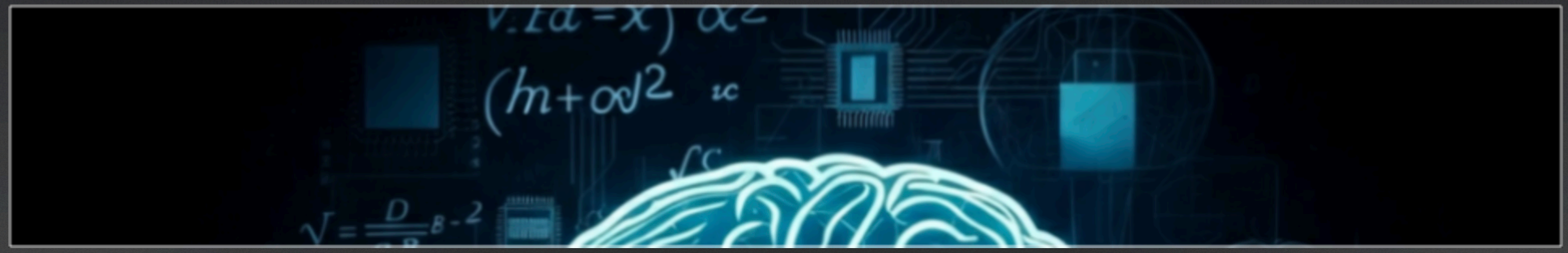
MODELLING NEURAL SYSTEMS



COMPUTATIONAL MODELLING OF NEURONS AND MICROCIRCUITS

Prof. Ing. Michele GIUGLIANO, PhD

Conductance-based neurons

ATTENDANCE TRACKING: **today's code is 33333**

(for my own statistical purposes)

Download for iPhone

Download for Android

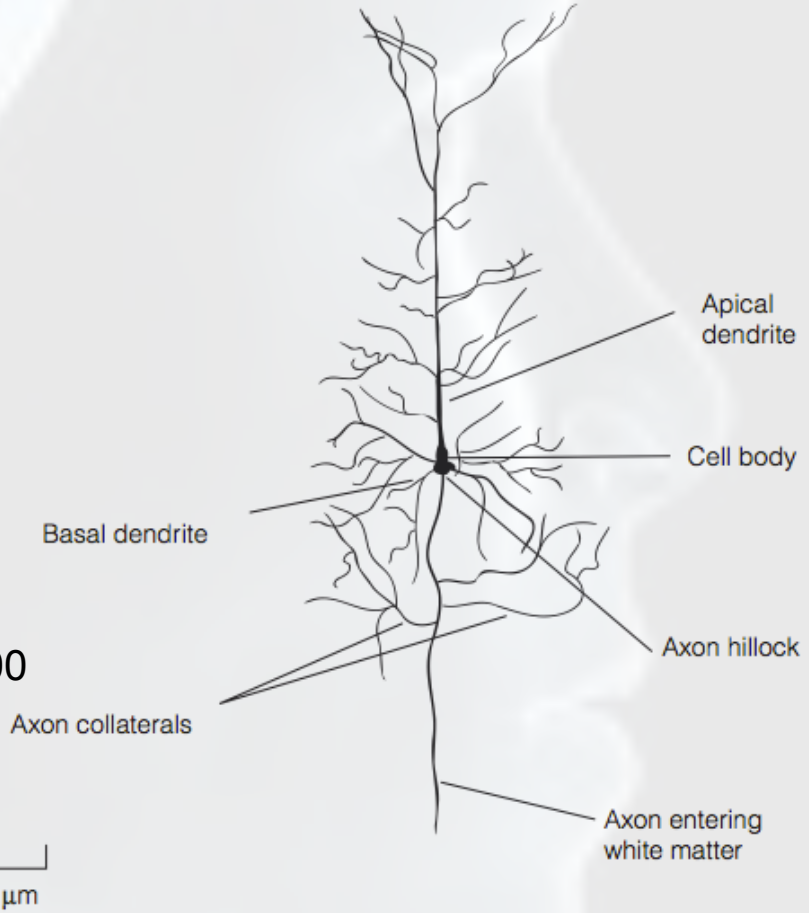
https://www.unimore.it/it/servizi/unimore-app

Inherent complexity of the brain

Human brain - just to mention a few highlights

- 100 000 000 000 nerve cells (**neurons**) (10^{11}) ;
- 1 000 000 000 000 000 connections (**synapses**) (10^{15}) ;
- Each neuron receives ~100 000 synapses from other neurons
- Many different types of neurons exists, in terms of size, shape and molecular properties (e.g. 12 types in the neocortex)
- Neurons are connected *ad hoc* to form functional circuits
- and communicate via **action potentials across their entire body and dendrites**
- The neurons of one human cerebral cortex would reach over 400'000 Km if placed end to end.

- **Are these bugs/leftovers or features??**



Levels of organization / of math. description

Complexity, N parameters
biophys. realism, data valid.

