{P_v}, t, I_{inj})$ and the target experimental traces as $T(t, I_{inj})$

If we assign values to the $\overrightarrow{P_v}$, all the parameters can be represented as $\vec{P}$. The model can be simulated under the same experimental conditions and can produce model traces.

We can define a cost function or error as following:

$$err = |model - experiment|$$

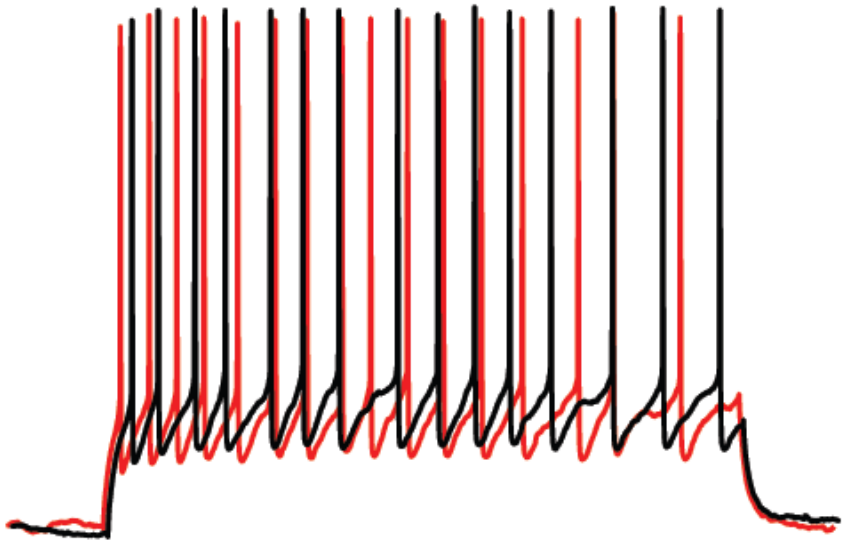
Aim of the optimization is to find a set of parameters $\vec{P}$ for which the error is sufficiently small.

EPFL

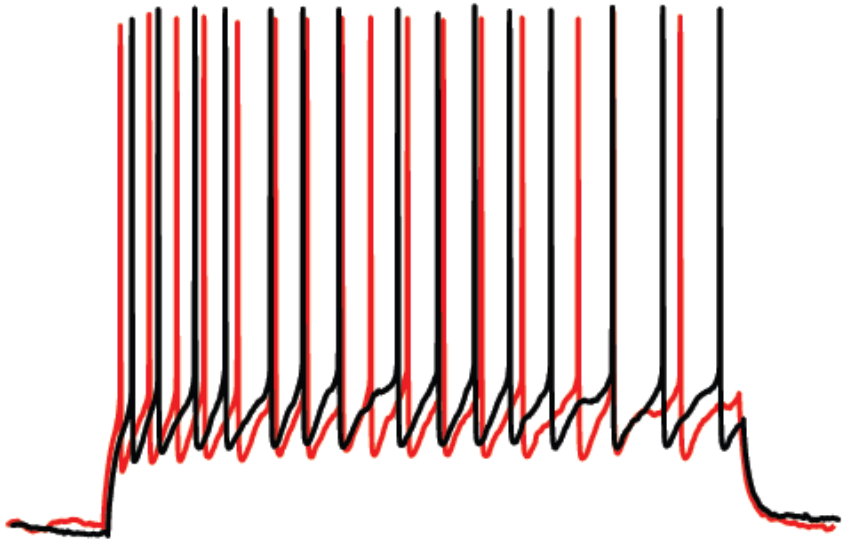
Cost functions

There are different ways in which we can define the cost function. The simplest way is to compare model and experiment traces point by point.

$$\text{rms} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}\left(V_{\text{data}}[i] - V_{\text{model}}[i]\right)^2}$$

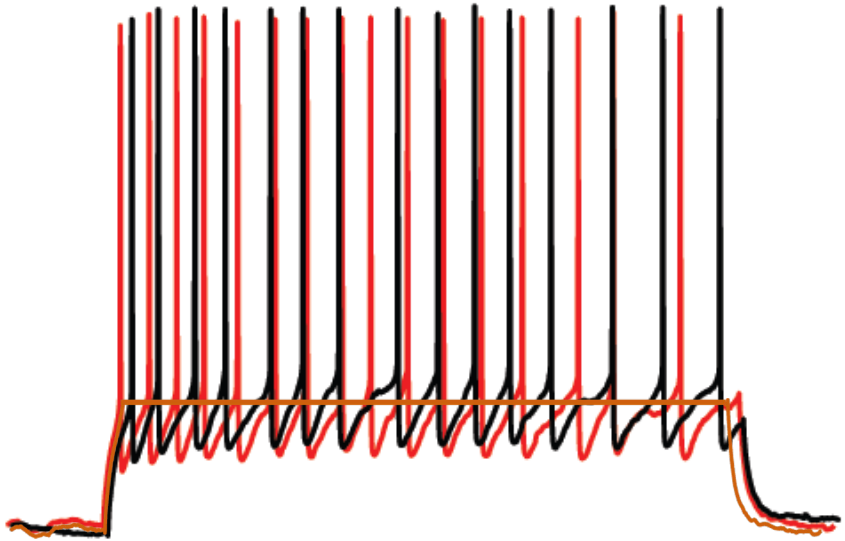


When Is a Model a Good Model?



→ actually those are two repetitions of the same cell!!

Choose the Metric Carefully!



Error: 0.3 (A.U.)

Error: 0.3 (A.U.)

Cost functions

However, not all the points have the same relevance for the behavior of the neuron. We could extract *features* (see figure) from model and experiment traces and compare them.

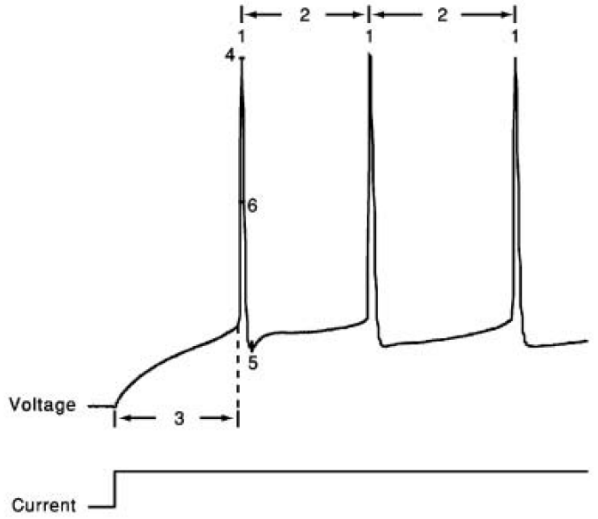


Figure 2. Feature extraction. *Voltage response (top) to a step depolarizing current (bottom) of the first 200 ms following stimulus onset of the trace displayed in Figure 1A. Extraction of the six features is schematically portrayed.* **1**, *spike rate;* **2**, *accommodation index (Equation 3);* **3**, *latency to first spike;* **4**, *AP overshoot;* **5**, *After hyperpolarization depth;* **6**, *AP width. For values of the different features in the case of the two electrical classes depicted in Figure 1, see Table 1.*

Drukmann et al., 2007

Cost functions

A feature-based cost function can be expressed as:

$$error = \sum \left| F_{mod,i} - F_{exp,i} \right|$$

Each term i is called *objective* or *objective function* and the optimization is called *multi-objective.*

Remember that we are comparing the model with one trace or one set of traces (one for each stimulus). The fact that we may have multiple stimuli and one trace per stimulus does not change much. In fact, different inputs give different features (AP height for 0.2 nA, AP height for 0.4 nA…).

Cost functions

If we have multiple experimental traces for each stimulus, so we have multiple measures for each feature, then for each feature we have a mean and a standard deviation (or another measure of the variability).

We can observe that some features are highly variable, while others are quite fixed.

A simple distance between model and experiment feature does not consider the variability of the features. We can weight the differences by the standard deviation of the feature measured in the experiment.

$$error = \sum \frac{\left|F_{mod,i} - E(F_{exp,i})\right|}{SD(F_{exp,i})}$$

EPFL

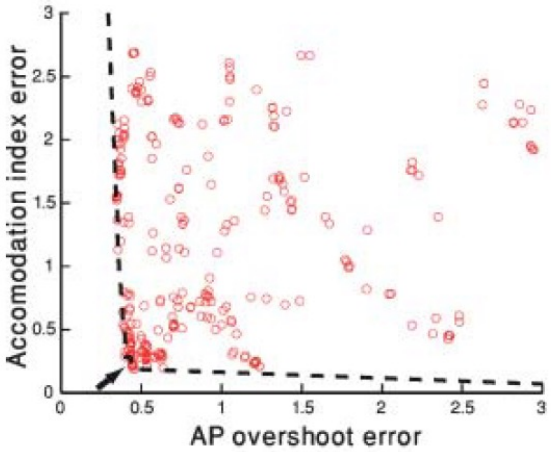
Pareto front

In multi-objective optimization, we say that one solution dominates another if it does better than the other solution in at least one objective and not worse than the other solution in all other objectives. If there are M objective functions $f_j(x)$, j=1 . . . M, then a solution x^1 is said to dominate a solution x^2 if both the following conditions hold:

$$f_j\left(x^1\right) \leq f_j\left(x^2\right) \text{ for all } j = 1 \ldots M$$

$$f_k\left(x^1\right) < f_k\left(x^2\right) \text{ for at least one } k \in \{1, 2\ , \ldots M\}$$

The optimization produces a set of solutions which do not dominate each other called pareto front (see image).



