

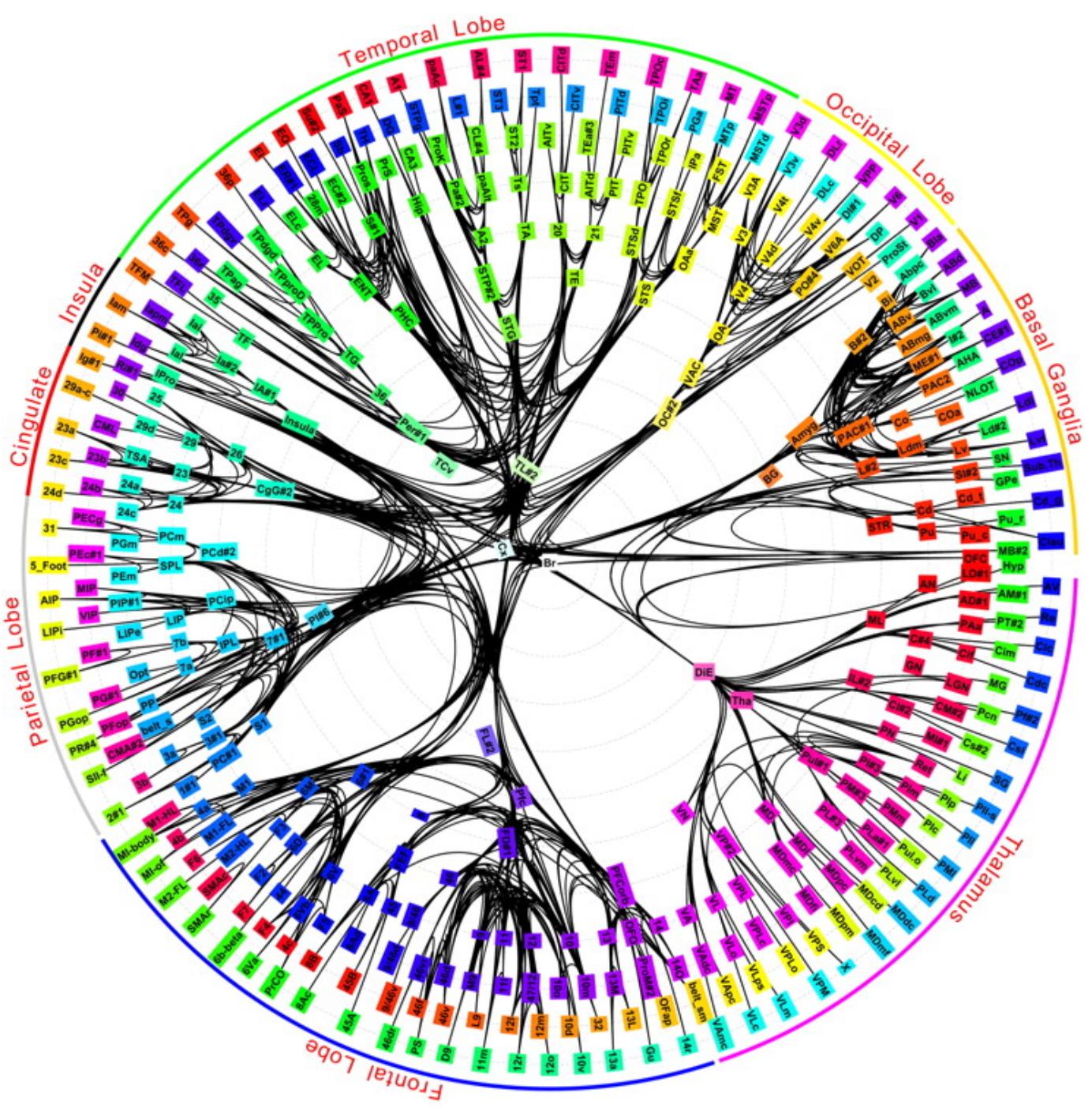
Modeling intracellular neuronal dynamics