Increas. abstraction
approxim., phenomenology

Molecular level

Cellular level

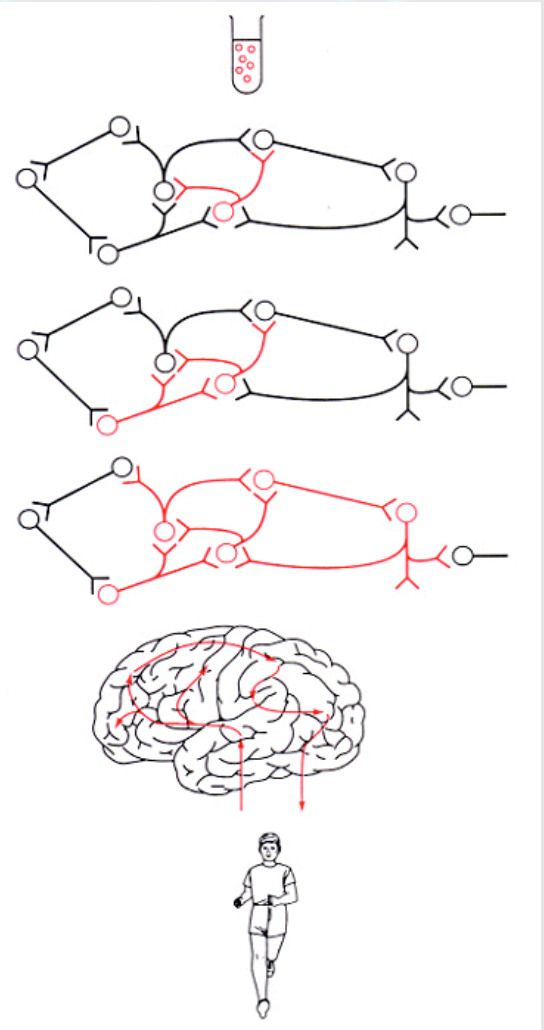
Microcircuit level

Population level

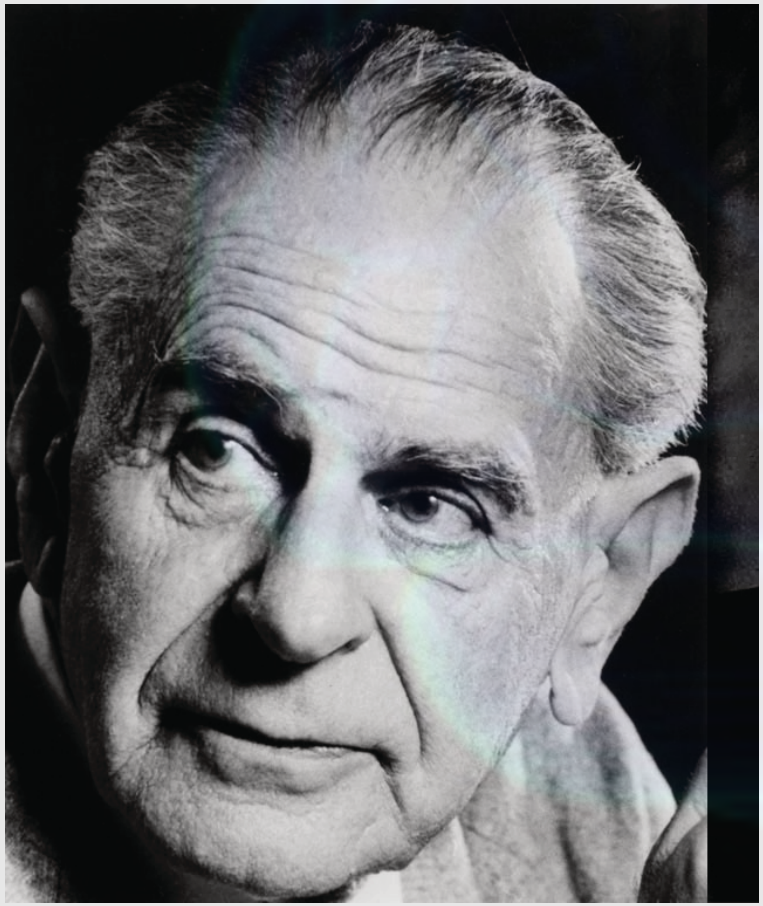
System level

Whole-brain level

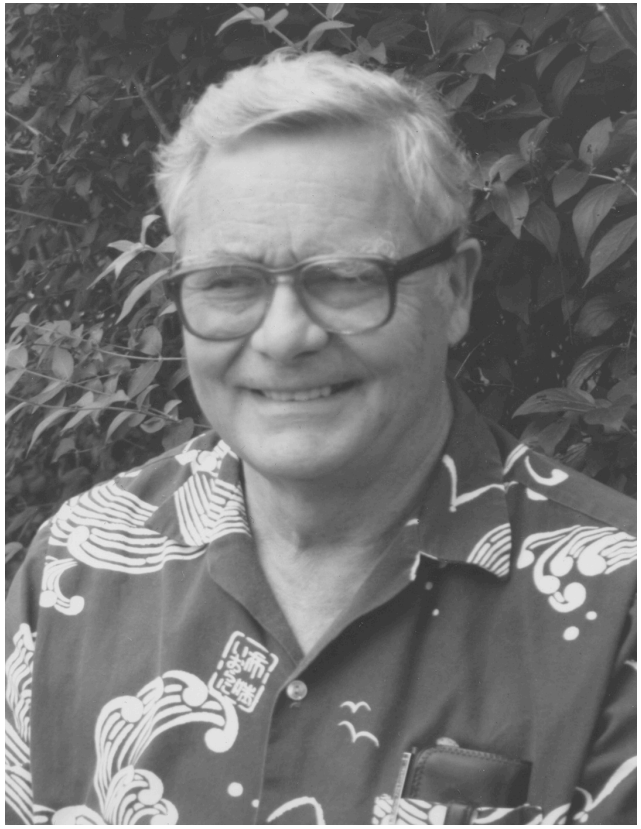
Behavioral level

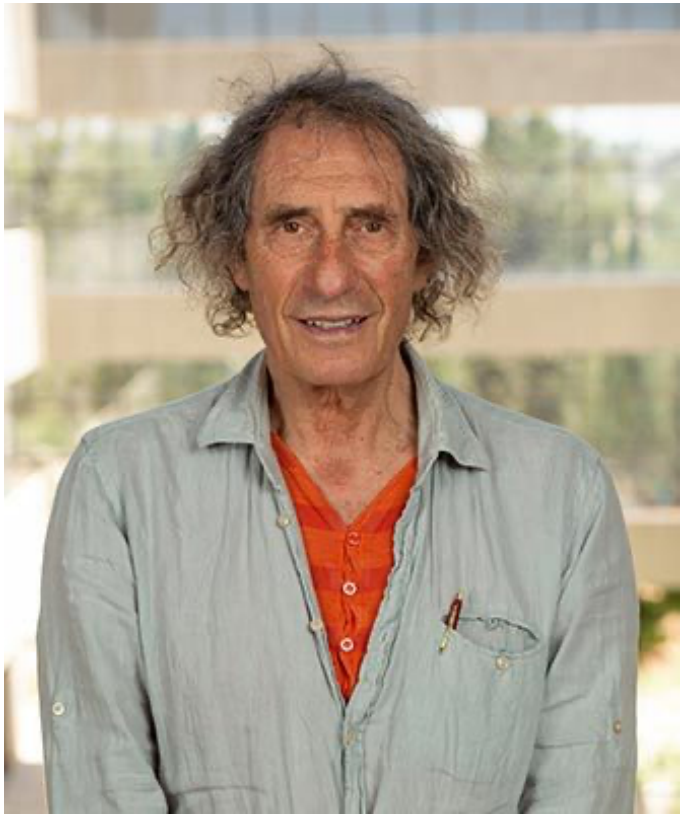


Karl Popper (1919) and Louis Lapique (1907)