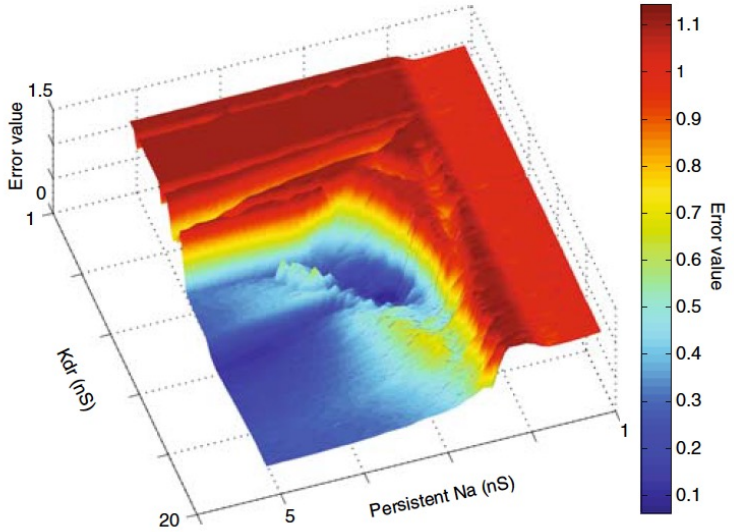
Drukmann et al., 2007

Search strategies

To find the *best* solution (global minimum) or at least an acceptable solution (local minimum), we have to test different parameter sets and evaluate the resulting model.

The simplest approach is the parameter scan that is possible only when the number of parameters and the parameter space is limited.

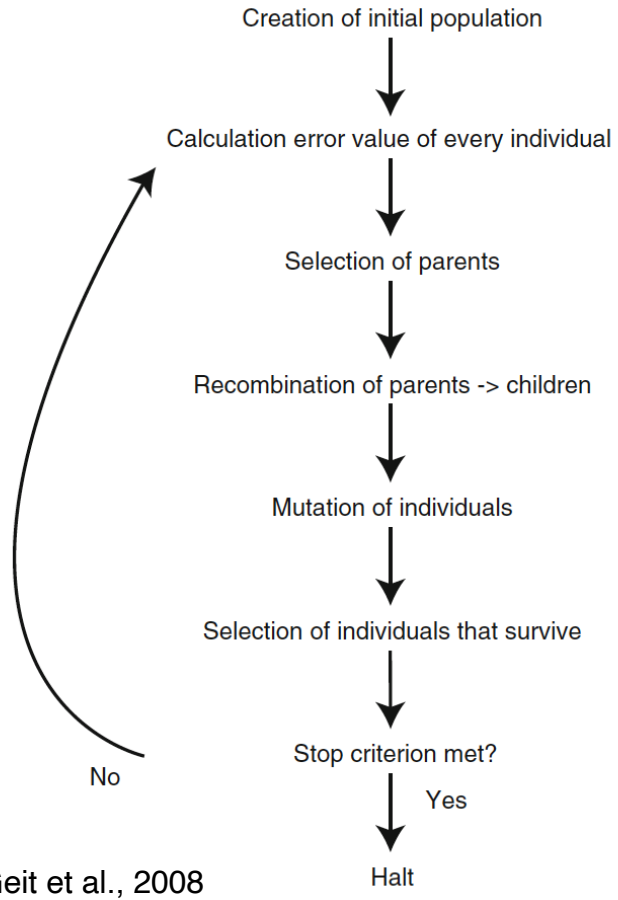
When this is not the case, the parameter space cannot be fully explore and we need strategies to sample the space in a faster and *smarter* way.



Van Geit et al., 2008

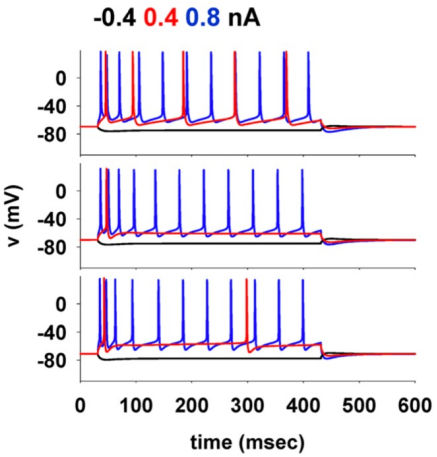
Evolutionary algorithm

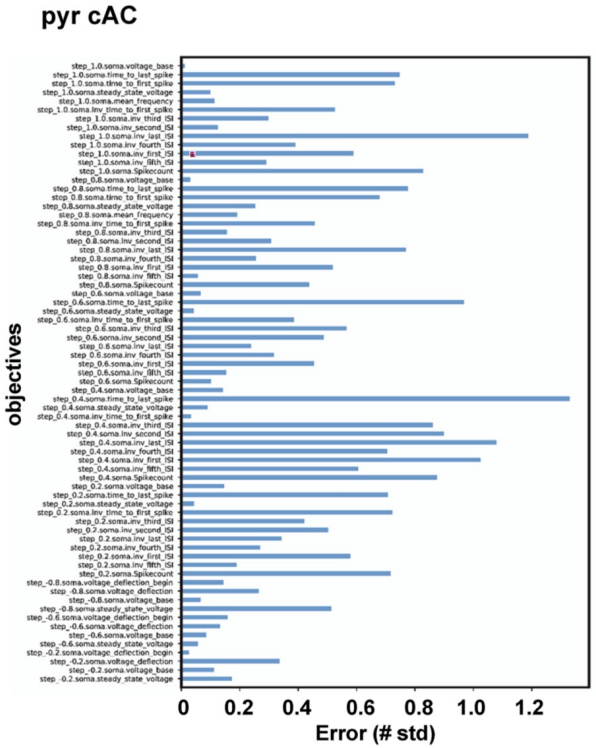
- Mutation. The value is changed within its boundary by a certain amount (generally, both defined by the user)
- Stop criteria. You can stop with the maximum number of generations
- In real cases, from the pareto front, it is possible to select an individual that perform slightly better than the others



Van Geit et al., 2008

Evolutionary algorithm





Migliore et al., 2018

Summary 2

- For model parameters that cannot be obtained from experiment, parameter optimization can be viable approach
- Metaheuristics such as evolutionary algorithms combined with smart metrics and multi-objective optimization have proved useful for finding conductance values and distributions for multi-compartment neuron models
- They can find parameter sets that produce a certain behavior and reveal trade-offs in the solution space
- Solutions are not guaranteed to be unique and generalization has to be tested

EPFL

Lecture Overview

- Scope
- Approaches
- **Applications**

Hay et al., 2011

OPEN ACCESS Freely available online

PLoS COMPUTATIONAL BIOLOGY

Models of Neocortical Layer 5b Pyramidal Cells Capturing a Wide Range of Dendritic and Perisomatic Active Properties

Etay Hay[1]*, Sean Hill[2], Felix Schürmann[2], Henry Markram[2], Idan Segev[1,3]

1 Interdisciplinary Center for Neural Computation and Edmond and Lily Safra Center for Brain Sciences, The Hebrew University, Jerusalem, Israel, **2** Brain Mind Institute, Ecole Polytechnique Fèdèrale de Lausanne (EPFL), Lausanne, Switzerland, **3** Department of Neurobiology, The Hebrew University, Jerusalem, Israel

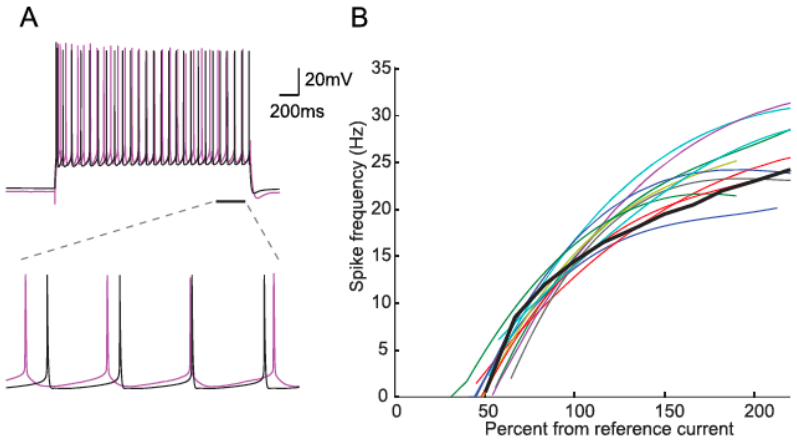
Abstract

The thick-tufted layer 5b pyramidal cell extends its dendritic tree to all six layers of the mammalian neocortex and serves as a major building block for the cortical column. L5b pyramidal cells have been the subject of extensive experimental and modeling studies, yet conductance-based models of these cells that faithfully reproduce both their perisomatic Na^+-spiking behavior as well as key dendritic active properties, including Ca^{2+} spikes and back-propagating action potentials, are still lacking. Based on a large body of experimental recordings from both the soma and dendrites of L5b pyramidal cells in adult rats, we characterized key features of the somatic and dendritic firing and quantified their statistics. We used these features to constrain the density of a set of ion channels over the soma and dendritic surface via multi-objective optimization with an evolutionary algorithm, thus generating a set of detailed conductance-based models that faithfully replicate the back-propagating action potential activated Ca^{2+} spike firing and the perisomatic firing response to current steps, as well as the experimental variability of the properties. Furthermore, we show a useful way to analyze model parameters with our sets of models, which enabled us to identify some of the mechanisms responsible for the dynamic properties of L5b pyramidal cells as well as mechanisms that are sensitive to morphological changes. This automated framework can be used to develop a database of faithful models for other neuron types. The models we present provide several experimentally-testable predictions and can serve as a powerful tool for theoretical investigations of the contribution of single-cell dynamics to network activity and its computational capabilities.