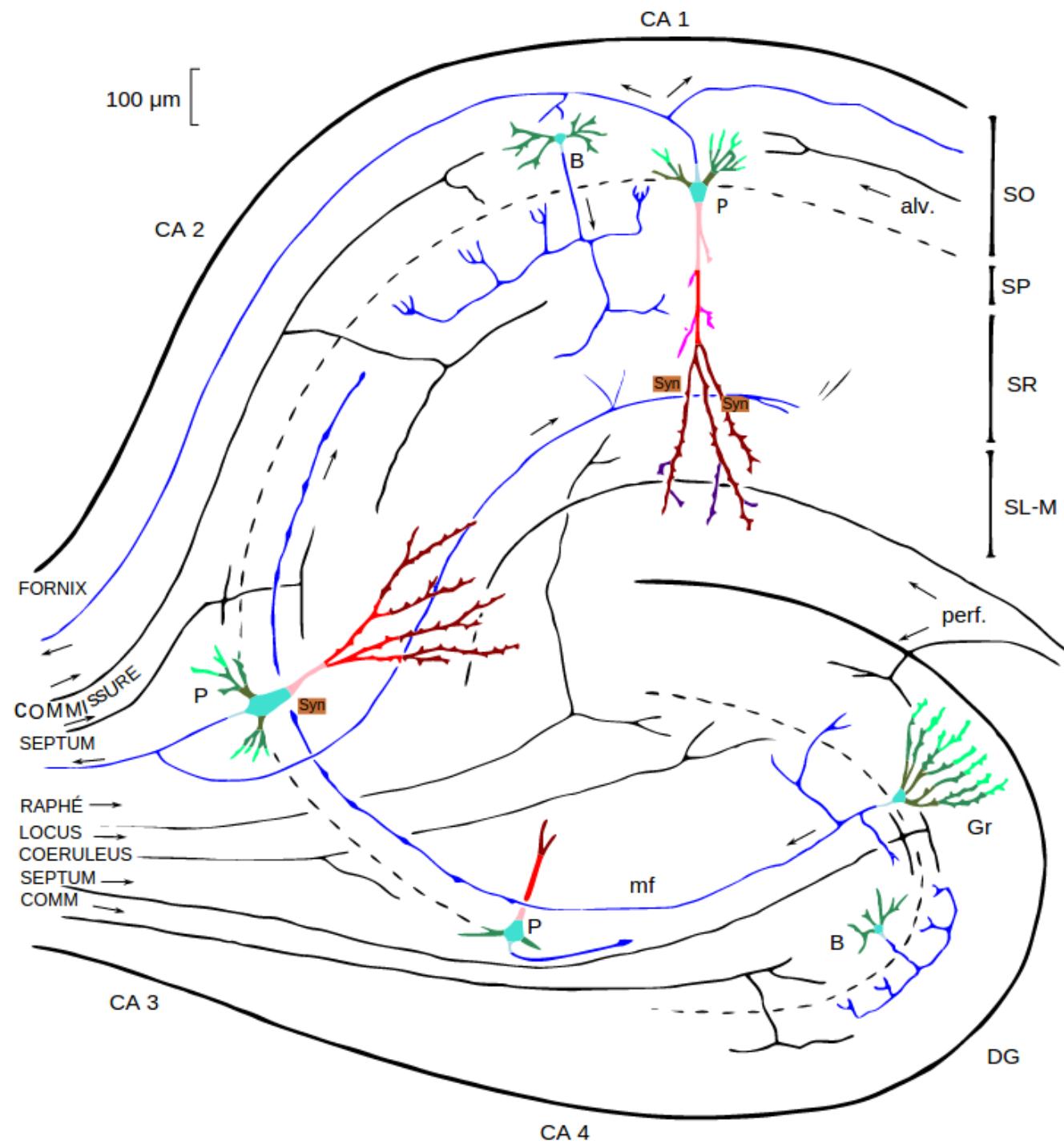
Robert A McDougal

13 July 2018

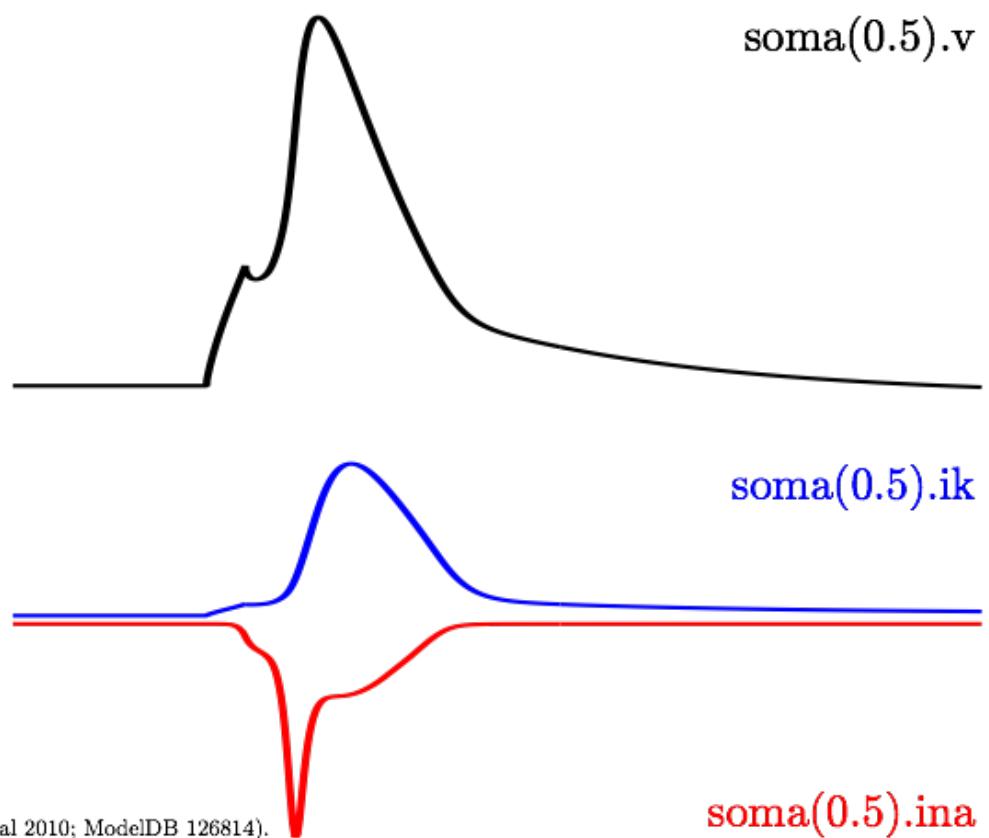
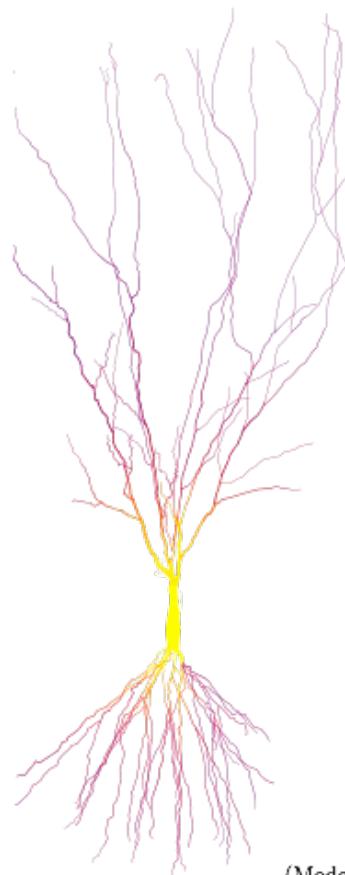
How does the brain work?

A multiscale story.



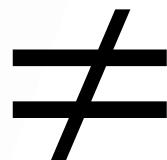
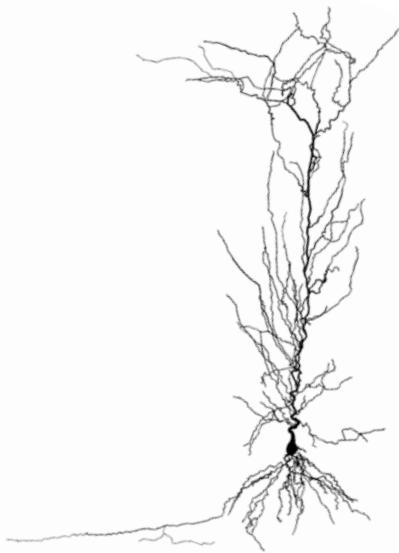


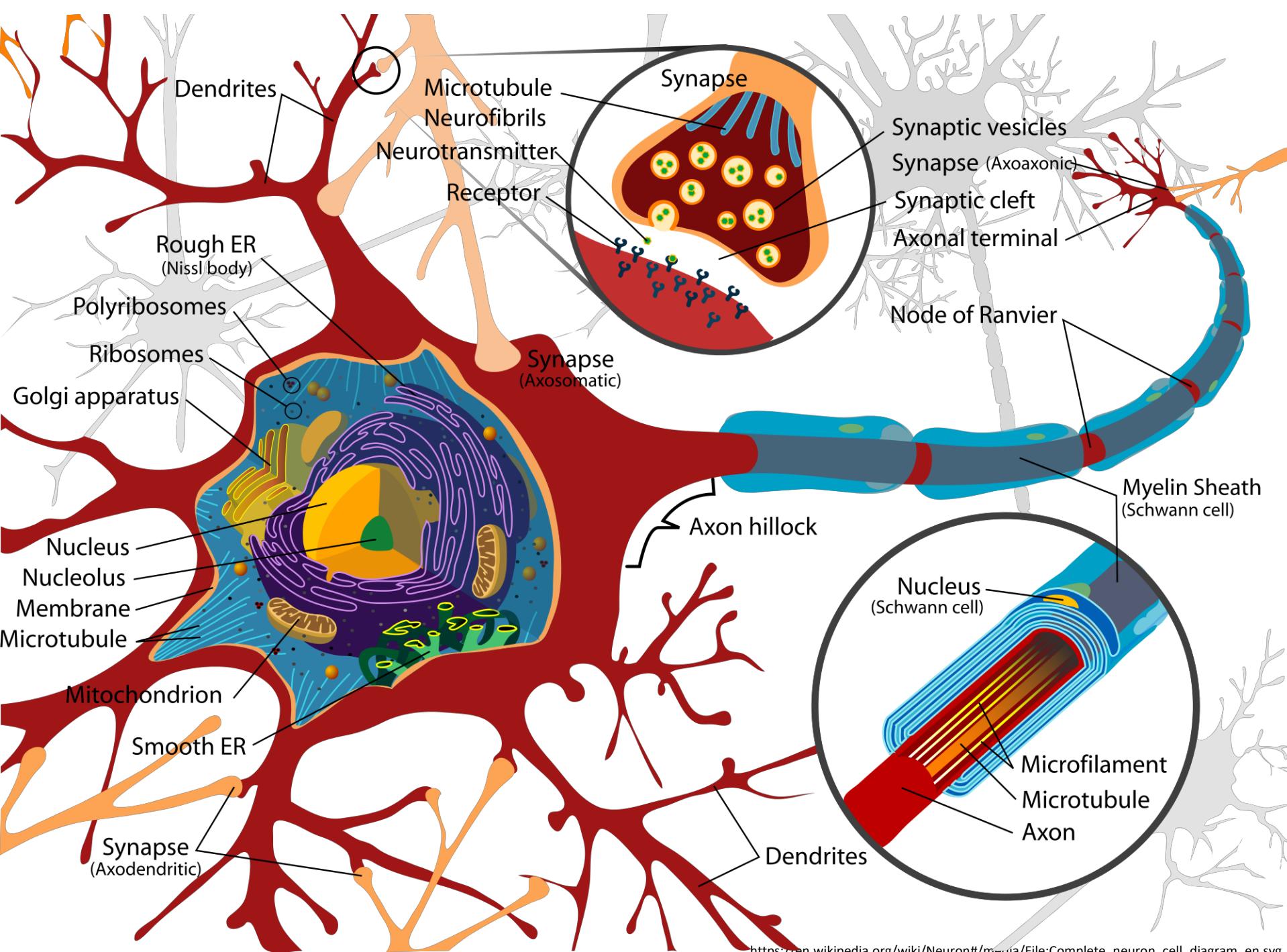
Neurons generate action potentials by moving ions across their membrane.