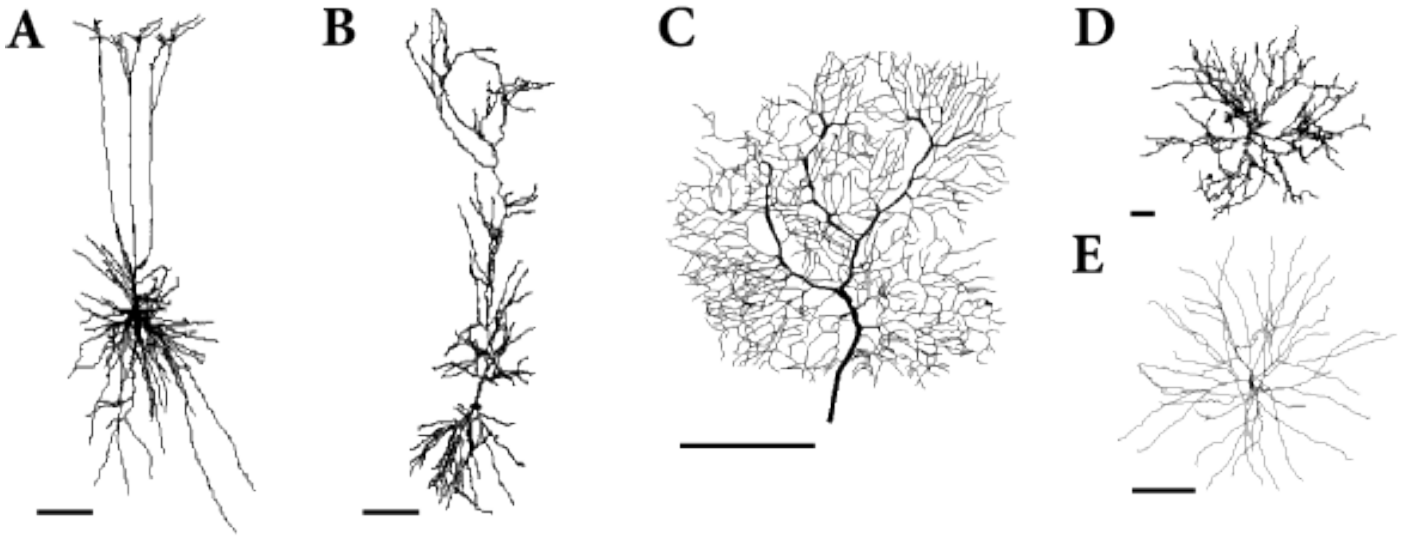
Wilfrid Rall, Idan Segev, Erik De Schutter



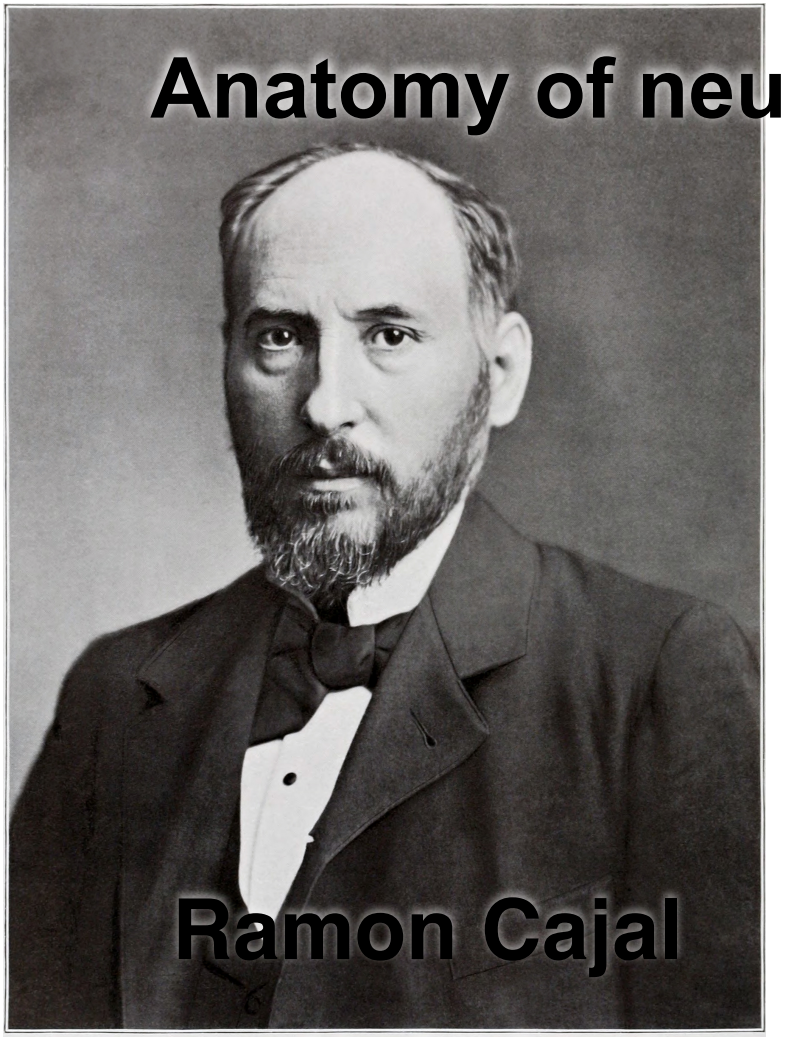


Morphology, diversity, complexity: does it matter?