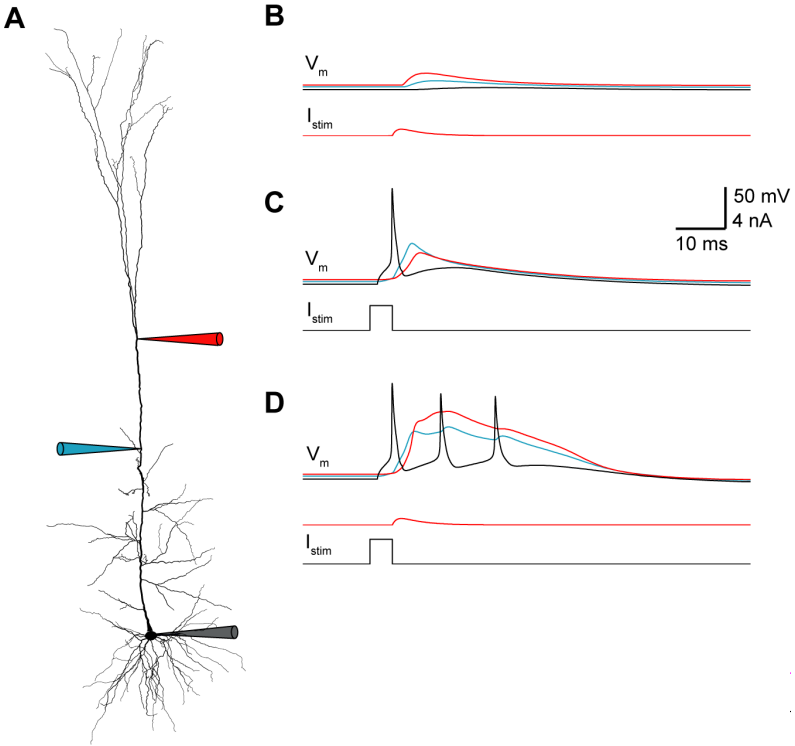
July 2011 | Volume 7 | Issue 7 | e1002107

Hay et al., 2011

Somatic behavior



Dendritic behavior



Target Features

Table 1. Mean and SD values of features of perisomatic step current firing and of BAC firing.

Features of perisomatic step current firing				Features of BAC firing	
Feature	**Mean±SD, Low frequency**	**Mean±SD, Reference frequency (15 Hz)**	**Mean±SD, High frequency**	**Feature**	**Mean±SD**
1. Spike frequency (Hz)	9±0.88	14.5±0.56	22.5±2.22	1. Ca^{2+} spike peak (mV)	6.73±2.54
2. Adaptation Index	0.0036±0.0091	0.0023±0.0056	0.0046±0.0026	2. Ca^{2+} spike width (ms)	37.43±1.27
3. ISI-CV	0.1204±0.0321	0.1083±0.0368	0.0954±0.0140	3. Somatic AP spike count (during somatic + dendrite current injection)	3±0
4. Initial Burst ISI (ms)	57.75±33.48	6.625±8.65	5.38±0.83	4. Mean somatic AP ISI (ms)	9.9±0.85
5. First spike latency (ms)	43.25±7.32	19.13±7.31	7.25±1	5. Somatic AHP depth (mV)	−65±4
6. AP peak (mV)	26.23±4.97	16.52±6.11	16.44±6.93	6. Somatic AP peak (mV)	25±5
7. Fast AHP depth (mV)	−51.95±5.82	−54.19±5.57	−56.56±3.58	7. Somatic AP half-width (ms)	2±0.5
8. Slow AHP depth (mV)	−58.04±4.58	−60.51±4.67	−59.99±3.92	8. Somatic AP spike count (during somatic current injection only)	1±0
9. Slow AHP time	0.238±0.030	0.279±0.027	0.213±0.037	9. BAP amplitude at 620 μm (mV)	45±10
10. AP half-width (ms)	1.31±0.17	1.38±0.28	1.86±0.41	10. BAP amplitude at 800 μm (mV)	36±9.33

EPFL

Channels used in the Model

"We included ten key active ionic currents known to play a role in L5 PCs or generally in neocortical neurons, with kinetics taken strictly from the experimental literature. Kinetics of ion conductances that were characterized in room temperature (21°C) were adjusted to the simulation temperature of 34°C using Q10 of 2.3…"

Fast inactivating Na+ current, I_{nat}
Persistent Na+ current, I_{Nap}

Non-specific cation current, I_h
Muscarinic K+ current, I_m
Slow inactivating K current, I_{Kp}
Fast inactivating K current, I_{Kt}

Fast, non inactivating K+ current, $I_{Kv3.1}$
Intracellular [Ca2+] dynamics
High voltage activated Ca2+ current, I_{Ca_HVA}
Low voltage activated Ca2+ current, I_{Ca_LVA}
Small-conductance, Ca2+ activated K+ current, I_{SK}

EPFL

Fitting for Somatic AND Dendritic Behavior is difficult

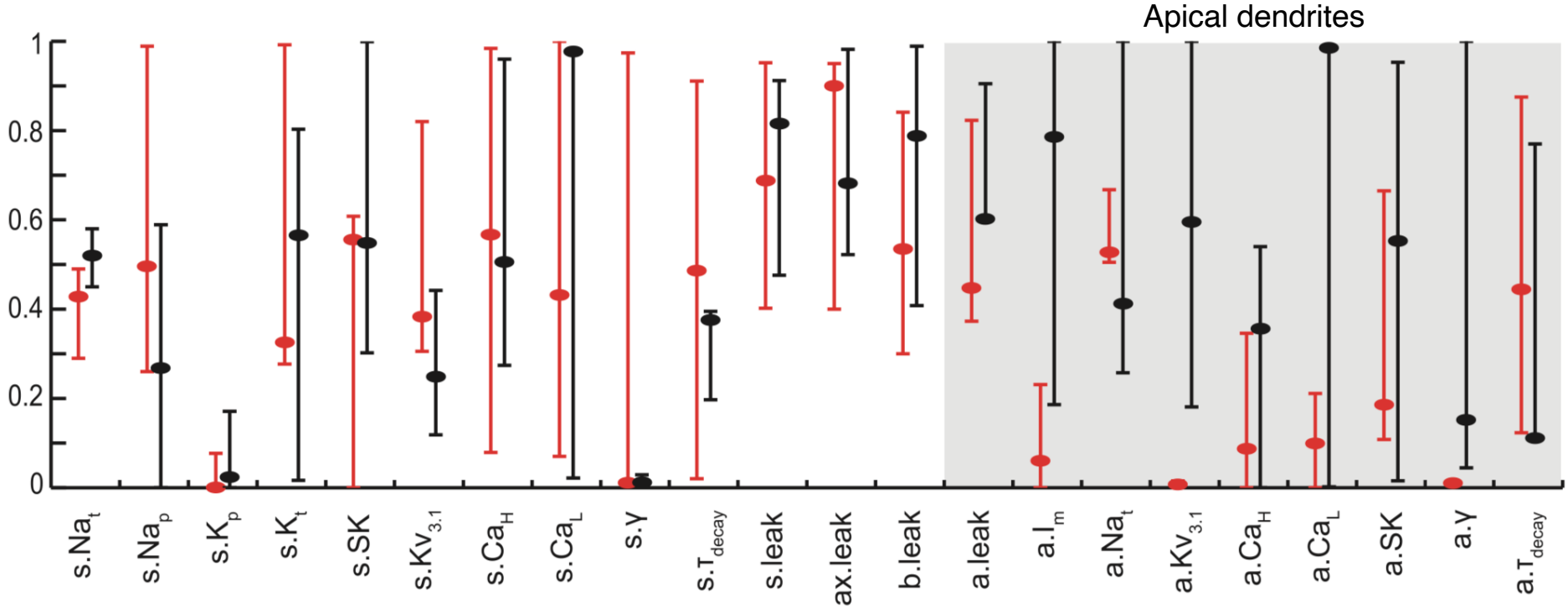


Figure 3. Parameter ranges for acceptable models for either perisomatic step current firing or BAC firing. Distribution of normalized parameter values in models constrained by BAC firing (red, n = 899 acceptable models) or by perisomatic step current firing (black, n = 52 acceptable models). For ease of viewing, the graph region containing parameters at the apical tree is shaded in gray. Red and Black circles correspond to specific normalized parameter values of the models shown in Figures 1 and 2,