(Model from Saifulina et al 2010; ModelDB 126814).

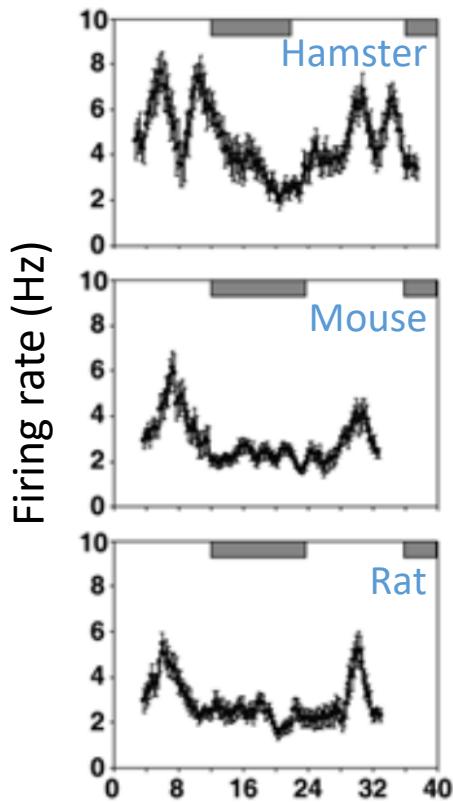
A neuron is not a transistor





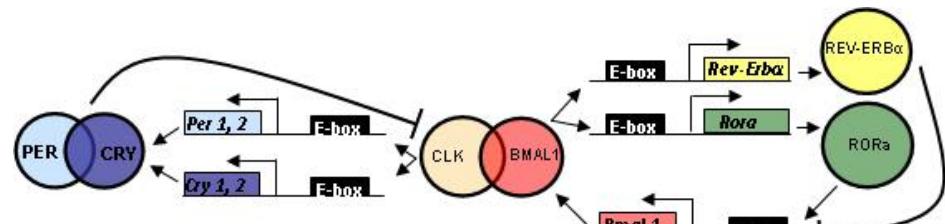
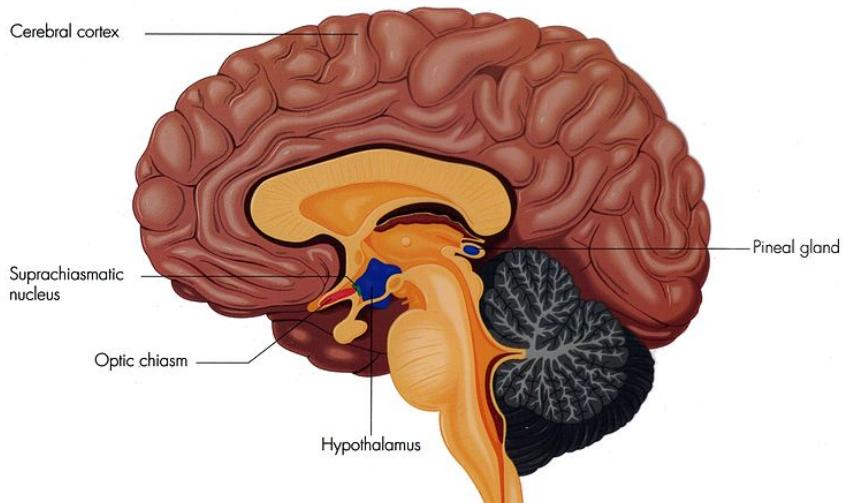
Neurons have state

(example: protein oscillations in the SCN)



Burgoon et al 2004

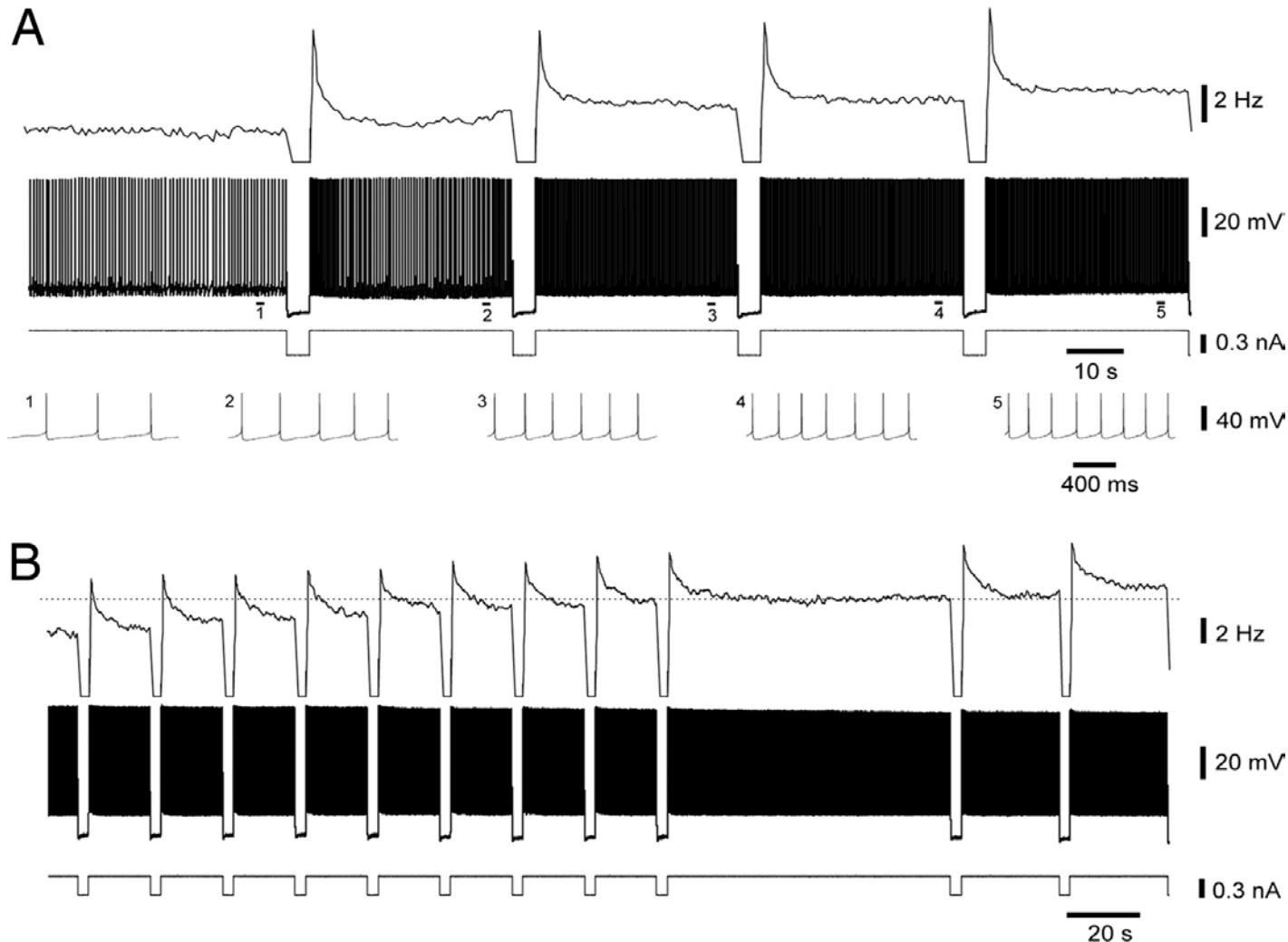
https://commons.wikimedia.org/wiki/File:Suprachiasmatic_Nucleus.jpg