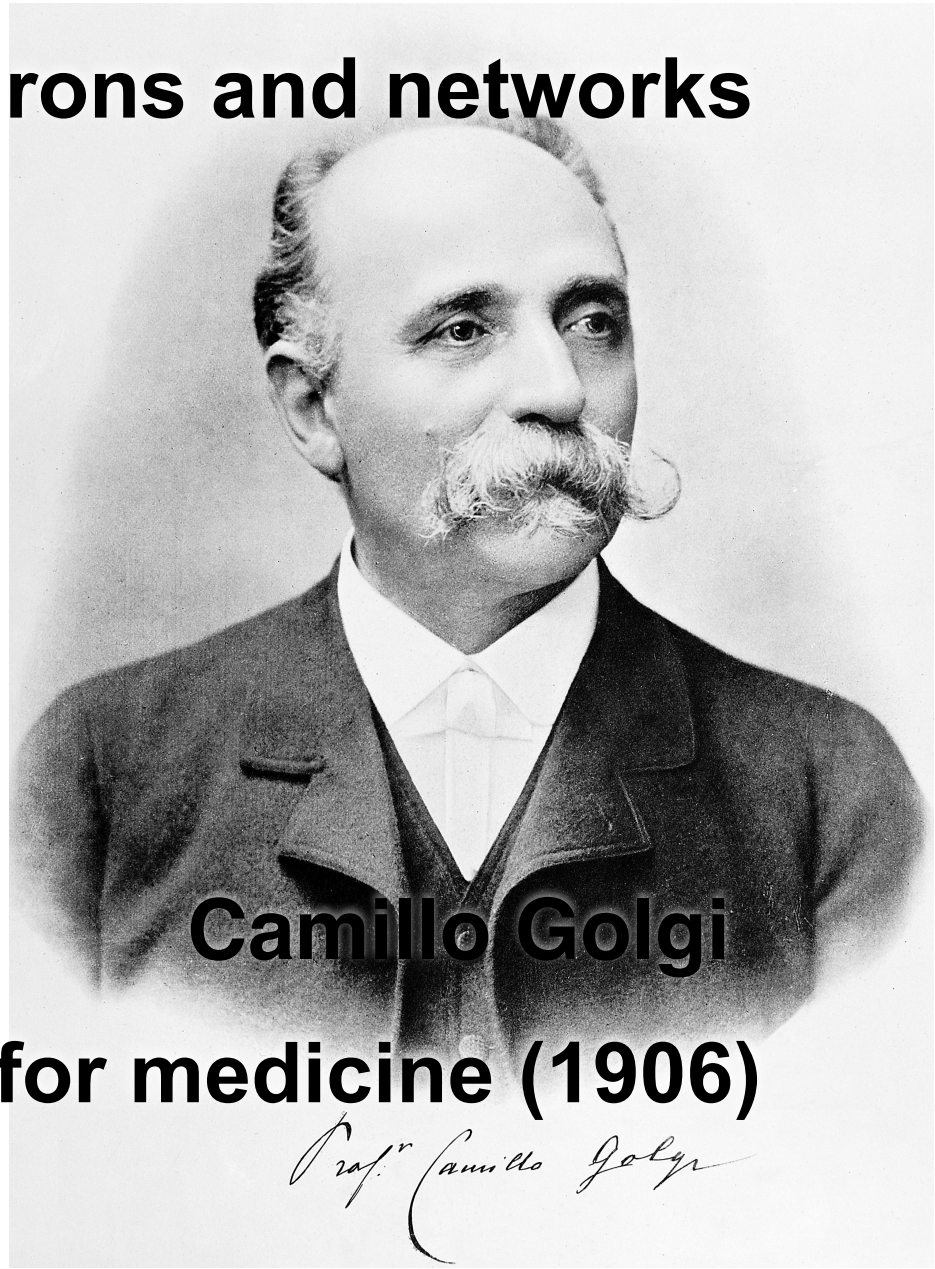
LV pyramidal neuron



Anatomy of neurons and networks



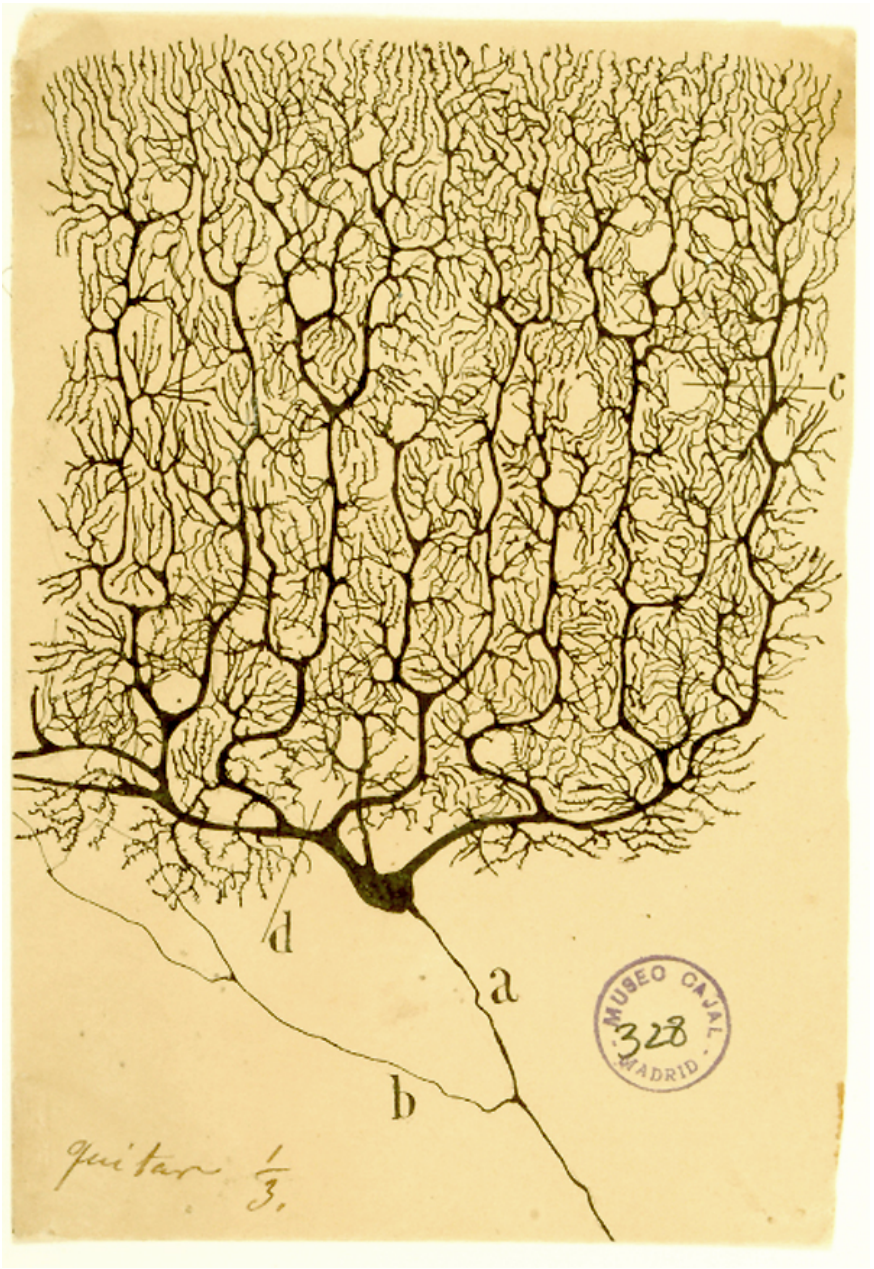