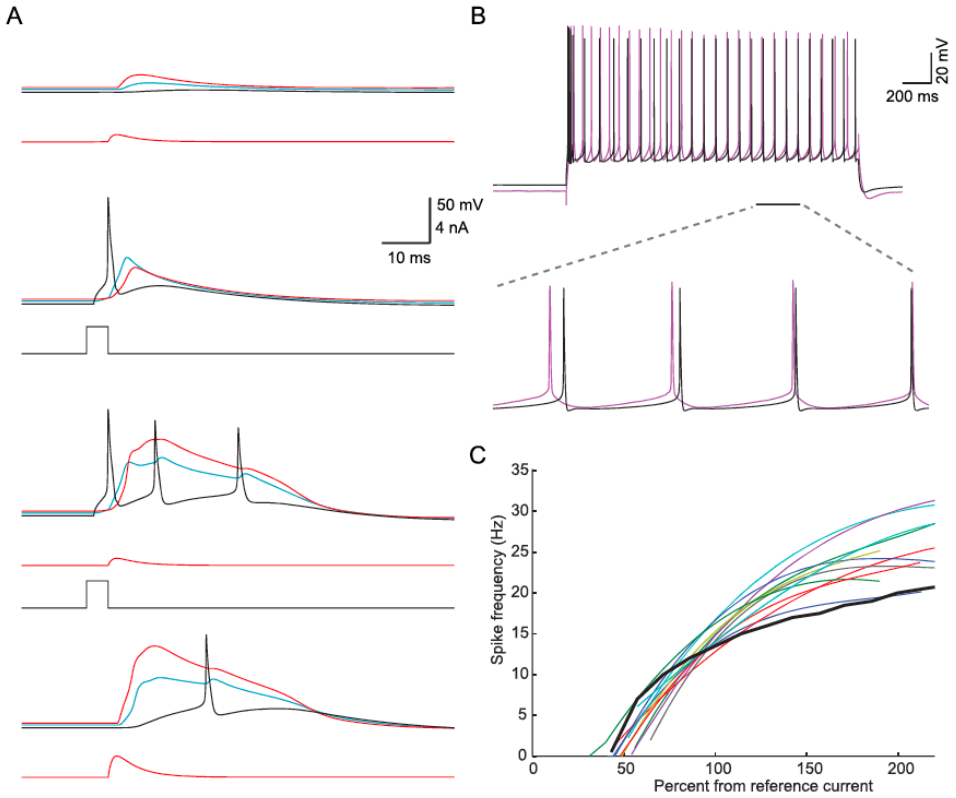
EPFL

Two Step Approach

- Fitting all 20 target features in one single optimization proved difficult (despite 2000 generations and population size of 5000), and models acceptable for somatic firing did not generalize well to BAC firing and vice versa
- However, BAC firing is mostly constrained by dendritic conductances, somatic firing by soma/AIS conductances
- Two-step approach:
  - First fit all conductances for optimal BAC firing behavior,
  - Use this to initialize a second optimization of somatic conductances for somatic firing (while keeping dendritic channels fixed)

EPFL

Results


A
50 mV
4 nA
10 ms
B
20 mV
200 ms
C
Spike frequency (Hz)
Percent from reference current
0
5
10
15
20
25
30
35
50
100
150
200

Take Home Messages from Hay et al. 2011

- Faithful model of a layer 5b pyramidal cell (rat), a principal cell of the neocortex, exhibiting a wide range of previously observed somatic and dendritic properties
- Demonstration that previously developed multi-objective optimization and feature-based fitness functions can be used to built such complex models, however, two step approach was needed
- The model has since become a reference model for this cell type (ie. used for many subsequent studies)
- Since the publishing of this paper, the adoption of a different class of optimization algorithm has made it possible to fit the model in one step → https://github.com/BlueBrain/BluePyOpt

EPFL

Migliore et al. 2018

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

The physiological variability of channel density in hippocampal CA1 pyramidal cells and interneurons explored using a unified data-driven modeling workflow

Rosanna Migliore[1]*, Carmen A. Lupascu[1], Luca L. Bologna[1], Armando Romani[2], Jean-Denis Courcol[2], Stefano Antonel[2], Werner A. H. Van Geit[2], Alex M. Thomson[3], Audrey Mercer[3], Sigrun Lange[3,4], Joanne Falck[3], Christian A. Rössert[2], Ying Shi[2], Olivier Hagens[5], Maurizio Pezzoli[5], Tamas F. Freund[6,7], Szabolcs Kali[6,7], Eilif B. Muller[2], Felix Schürmann[2], Henry Markram[2], Michele Migliore[1]

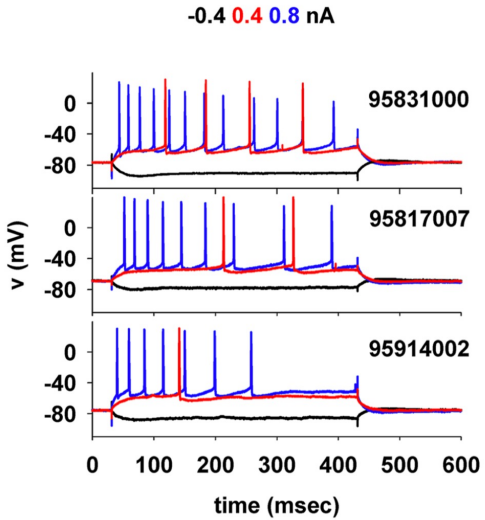
1 Institute of Biophysics, National Research Council, Palermo, Italy, **2** Blue Brain Project, École Polytechnique Fédérale de Lausanne, Campus Biotech, Geneva, Switzerland, **3** University College London, London, United Kingdom, **4** University of Westminster, London, United Kingdom, **5** Laboratory of Neural Microcircuitry (LNMC), Brain Mind Institute, EPFL, Lausanne, Switzerland, **6** Institute of Experimental Medicine, Hungarian Academy of Sciences, Budapest, Hungary, **7** Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Hungary