https://en.wikipedia.org/wiki/Suprachiasmatic_nucleus#/media/File:SCN_mam.jpg

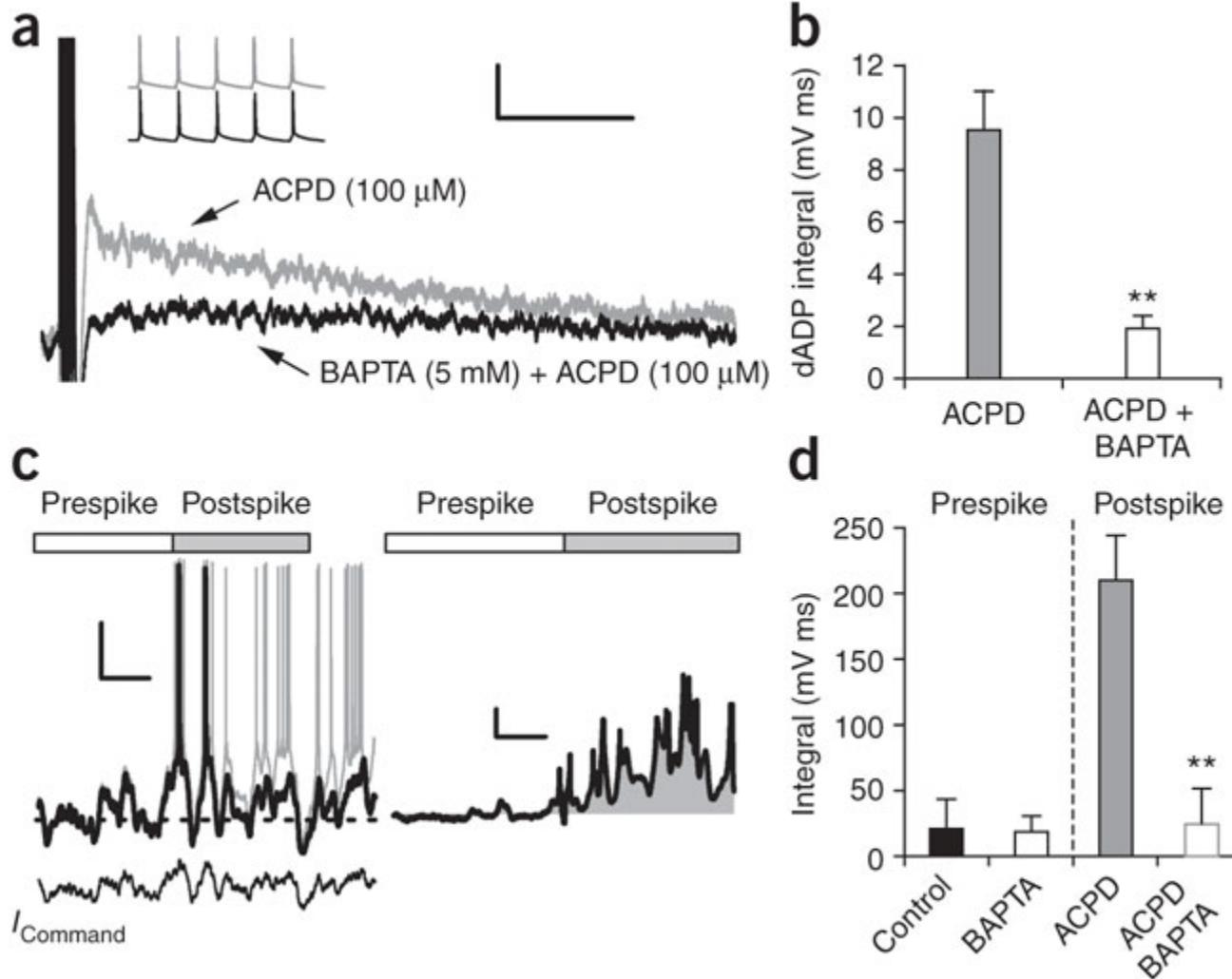
Neurons have state

(example: HAGPA in PFC)



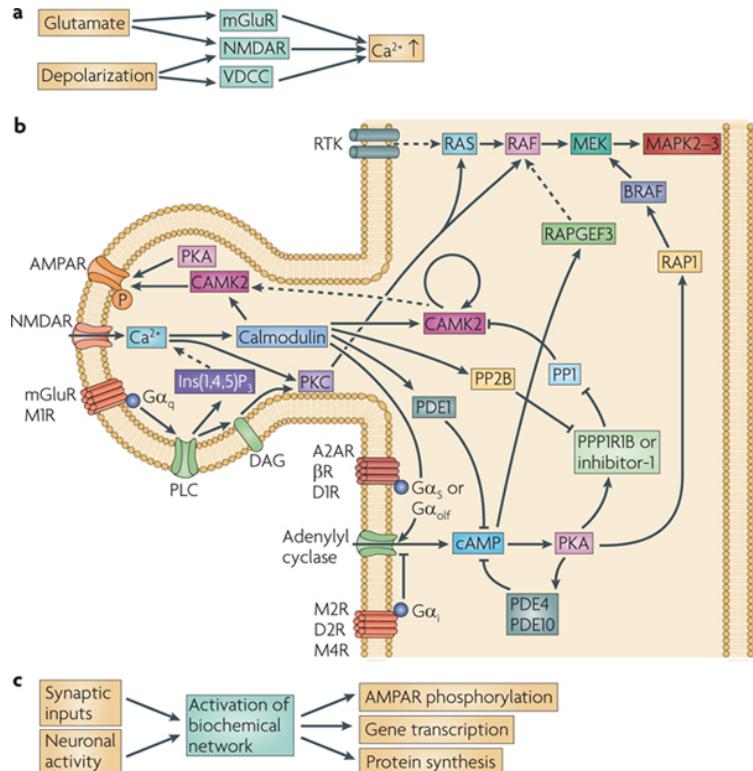
Neurons have state

(example: intracellular calcium)