Ramon Cajal

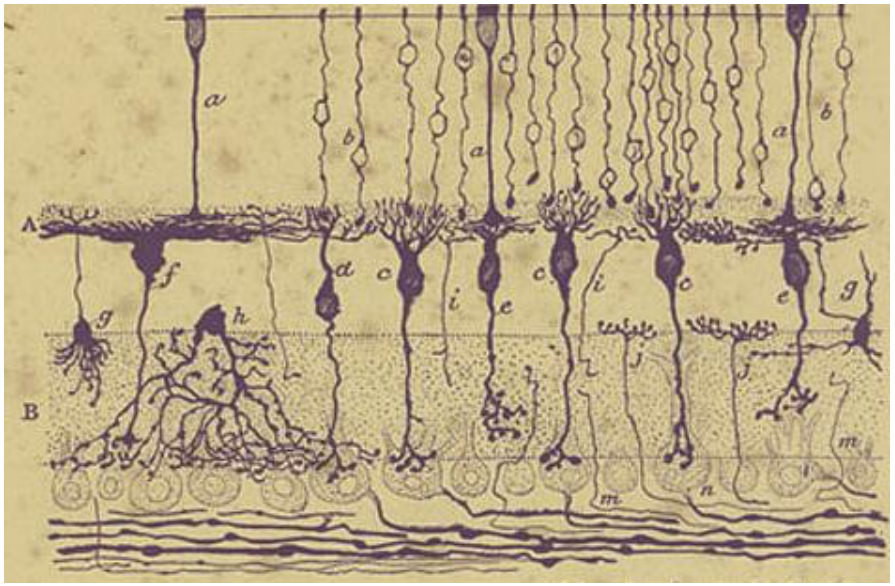


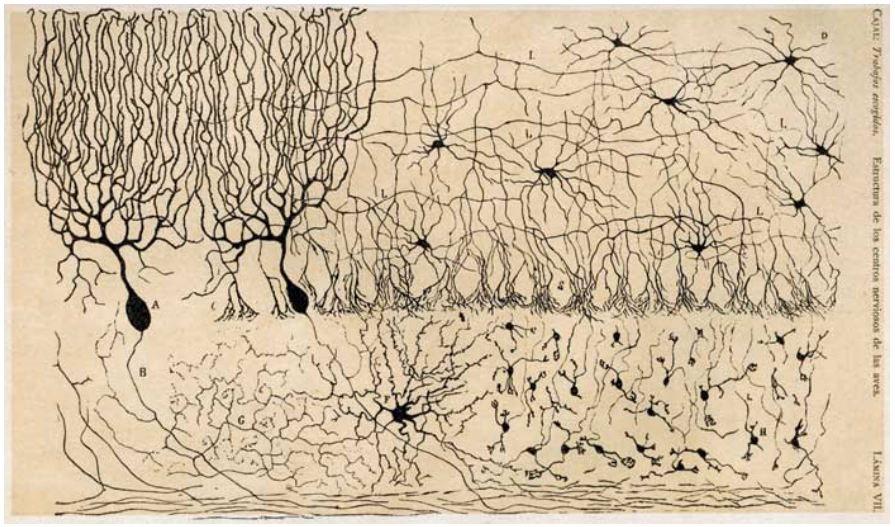
Camillo Golgi

Nobel prize for medicine (1906)