Check for updates

Experimental Data – Hippocampus CA1 Pyramidal Cells

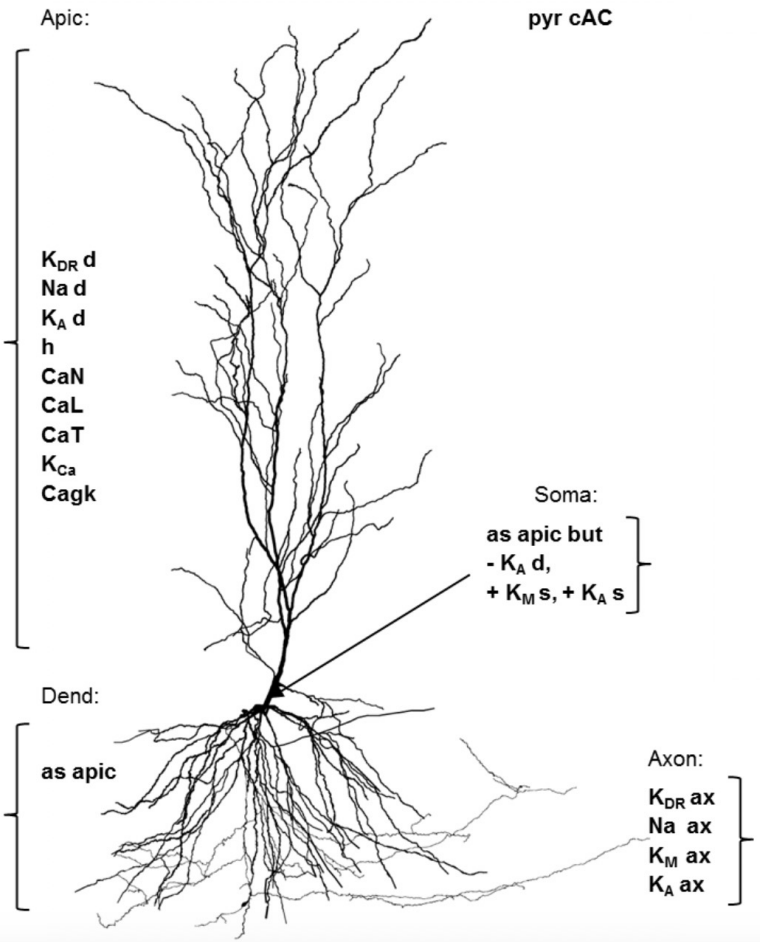
Morphology

Electrophysiology

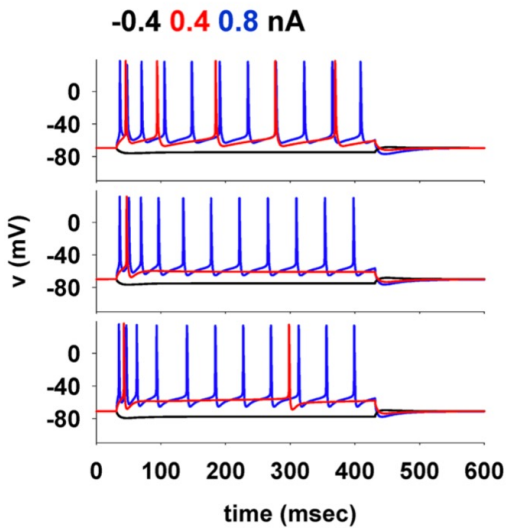


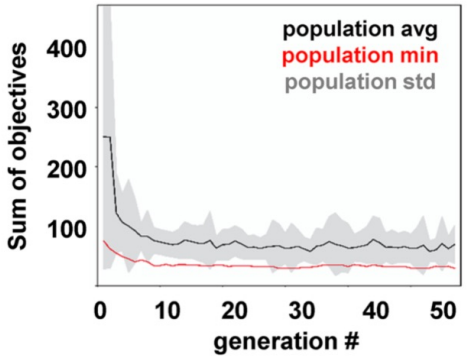
Feature / Input current	-0.4 nA	-0.2 nA	0.2 nA	0.4 nA	0.6 nA	0.8 nA
Voltage deflection	-19.77±2.03	-9.86±1.08				
Voltage base	-74.62±2.14	-74.96±2.87	-74.85±3.32	-74.86±3.06	-74.66±3.31	-74.62±3.49
Spikecount			0.83±1.18	7.92±4.82	15.67±7.52	23.17±8.56
Time to last spike			65.93±93.24	276.67±89.38	347.74±47.00	351.53±29.35
Inv time to first spike			4.80±6.78	72.80±51.42	157.46±71.90	251.37±64.96
Inv first ISI			4.30±6.09	38.72±25.55	75.99±40.95	122.74±31.38
Inv second ISI			1.81±2.56	30.74±18.08	72.92±29.61	113.52±29.38
Inv third ISI				30.74±21.88	67.76±28.65	109.43±29.03
Inv fourth ISI				21.40±16.11	61.18±26.89	100.68±30.11
Inv fifth ISI				21.21±16.13	53.56±35.23	99.49±29.37
Inv last ISI			3.64±5.14	16.60±10.71	37.22±2.62	59.24±5.78
Mean frequency					43.11±16.84	65.79±24.72
Time to first spike					14.19±12.80	5.54±2.77
AHP_depth					13.08±2.67	19.98±6.48

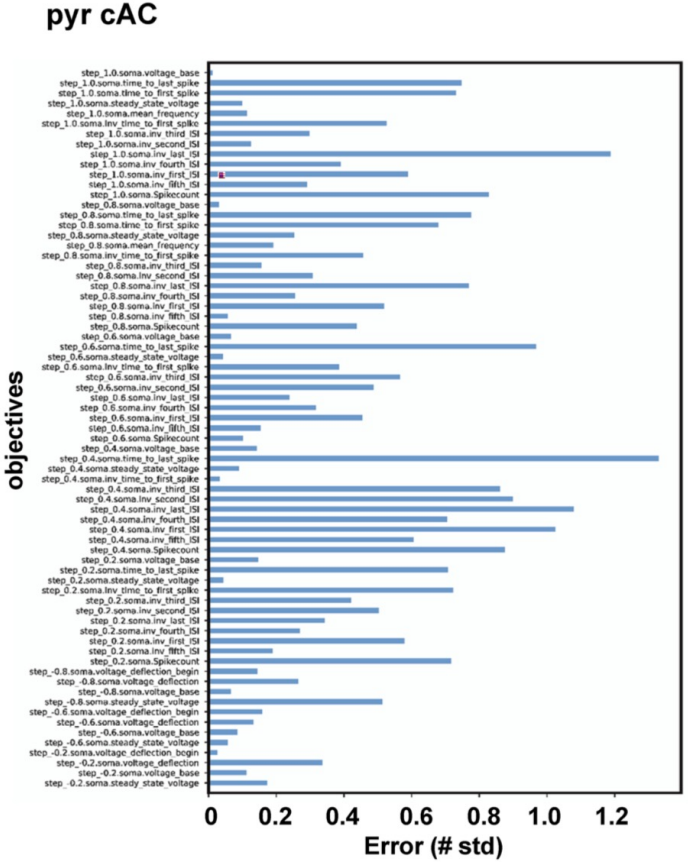
Model Assumptions (Pyr cAC)



Optimization Results (Pyr cAC)

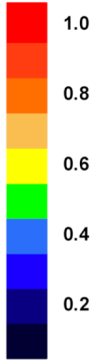




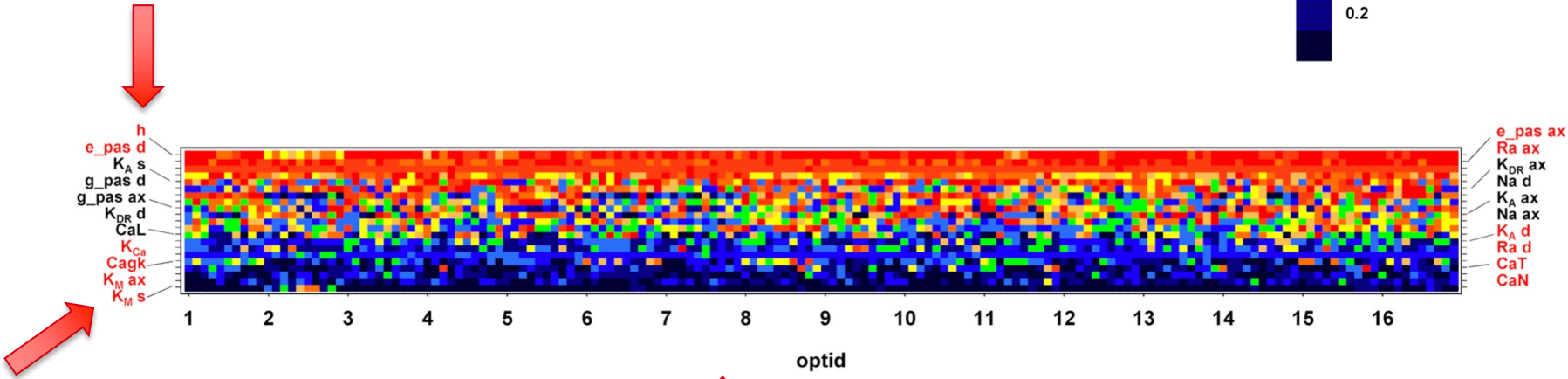


Degeneracy of Found Solutions (Pyr cAC)

The pixel colors represent the value of the parameter, normalized to the maximum value obtained from all optimizations



Red labels: parameters with <0.2 sd across all solutions



Sorted according to average value over all optimizations

Individual optimization yielding acceptable solutions

Take Home Messages from Migliore et al. 2018

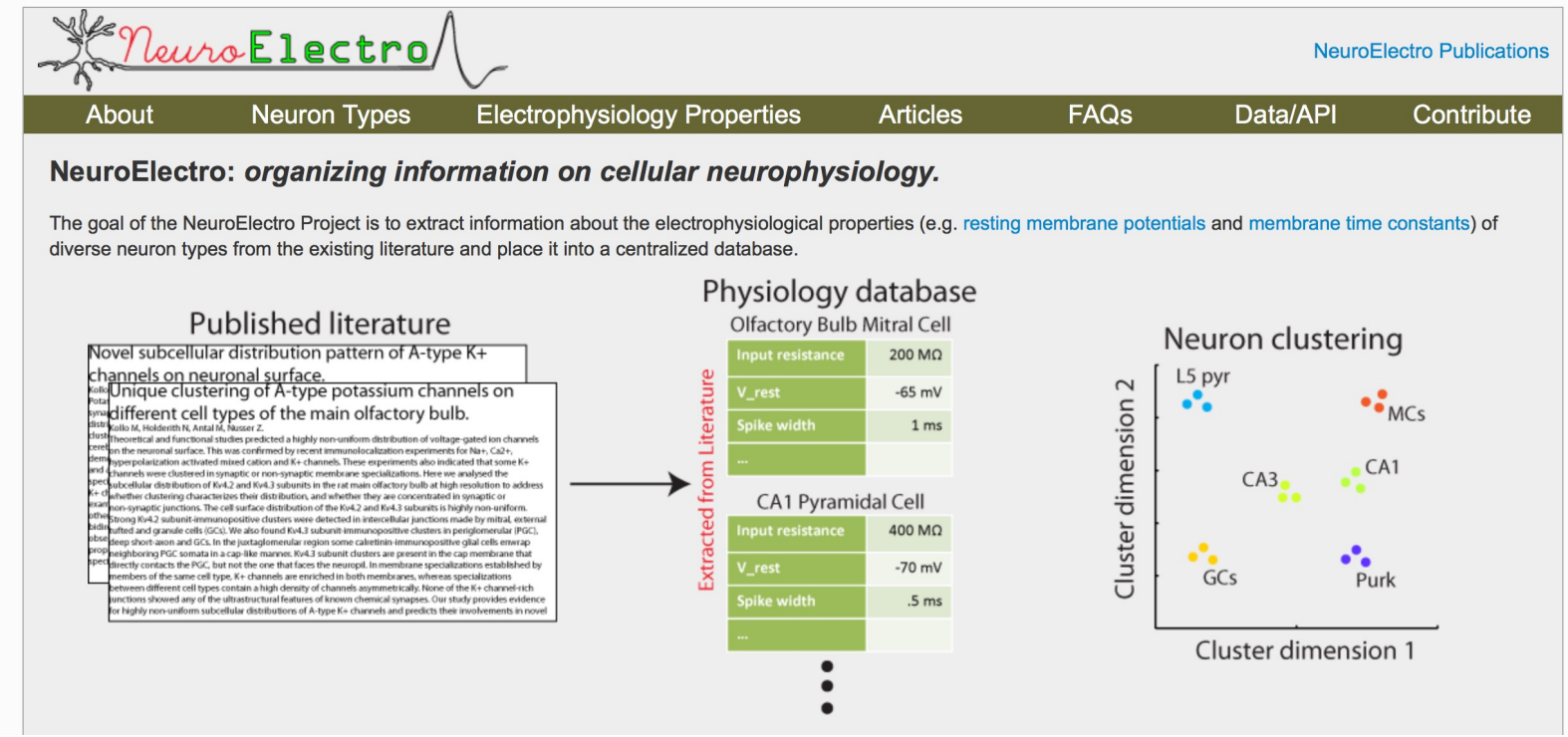
- Multi objective optimization and feature-based fitness used to constrain morphologically detailed models of different hippocampal neuron types
- Several models for each type passed the selection threshold and this is used to analyze what parameters are conserved or different (degeneracy)
- For the pyramidal cell class, the most stable (ie. for which the optimization did not identify degeneracy) parameters were: passive properties, I_h, K_M, Calcium, and Ca-dependent K currents
- This links with and provides some explanation for previous findings, where
  - K_M was previously known to be dominant factor for excitability and accommodation and specific mutations can lead to neonatal epilepsy
  - I_h was previously known for its importance for synaptic integration and that even small decreases have previously been show to have major effects mechanisms related to cognitive function