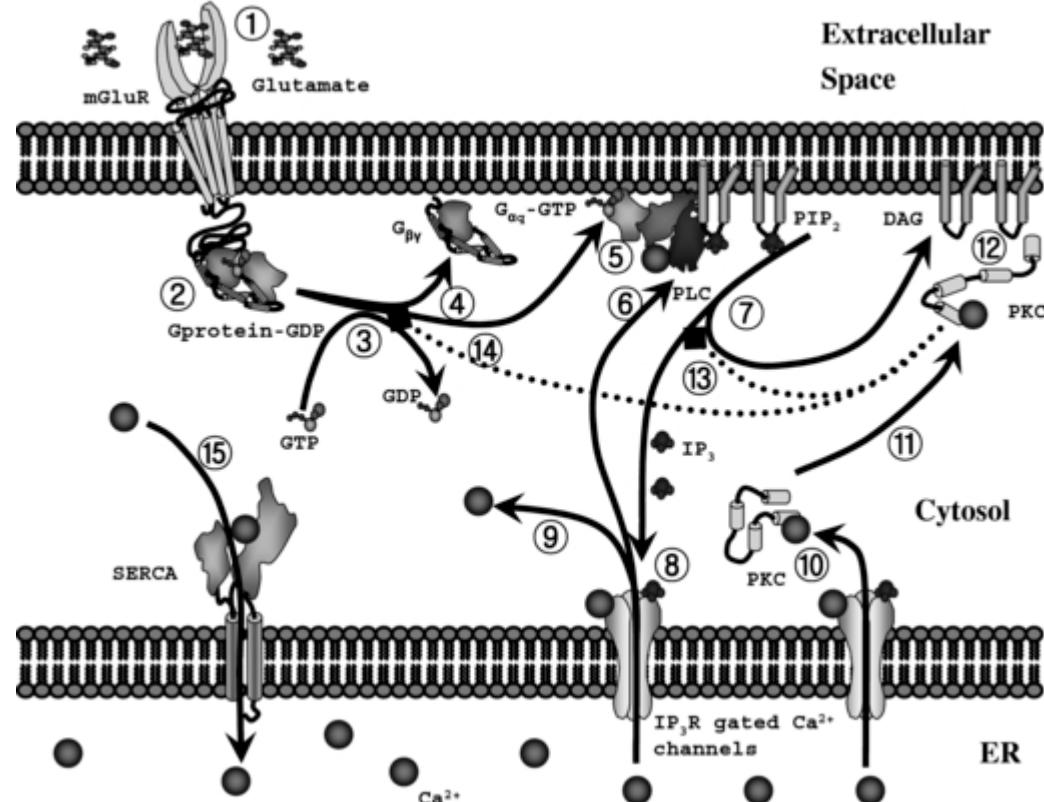
Neurons have state

(example: synaptic pathways)



Nature Reviews | Neuroscience

Jeanette Hellgren Kotaleski & Kim T. Blackwell 2010.



Minchul Kang and Hans G Othmer 2007.

How do we model this?

“**Reaction–diffusion systems** are mathematical models which explain how the **concentration** of one or more substances distributed in space changes under the influence of two processes: **local chemical reactions** in which the substances are transformed into each other, and **diffusion** which causes the substances to spread out over a surface in space.”

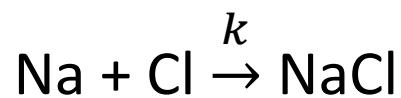
Mass-Action kinetics

The model

- A reaction's product is formed at a rate proportional to the concentration of the reactants.

Example

- Consider the reaction



- Then:

$$[\text{Na}]' = -k[\text{Na}][\text{Cl}]$$

$$[\text{Cl}]' = -k[\text{Na}][\text{Cl}]$$

$$[\text{NaCl}]' = k[\text{Na}][\text{Cl}]$$

Conservation of mass.

Matter is neither created nor destroyed by reactions.

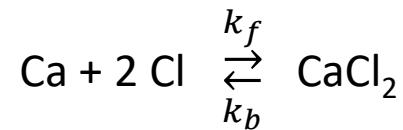
In our equations, this means:

$$[\text{Na}] + [\text{NaCl}] = \text{constant}$$

$$[\text{Cl}] + [\text{NaCl}] = \text{constant}$$

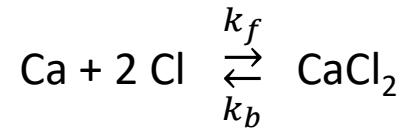
Exercise

Use the law of mass-action to write a system of equations describing the formation of *calcium chloride*:



Exercise

Use the law of mass-action to write a system of equations describing the formation of *calcium chloride*:



Answer:

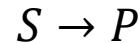
$$[\text{Ca}]' = -k_f [\text{Ca}][\text{Cl}]^2 + k_b [\text{CaCl}_2]$$

$$[\text{Cl}]' = -2k_f [\text{Ca}][\text{Cl}]^2 + 2k_b [\text{CaCl}_2]$$

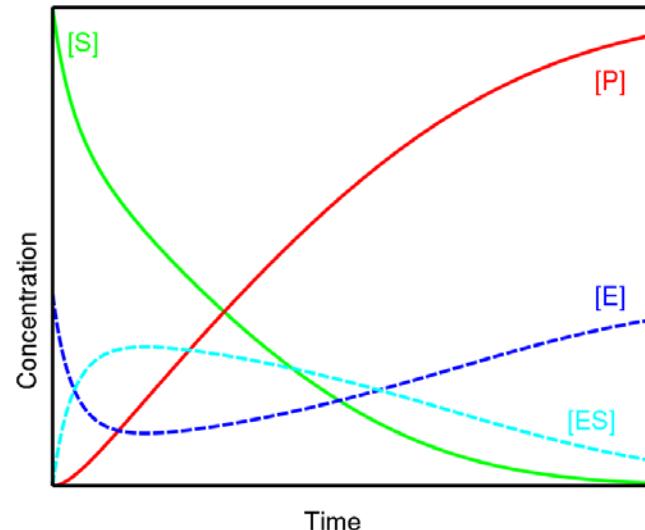
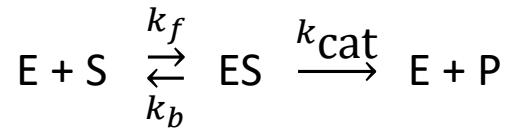
$$[\text{CaCl}_2]' = k_f [\text{Ca}][\text{Cl}]^2 - k_b [\text{CaCl}_2]$$

Enzyme kinetics

It is generally **not** the case that a substrate transforms directly into a product:



Instead, an enzyme is often involved:



Michaelis-Menten

If we can assume either:

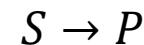
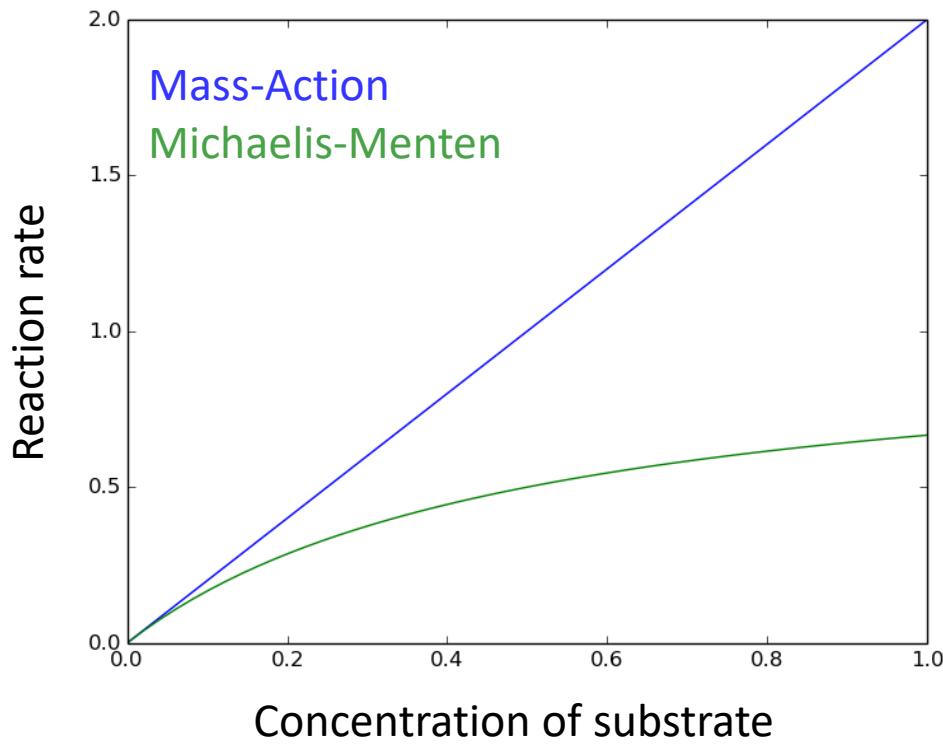
- the substrate (S) and the complex (ES) are in instantaneous equilibrium, or
- the concentration of the complex (ES) does not change on the time-scale of product formation

Then the rate of the enzymatic reaction reduces to:

$$\frac{V_{max} [S]}{K_M + [S]}$$

K_M is called the *Michaelis constant*. It is the concentration at which the reaction proceeds at half its maximum rate.