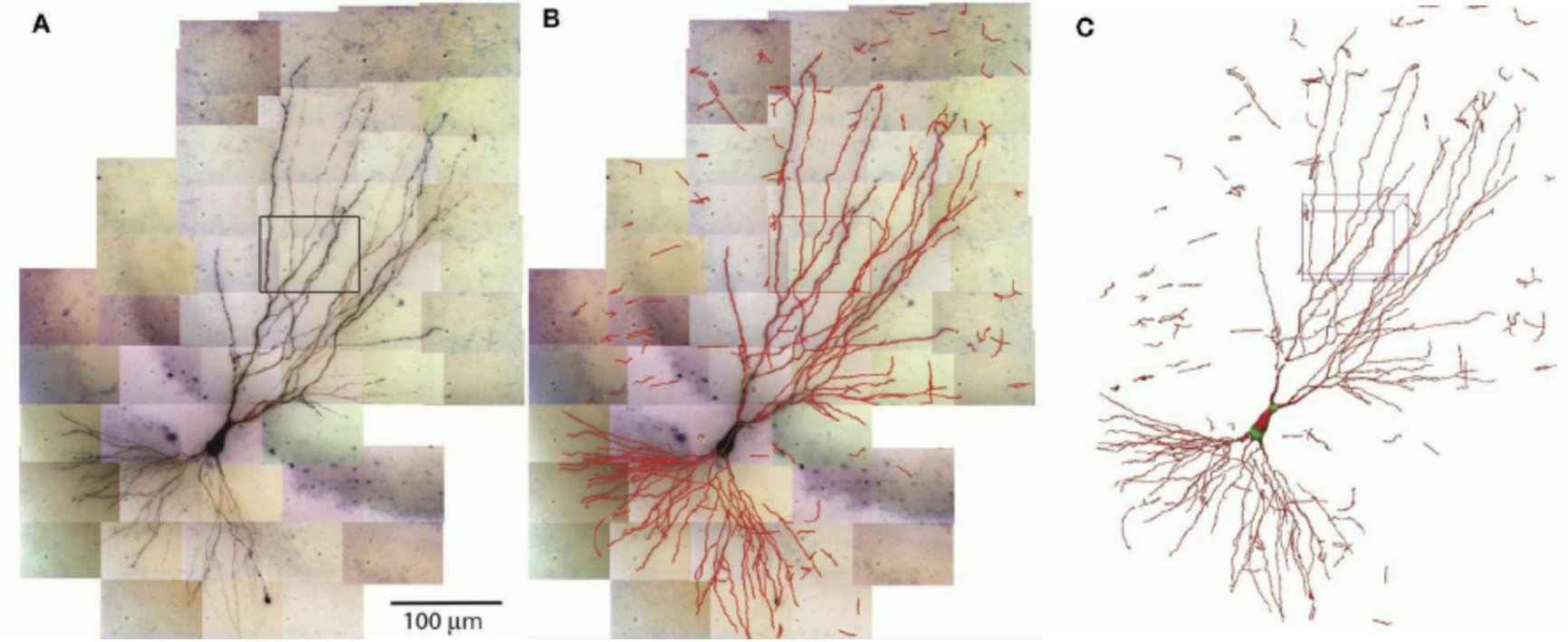
S. Ramon Cajal




A
B

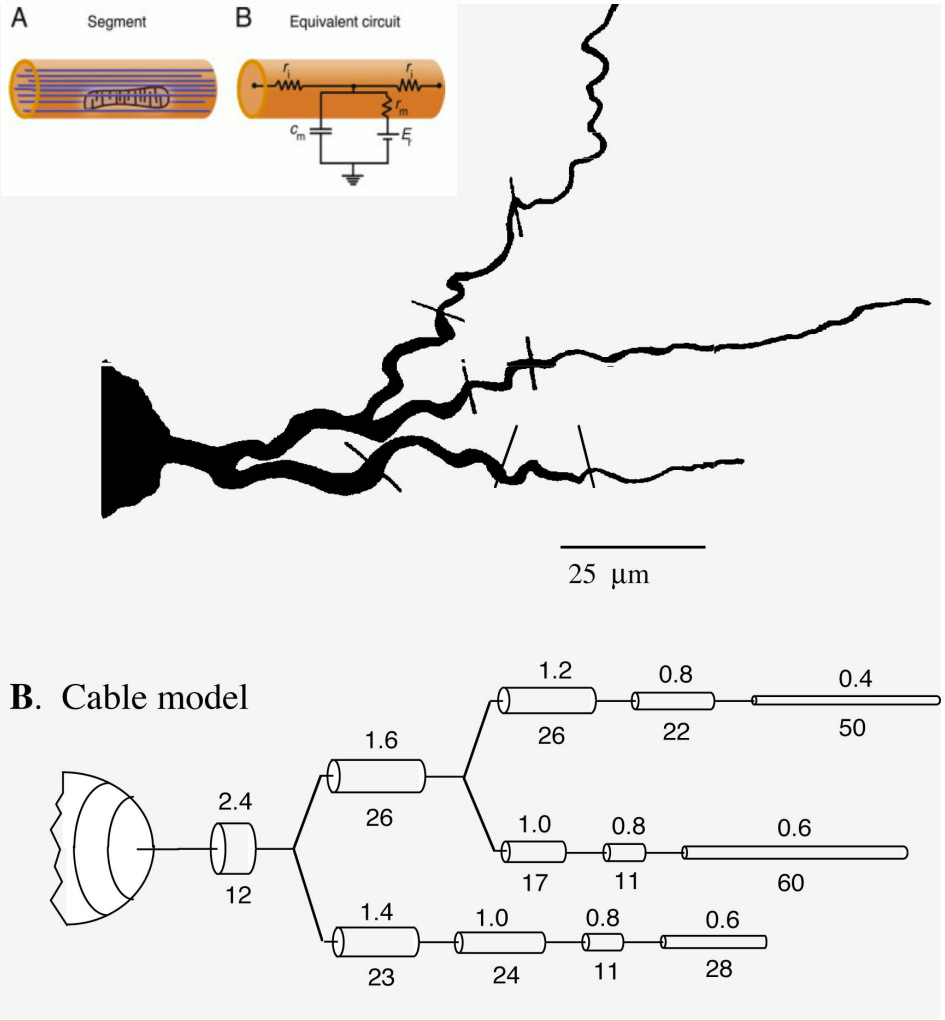

CAJAL: Tratado elem. Estructura de los centros nerviosos de las aves. Lámina VII.

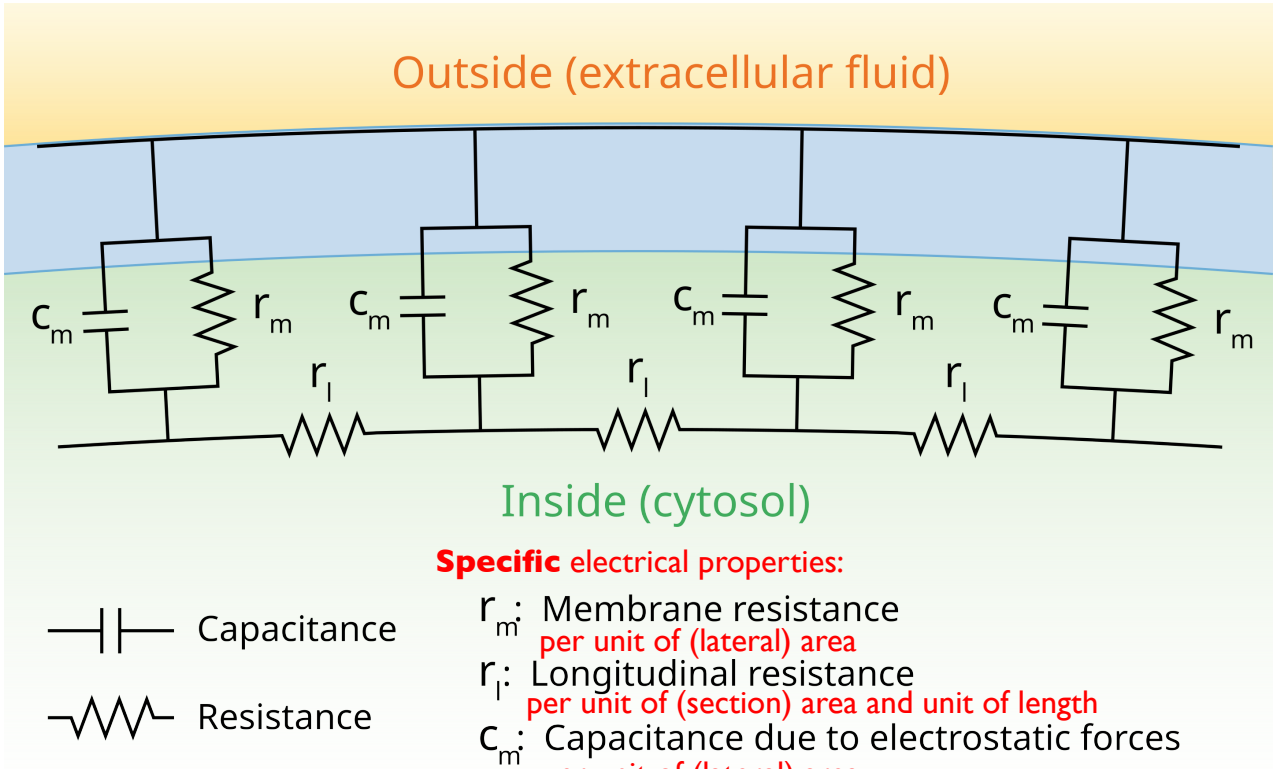
Digital reconstruction of neuronal morphologies