EPFL

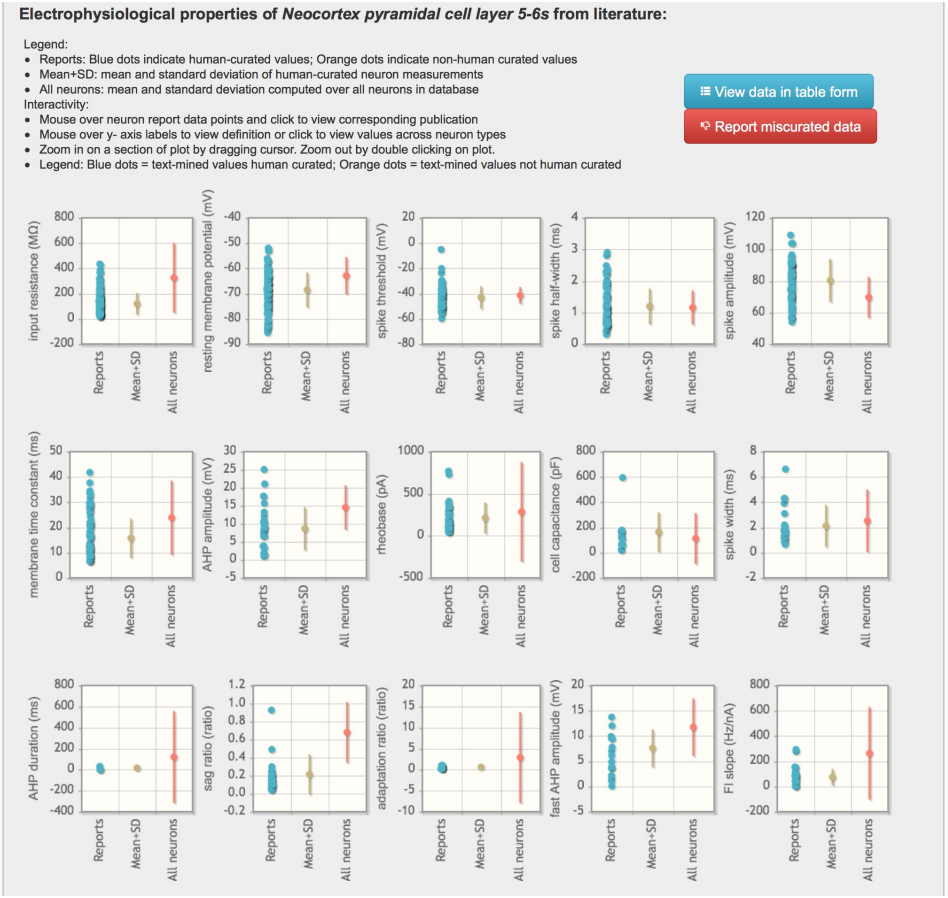
Summary 3

- Multi objective optimization and feature based fitness functions have proven to be effective tools to build some of the most faithful and complex models of neuron types to date (in diverse rodent brain regions such as cortex, hippocampus, cerebellum, basal ganglia and even for human cells)
- Not only are those methods useful to generate models, but the method's property of producing families of models (pareto-equivalent) can be used to gain insights on the degeneracy and criticality of certain conductances
- The methods are readily available (open source, as a service); will present in more detail in next lecture

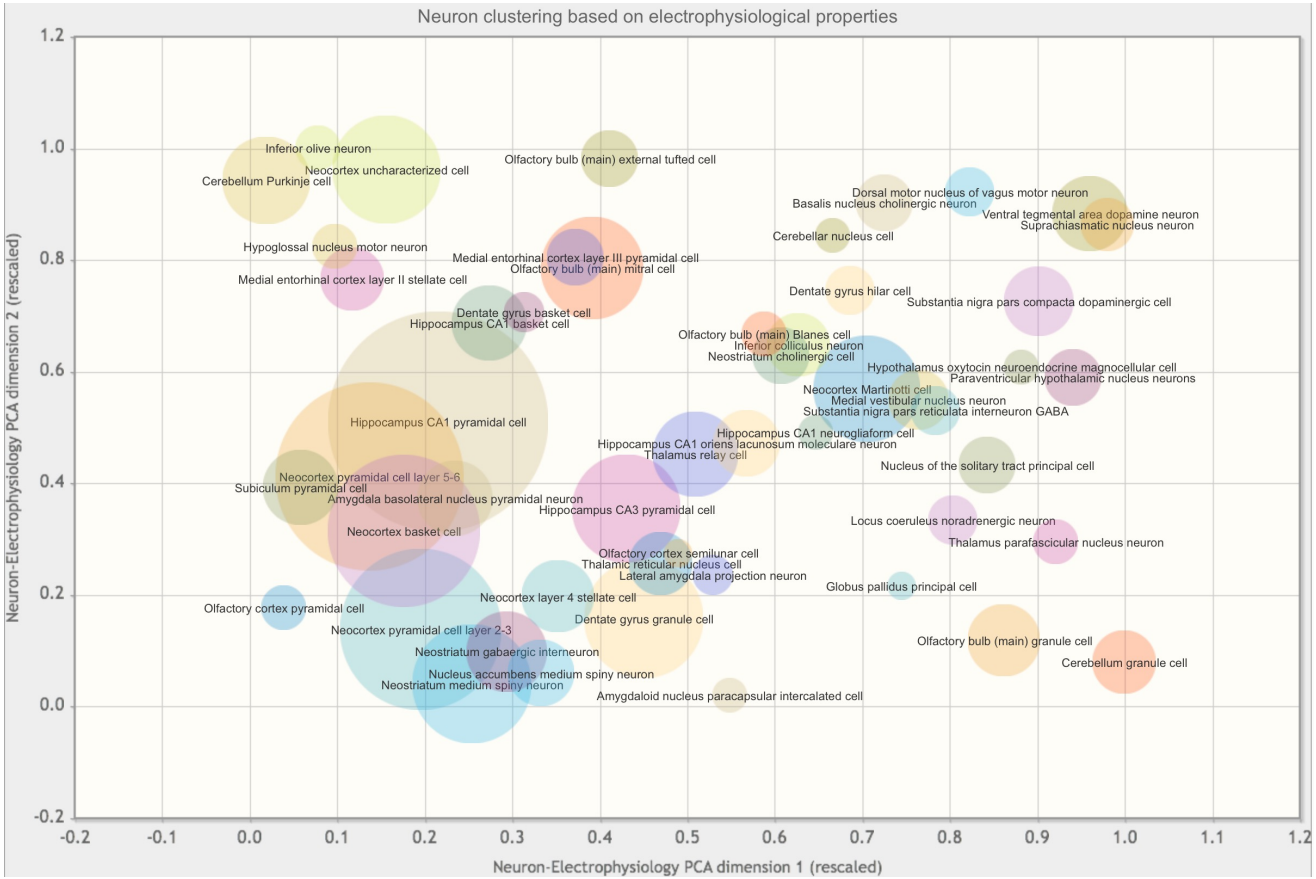
Web Resources 1 - Neuroelectro.org



Integrated Data from Literature (via neuroelectro.org)