Michaelis-Menten vs Mass-Action



Both curves on the left have the same rate of reaction when the substrate concentration is low, but the Michaelis-Menten rate levels off (due to limited enzyme availability) as concentrations increase.

$$y = 2x$$

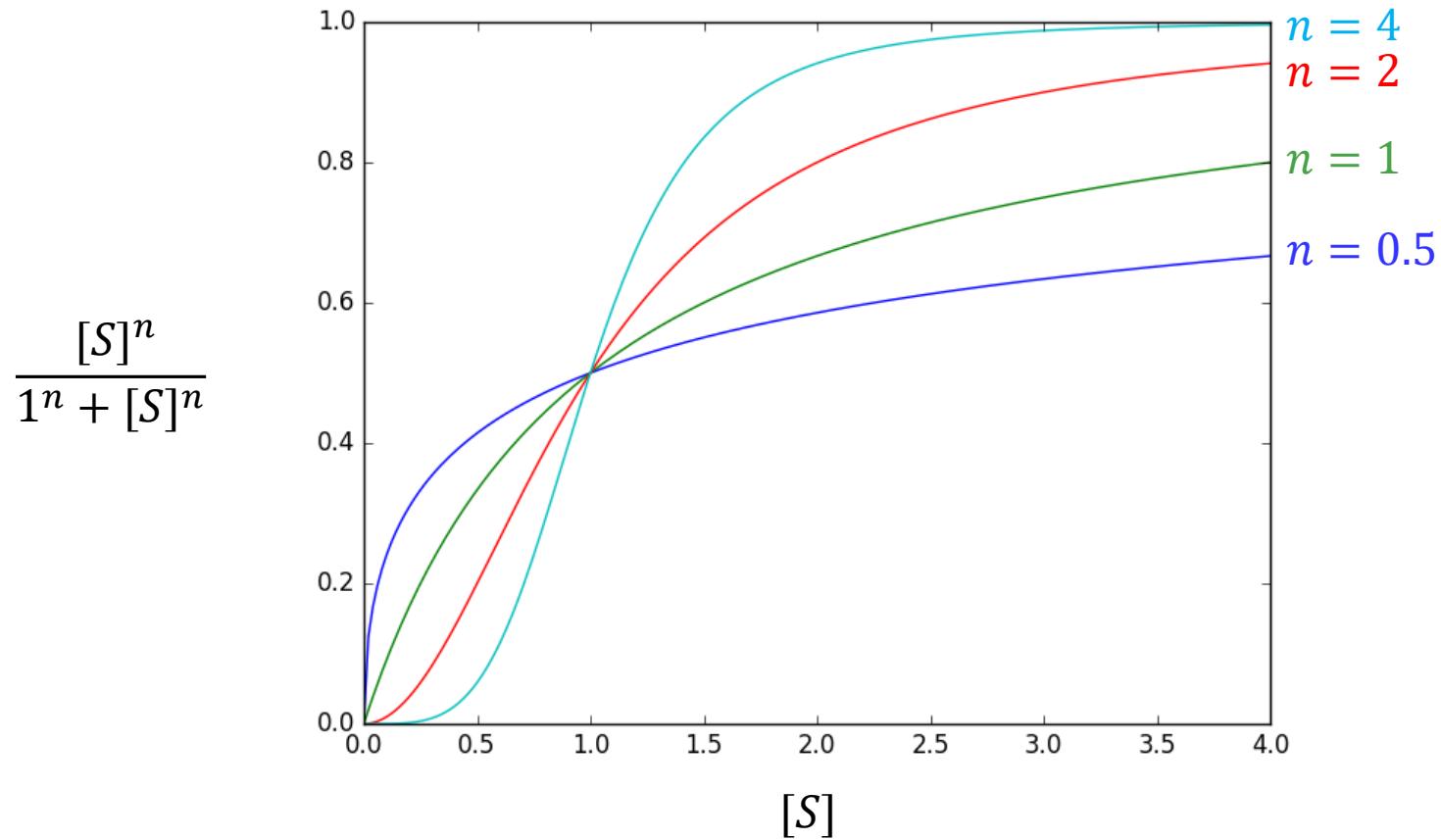
$$y = \frac{x}{x + 0.5}$$

Hill equation: cooperative binding

$$\frac{V_{max} [S]^n}{[k_A]^n + [S]^n}$$

If $n > 1$, positive cooperativity.

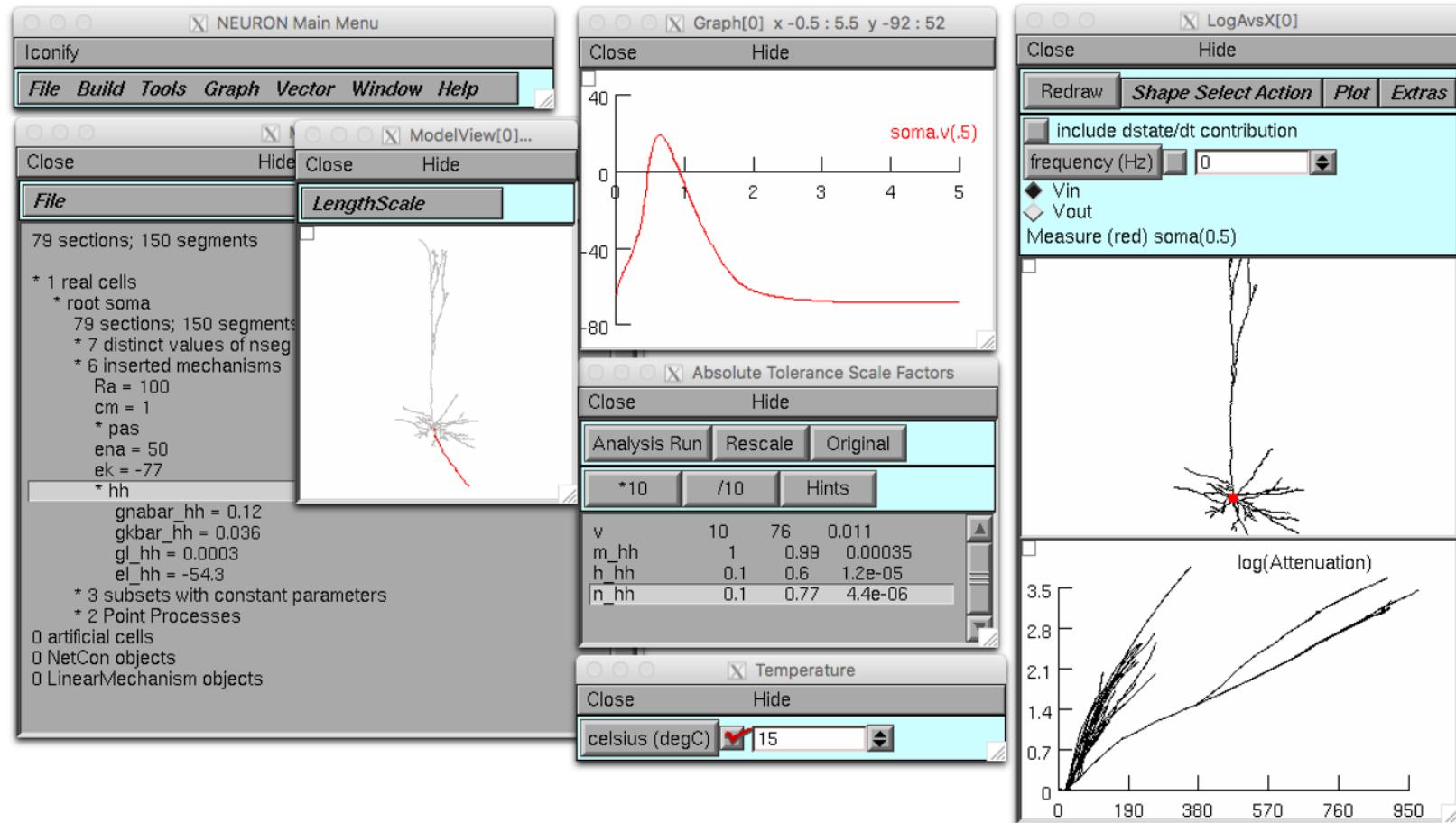
If $n < 1$, negative cooperativity.