Jin *et al.* (2019) Frontiers in Neuroinformatics

Conductance-based, multi-compart. models





$$c = 1\mu F cm^{-2} \qquad r_m = 20k\Omega cm^2 \qquad r_i = 200\Omega cm$$

$$cr_m = 1\mu F cm^{-2} 20k\Omega cm^2 = 20ms$$

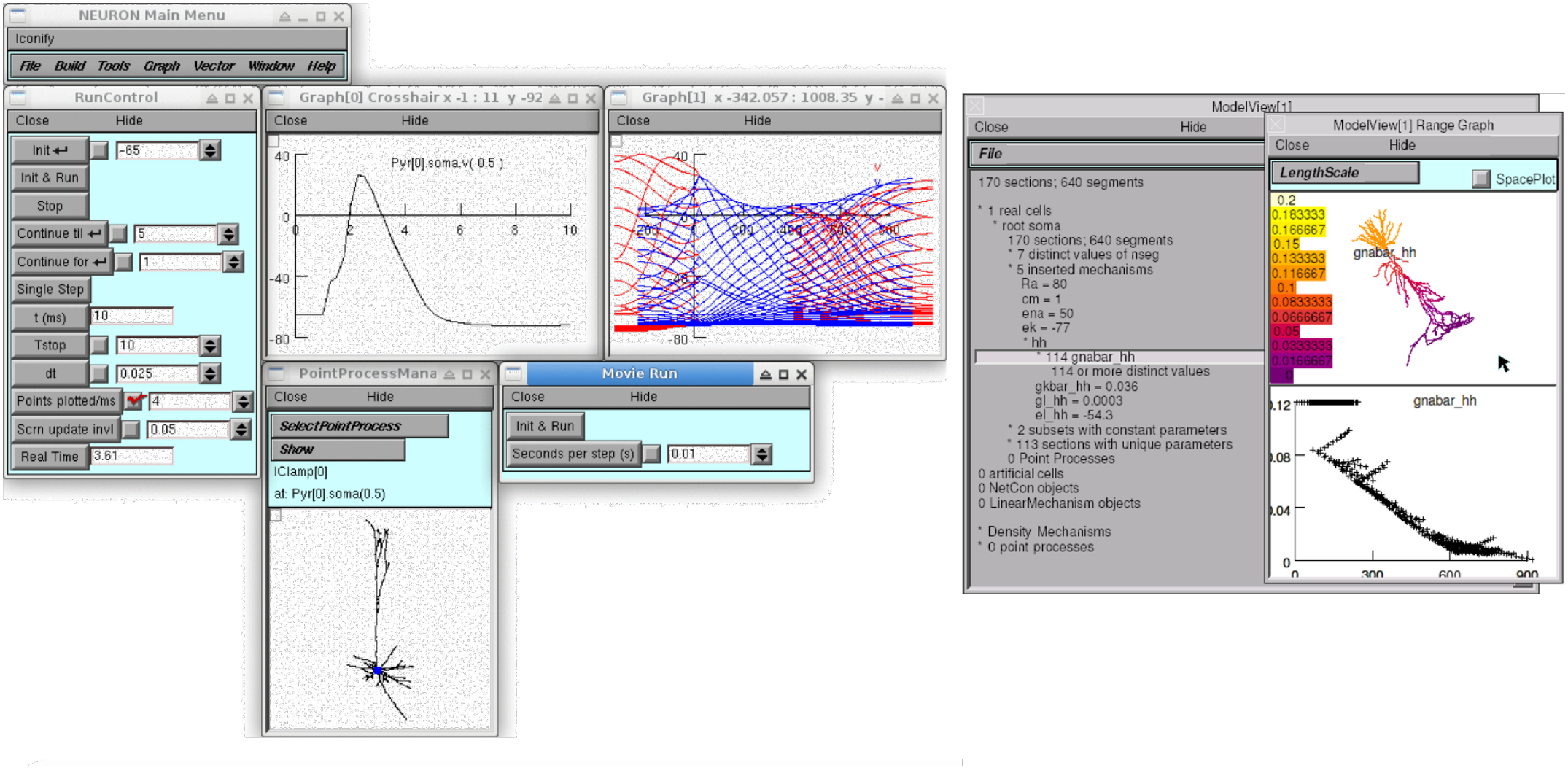
Introduction to the NEURON simulator

The **NEURON Simulation Environment** has been designed for modeling single single neurons and for modeling networks of neurons. It is particularly well-suited to explore problems which are closely linked to experimental data.

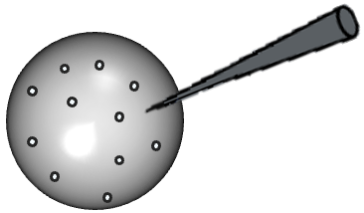
NEURON was build and maintained by a group at Yale University (Ted Carnevale and Michael Hines), with recent contributions from the Blue Brain Project (EPFL, Lausanne, Switzerland). **NEURON v. 9.0 is available today.**
Neuron is written in C & C++, the interface with the simulator is with HOC, but a python wrapper was build and now commonly used instead of HOC (NEURON is used as a python package).

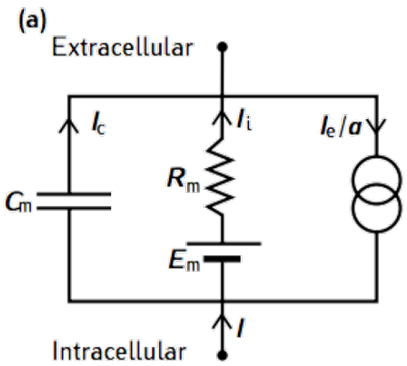
In addition User-defined mechanisms, such as voltage- and ligand-gated ion channels, are used in order to expand NEURON (mod files, need to be compiled).

NEURON's own Graphical User Interface



Single compartmental model - passive soma

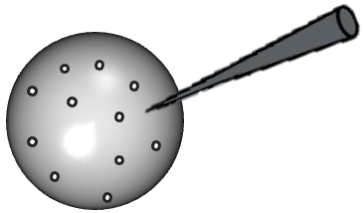


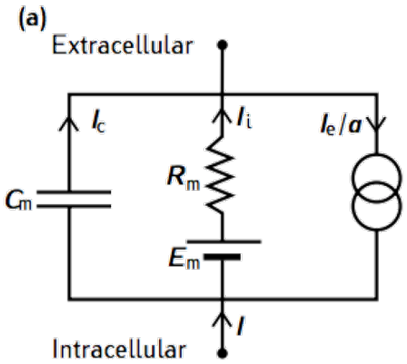



```
from neuron import n
# create model
soma  = n.Section(name="soma")
soma.L    = 10 # length µm
soma.diam = 10 # diameter µm
soma.insert('pas')    # add passive properties
soma.g_pas = 1/10000 # set the specific membrane
                     # res. to 10000 ohm*cm^2
```

```
# current clamp
stim = n.IClamp(soma(0.5))
stim.delay = 20 # start of the current injection
stim.dur   = 100 # duration (ms)
stim.amp   = 0.01 # amplitude (nA)
```

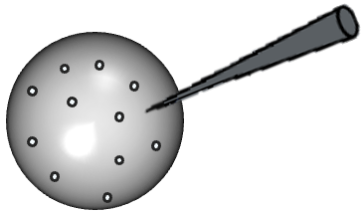
Single compartmental model - passive soma