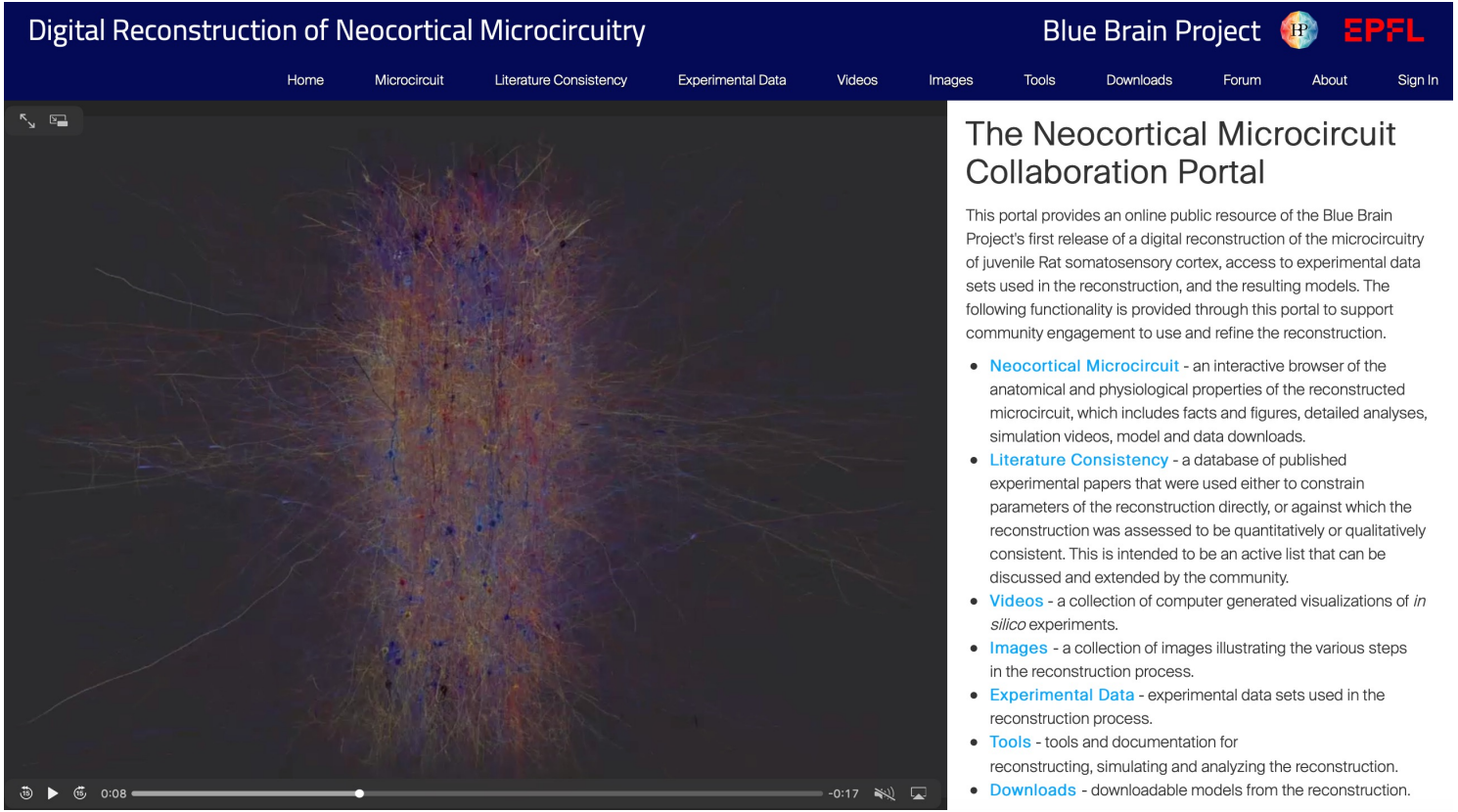


Cell types reported in the literature



from neuroelectro.org

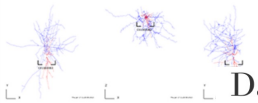
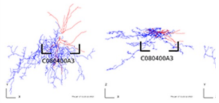
Web Resources 2 – Blue Brain Neocortical Microcircuit Portal



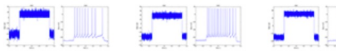
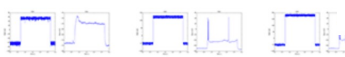
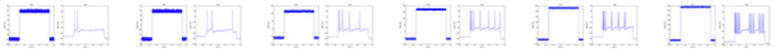
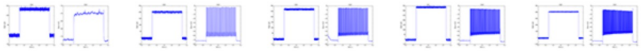
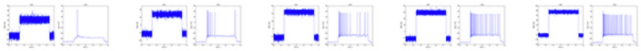
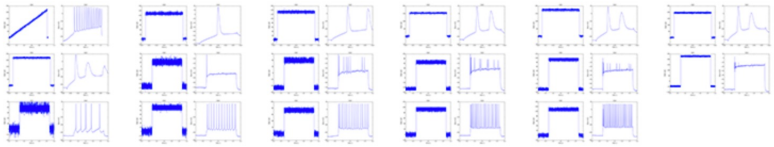
https://bbp.epfl.ch/nmc-portal/

Morphologies, Electrophysiology & Neuron Models

Data Download all files

animal	preview
animal C060600A2	
animal C240300C1	
animal C010600B2	
animal C080400A3	

Data Download all files

animal	preview
animal B95	
animal C9	
animal C12	
animal A53	
animal A61	
animal A13	

Neurons

Models of individual neurons (in the NEURON simulation environment) can be obtained from the me-type fact sheets. A model package contains a morphology file, the ion channel descriptions, the synapse descriptions and templates to instantiate the cell. There is code to construct the model, distribute the ion channels and synapses on the morphology, generate a GUI, and inject current clamp and synaptic input into the model.

The morphology description is a formatted as a Neurolucide ASCII file. The ion channel and synapse description are available as NEURON MOD files. The code for the templates, GUI and simulations is written in the NEURON HOC language or Python. The files containing the synaptic parameters and m-type mappings are tab-separated files.

The complete set of neuron models is available here

You can select specific neuron models in the interface below or by browsing to the relevant me-type fact sheet.

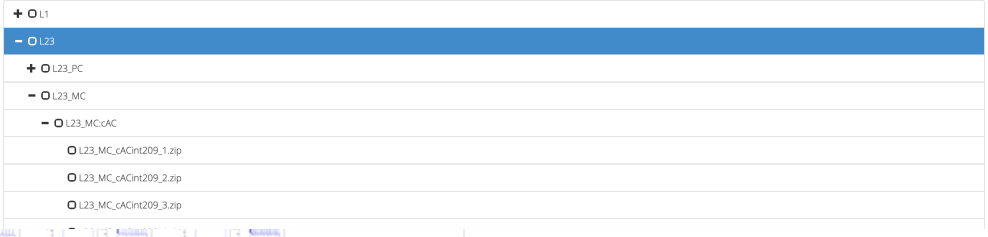
A JSON file with some extra data about each neuron model, like the resting membrane potential, input resistance and membrane time constant, can be downloaded here

NeuroML2 version

Thanks to a collaboration with Padraig Gleeson of the Open Source Brain, the model packages can be converted to NeuroML2:

https://github.com/OpenSourceBrain/BlueBrainProjectShowcase/blob/master/NMC/NeuroML2/README.md

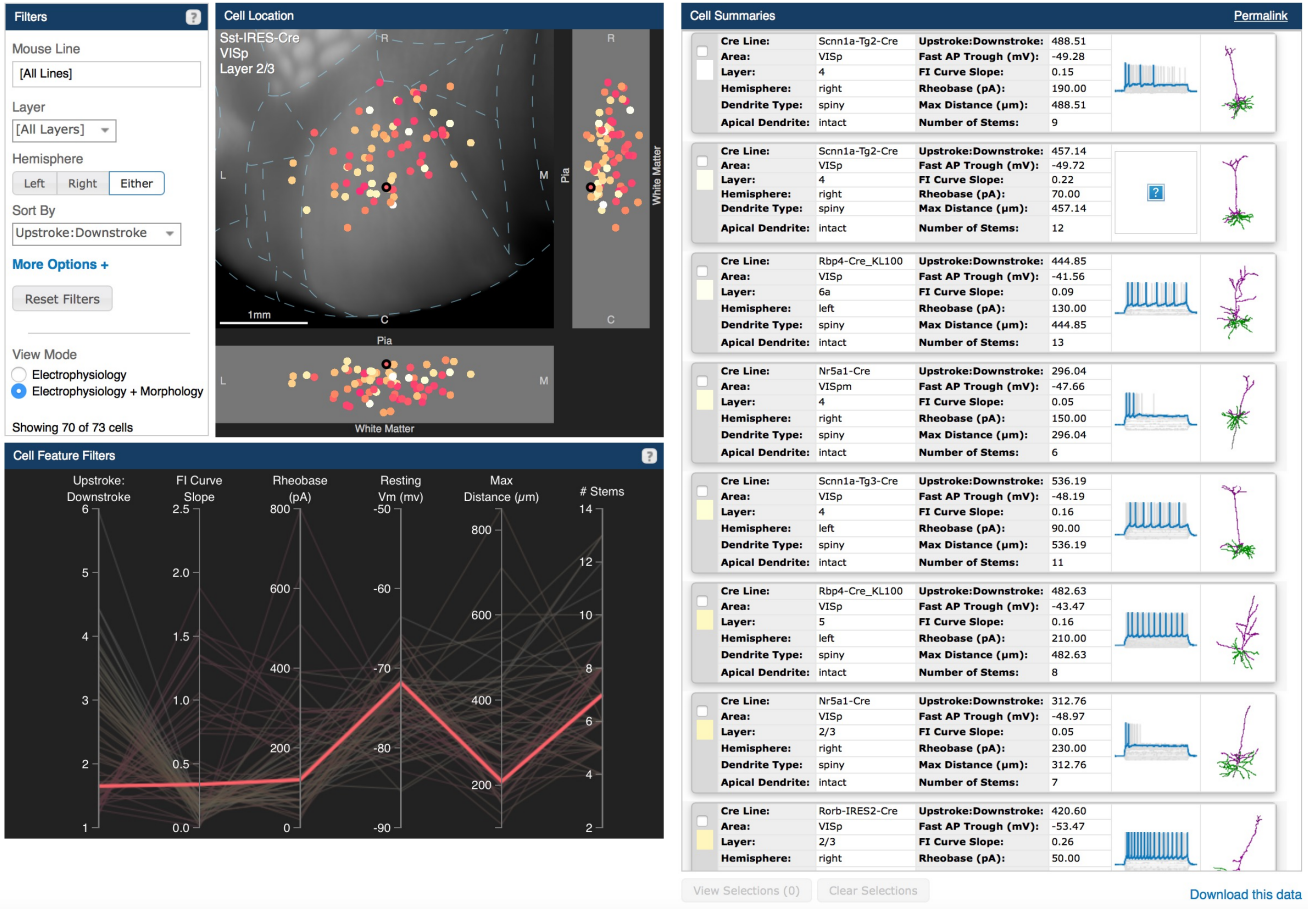
A file with the NeuroML2 versions of the model packages is available here