NEURON time

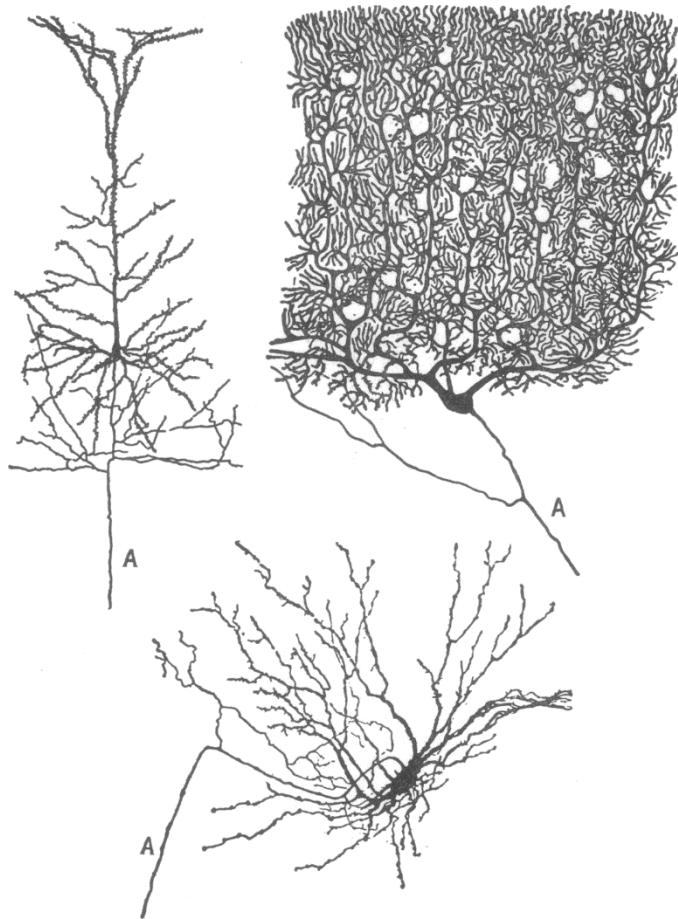


Aside: NEURON GUI tools



The NEURON GUI may be loaded via: [from neuron import gui](#)

Neurons have spatial extent



Cajal 1909 as reproduced in Rall 1962.

Effects of non-point-ness:

- Ion and protein concentrations vary with space.
- Cellular mechanisms (ER, ion channels, etc) vary with space.

Concentrations at different locations affect each other:

- Transport
- Diffusion

Fick's First Law and the diffusion equation

Fick's First Law:

- Diffusive flux is proportional to the concentration gradient.

$$J = -D\nabla\varphi$$

- Here D is called the *diffusion coefficient*.

Fick's Second Law (the diffusion equation):

$$\frac{\partial\varphi}{\partial t} = \nabla \cdot (D\nabla\varphi) = D \nabla^2\varphi$$

where the last equality only holds if D is constant.

Where does diffusion occur?

- Cytosol
 - But not full cross section because of organelles
- Organelles (e.g. ER)
- Extracellular space
 - Tortuosity
 - Anisotropy
 - Volume fraction

NEURON time

- Suppose that a 100 μm long dendrite starts out with 1 mM of IP₃ at positions 40-60 μm , and 0 elsewhere.
- Suppose that IP₃ diffuses in this dendrite at 1 $\mu\text{m}^2/\text{ms}$.
- What does the distribution of IP₃ look like at times 0, 50, 100, 150?

NEURON time

- Suppose that a 100 μm long dendrite starts out with 1 mM of IP₃ at positions 40-60 μm , and 0 elsewhere.
- Suppose that IP₃ diffuses in this dendrite at 1 $\mu\text{m}^2/\text{ms}$.
- What does the distribution of IP₃ look like at times 0, 50, 100, 150?
- At what time does position 70 μm exceed 100 nM? What is its peak value?

Practical limits of pure diffusion

The expected time $E[t]$ for a molecule with diffusion constant D to diffuse a distance x is:

$$E[t] = \frac{x^2}{2D}$$

So in particular, if

$$D = 1 \text{ } \mu\text{m}^2/\text{ms} \text{ and}$$

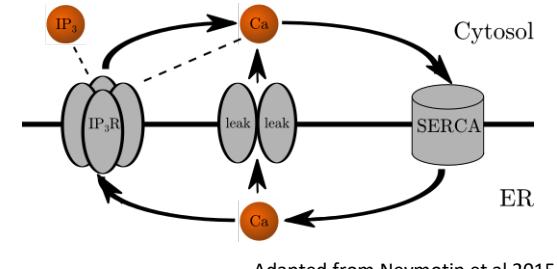
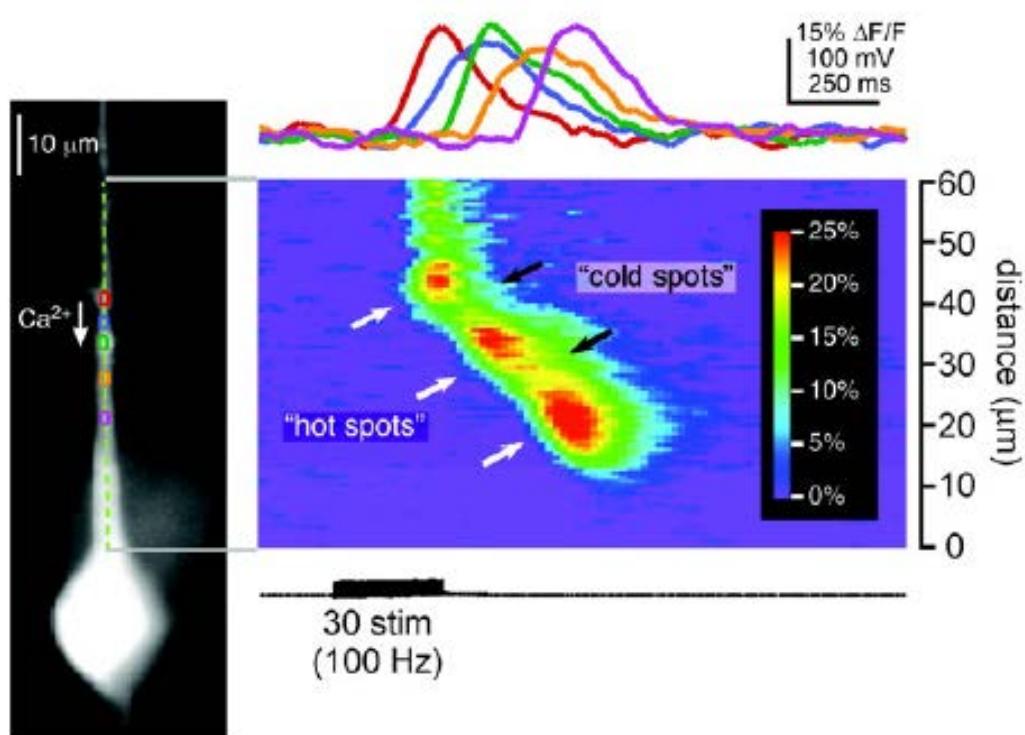
$$x = 100 \text{ } \mu\text{m},$$

Then

$$E[t] = \frac{100^2}{2} = 5000 \text{ ms.}$$

Diffusion with regenerative dynamics can quickly spread signals

A



Adapted from Neymotin et al 2015.