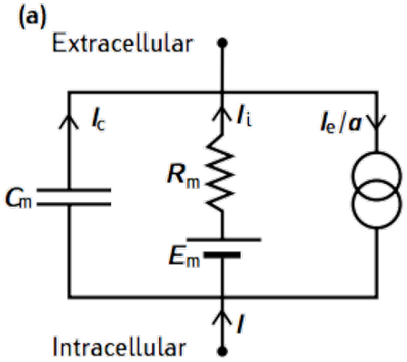


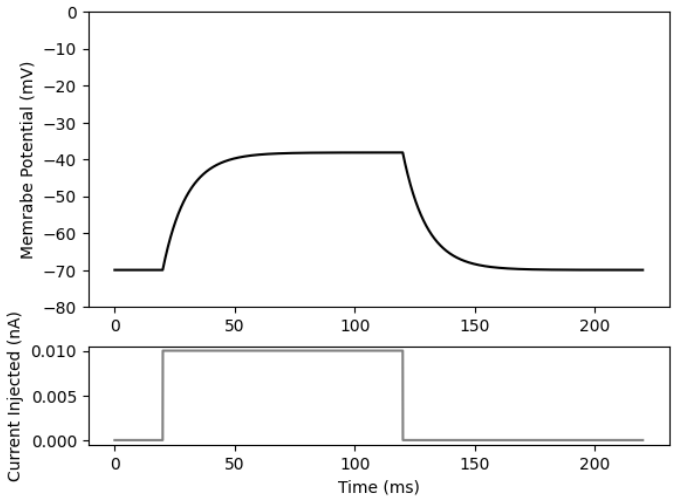

```
# record voltage of some and injected current
# and the time
soma_v = n.Vector()
soma_v.record(soma(0.5)._ref_v)

stim_current = n.Vector()
stim_current.record(stim._ref_i)

t = n.Vector()
t.record(n._ref_t)

# run simulation
h.tstop = 220 # set the simulation time
h.dt = 0.025
h.v_init = -70
h.run()
```

Single compartmental model - passive soma

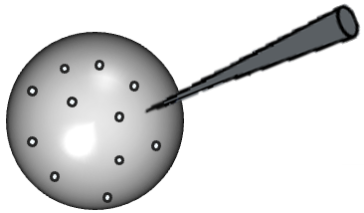


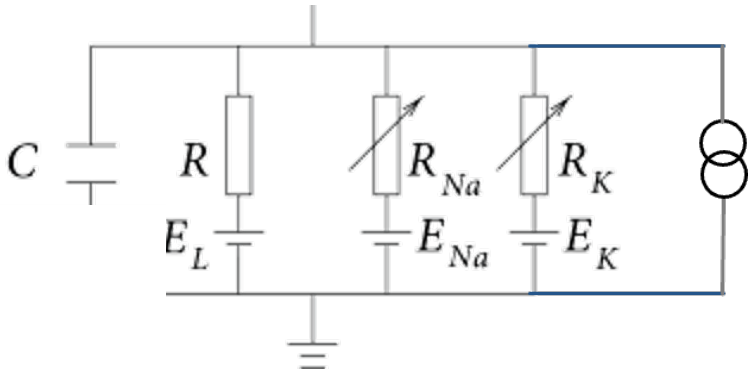
```
# plotting
import matplotlib.pyplot as plt

f, (ax0, ax1) = plt.subplots(2,1, gridspec_kw =
{'height_ratios':[3, 1]})
ax0.plot(t,soma_v, 'k')
ax1.plot(t,stim_current, 'gray', label='I (nA)')
```

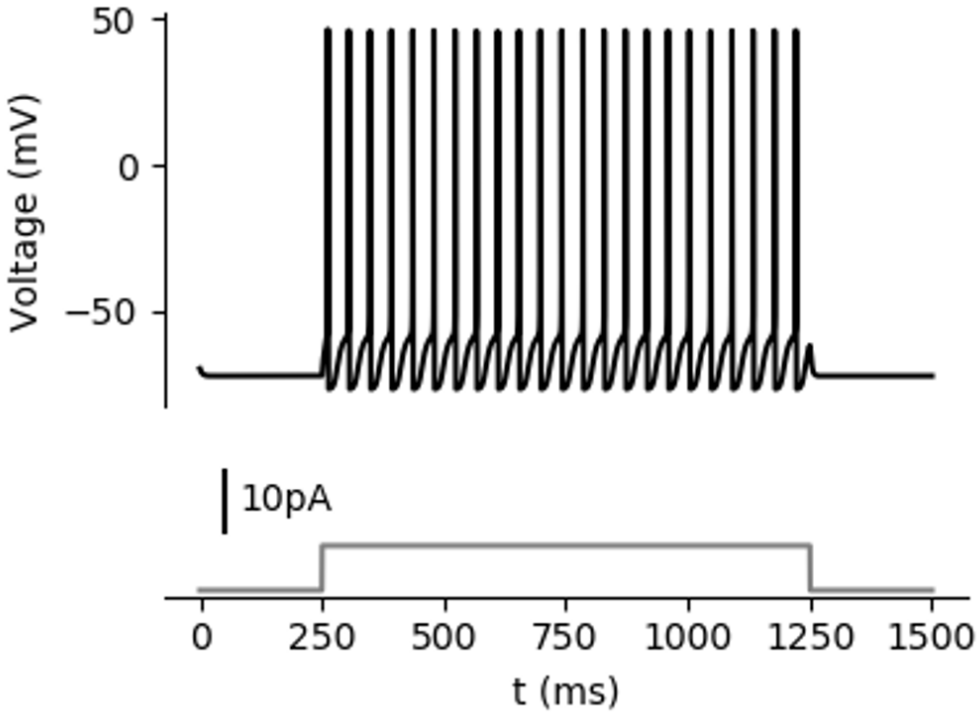




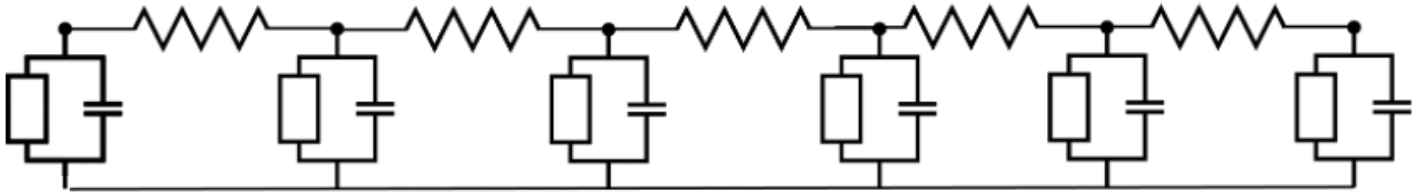
Single compartmental model - active soma

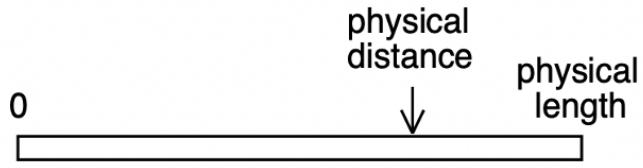


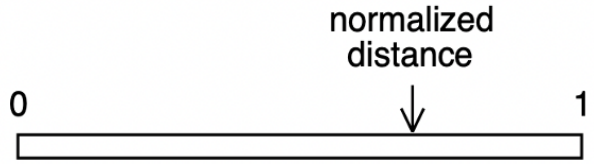




Multicompartmental model - Ball + Stick







Example:

Length of a section named dend is 10 μm, if we want to examine the voltage at location which is after 7 μm from point 0, we should do:

dend(0.7).v

Also if the length of the section is 100 μm and we want to examine location 70 μm we would write dend(0.7).

Multicompartmental model - Ball + Stick