https://bbp.epfl.ch/nmc-portal/

Web Resources 3 - Allen Cell Types Database

https://celltypes.brain-map.org



Allen SDK and API

ALLEN BRAIN ATLAS

SOFTWARE DEVELOPMENT KIT

CONTENTS

QUICK SEARCH

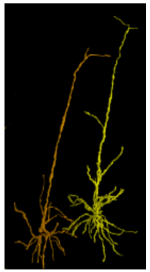
Go

Enter search terms or a module, class or function name.

WELCOME TO THE ALLEN SDK

The Allen Software Development Kit houses source code for reading and processing Allen Brain Atlas data. The Allen SDK focuses the Allen Cell Types Database and the Allen Mouse Brain Connectivity Atlas. Functionality relevant to other atlases is coming in future releases.

ALLEN CELL TYPES DATABASE



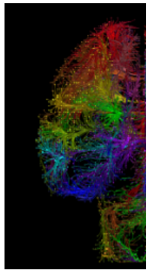
The Allen Cell Types Database contains electrophysiological and morphological characterizations of individual neurons in the mouse primary visual cortex. The Allen SDK provides Python code for accessing electrophysiology measurements (NWB files) for all neurons and morphological reconstructions (SWC files) for a subset of neurons.

The Database also contains two classes of models fit to this data set: perisomatic biophysical models produced using the NEURON simulator and generalized leaky integrate and fire models (GLIFs) produced using custom Python code provided with this toolkit.

The Allen SDK provides sample code demonstrating how to download neuronal model parameters from the Allen Brain Atlas API and run your own simulations using stimuli from the Allen Cell Types Database or custom current injections:

- *Perisomatic Biophysical Models*
- *Generalized LIF Models*

ALLEN MOUSE BRAIN CONNECTIVITY ATLAS



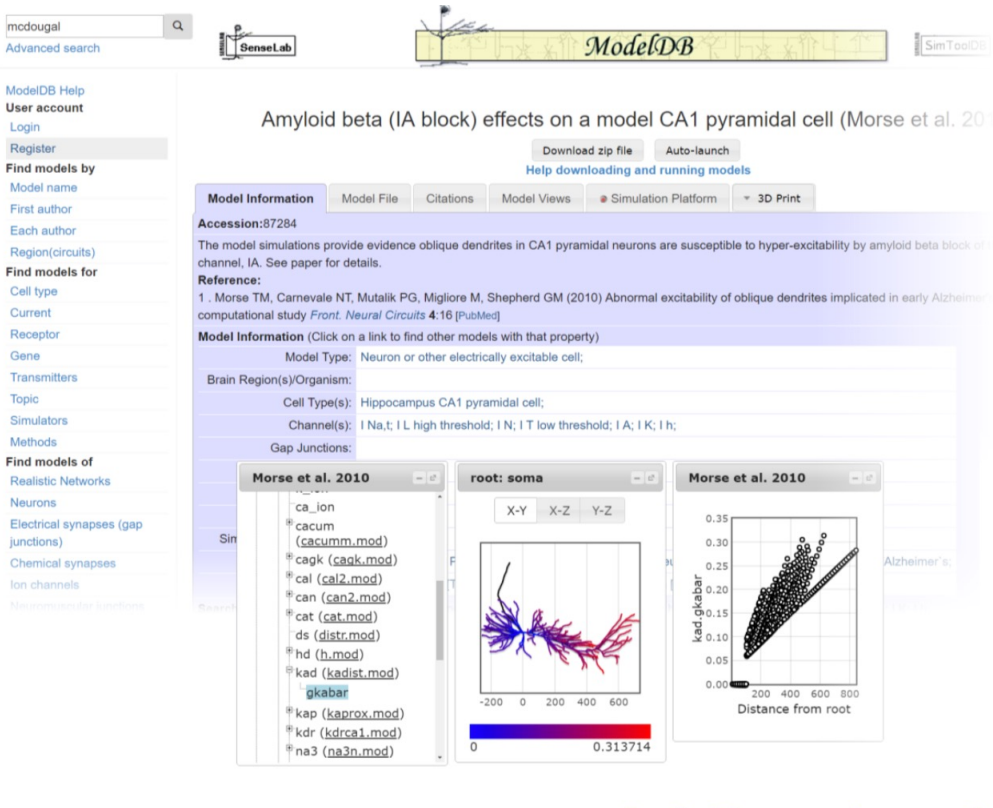
The Allen Mouse Brain Connectivity Atlas is a high-resolution map of neural connections in the mouse brain. Built on an array of transgenic mice genetically engineered to target specific cell types, the Atlas comprises a unique compendium of projections from selected neuronal populations throughout the brain. The primary data of the Atlas consists of high-resolution images of axonal projections targeting different anatomic regions or various cell types using Cre-dependent specimens. Each data set is processed through an informatics data analysis pipeline to obtain spatially mapped quantified projection information.

The Allen SDK provides Python code for accessing experimental metadata along with projection signal volumes registered to a common coordinate framework. This framework has structural annotations, which allows users to compute structure-level signal statistics.

See the mouse connectivity section for more details.

EPFL

Web Resource 4 – ModelDB (https://modeldb.yale.edu)




```
from neuron import h, rxd
import neuron.rxd.node as node
from matplotlib import pyplot
import time


h.load_file('stdrun.hoc')

soma = h.Section()
soma.L = 10
soma.diam = 10
soma.nseg = 11
dend = h.Section()
dend.connect(soma)
dend.L = 50
dend.diam = 2
dend.nseg = 51

def print_nodes():
    print ', '.join(str(v) for v in node._states)

print 'defining rxd'
region = rxd.Region(h.allsec(), nrn_region='i')
ca = rxd.Species(region, name='ca', d=1, charge=2, initial
reaction = rxd.Rate(ca, -ca * (1 - ca) * (0.3 - ca))

print 'initializing'
```


Morse TM, Carnevale NT, Mutalik PG, Migliore M, Shepherd GM (2010) Abnormal excitability of oblique dendrites implicated in early Alzheimer's: a computational study *Front. Neural Circuits* **4**:16 [PubMed]

References and models cited by this paper

Acker CD, White JA (2007) Roles of I(A) and morphology in action potential propagation in CA1 pyramidal cell dendrites. *J Comput Neurosci* **23(2)**:201-16 [Journal] [PubMed]

- **Roles of I(A) and morphology in AP prop. in CA1 pyramidal cell dendrites (Acker and White 2007) [Model]**

Anderton BH, Callahan L, Coleman P, Davies P, Flood D, Jicha GA, Ohm T, Weaver C (1998) Dendritic changes in Alzheimer's disease and factors that may underlie these changes. *Prog Neurobiol* **55**:595-609 [PubMed]

Andrasfalvy BK, Makara JK, Johnston D, Magee JC (2008) Altered synaptic and non-synaptic properties of

References and models that cite this paper

Culmone V, Migliore M (2012) Progressive effect of beta amyloid peptides accumulation on CA1 pyramidal neurons: a model study suggesting possible treatments *Front Comput Neurosci* **6**:52 [Journal] [PubMed]

- **CA1 pyramidal neurons: effects of Alzheimer (Culmone and Migliore 2012) [Model]**

McDougal RA, Morse TM, Hines ML, Shepherd GM (2015) ModelView for ModelDB: online presentation of model structure *Neuroinformatics* **13(4)**:459-70 [Journal] [PubMed]

- **ModelView: online structural analysis of computational models (McDougal et al. 2015) [Model]**

Lecture Summary

- Building “realistic” models of specific neurons requires many parameters (morphology, ion channels, ion channel kinetics, ion channel distributions)
- One way to make models meaningful and achievable is to derive as many parameters as possible from experimental data, when this is not possible, resorting to parameter optimization can be a viable approach
- Specifically, multi-objective optimization with feature-based fitness functions have shown to be effective for modeling morphologically detailed neuron classes across brain regions and species and are widely used
- Multiple online resources are available to provide useful data or data & corresponding models, but piecing together all relevant data for a particular neuron class of interest is still very time-consuming and an underconstrained problem

What you have learnt

- Different electrical behavior. Petilla nomenclature
- Challenges in modeling the neurons. Single neuron vs. neuron class, lack of data…
- BPAP, dendritic spike, dendritic computation
- Example of optimization. Cost function.
- Electrical features. Evolutionary algorithm. Pareto front.