Fitzpatrick et al 2009.

Exercise: regenerative signaling

Action potentials and calcium waves are examples of regenerative signaling in neurons.

For this exercise, we consider the simplest reaction-diffusion kinetics that generate a regenerative signal (think of this as a *very* phenomenological model):

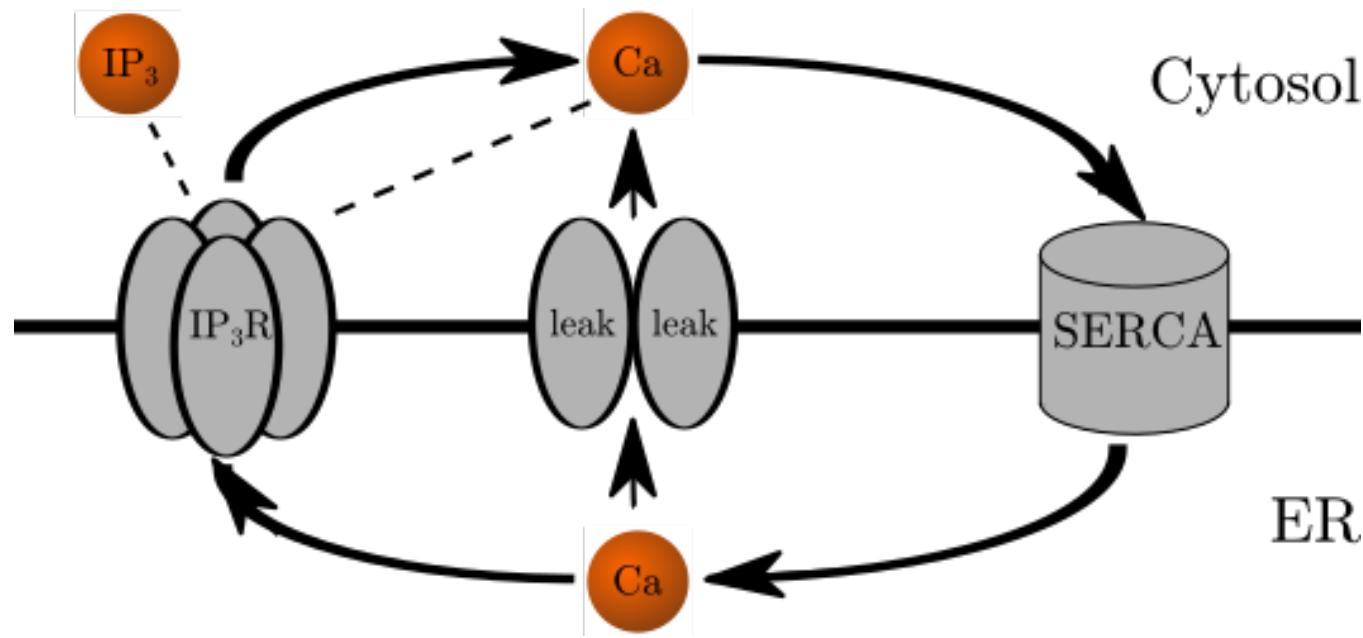
$$\frac{\partial y}{\partial t} = D \nabla^2 y - ky(\alpha - y)(1 - y)$$

Examine this equation and **make a prediction** about the kinetics it describes.

On a 500 micron long piece of dendrite add a species Y changing by the above kinetics (use an `rxd.Rate`). For a first test take $k = 1$, $D = 1$, and $\alpha = 0.3$. Set initial conditions such that Y is 1 where $\text{seg.x} < 0.5$ and 0 elsewhere. Plot the concentration of Y vs position at several time points to show the development of a traveling wave front.

Run the same simulation with $\alpha = 0.7$. What changes? More generally, how does α affect the wave?

Calcium wave model



Calcium wave model

$$\frac{\partial \text{Ca}_{\text{cyt}}^{2+}}{\partial t} = d_{\text{Ca}_{\text{cyt}}^{2+}} \cdot \Delta \text{Ca}_{\text{cyt}}^{2+} + \frac{J_{\text{IP3R}} - J_{\text{SERCA}} + J_{\text{leakER}}}{f_{\text{cyt}}} + c_{\text{ionic}},$$

$$\frac{\partial \text{Ca}_{\text{ER}}^{2+}}{\partial t} = d_{\text{Ca}_{\text{ER}}^{2+}} \cdot \Delta \text{Ca}_{\text{ER}}^{2+} - \frac{J_{\text{IP3R}} - J_{\text{SERCA}} + J_{\text{leakER}}}{f_{\text{ER}}},$$

$$\frac{\partial \text{IP}_3}{\partial t} = d_{\text{IP}_3} \cdot \Delta \text{IP}_3,$$

$$J_{\text{IP3R}} = \bar{p}_{\text{IP}_3\text{R}} \cdot m_{\text{IP}_3\text{R}}^3 \cdot n_{\text{IP}_3\text{R}}^3 \cdot h_{\text{IP}_3\text{R}}^3 \cdot (\text{Ca}_{\text{ER}}^{2+} - \text{Ca}_{\text{cyt}}^{2+}) / \Xi,$$

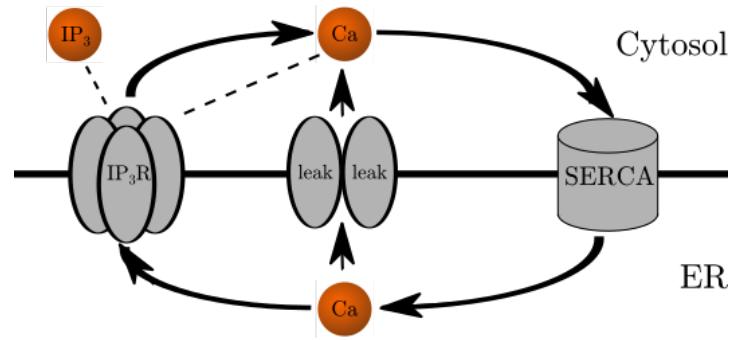
$$J_{\text{SERCA}} = -\frac{\bar{p}_{\text{serca}} \cdot \text{Ca}_{\text{cyt}}^{2+^2}}{(k_{\text{serca}}^2 + \text{Ca}_{\text{cyt}}^{2+^2}) \cdot \Xi},$$

$$J_{\text{leakER}} = \bar{p}_{\text{leakER}} \cdot (\text{Ca}_{\text{ER}}^{2+} - \text{Ca}_{\text{cyt}}^{2+}) / \Xi.$$

$$\frac{dh}{dt} = \frac{h_\infty - h}{\tau_h}$$

$$h_\infty = \frac{1}{1 + 1000 \frac{\text{Ca}_c}{0.3}}$$

$$m = \frac{\text{IP3}}{\text{IP3} + k_{ip3}} \frac{1000 \text{Ca}_c}{1000 \text{Ca}_c + k_{act}}$$



Adapted from Neymotin, McDougal et al 2015.

Calcium wave model: cawave.py

Where do the dynamics occur?

Calcium wave model: cawave.py

Who are the actors? (ions, proteins, states, etc)

```
ca = rxd.Species([cyt, er], d=caDiff, name='ca' ,  
                  charge=2, initial=cac_init)  
  
ip3 = rxd.Species(cyt, d=ip3Diff,  
                   initial=ip3_init)  
  
ip3r_gate_state = rxd.State(cyt_er_membrane ,  
                             initial=0.8)  
  
h_gate = ip3r_gate_state[cyt_er_membrane]
```

Calcium wave model: cawave.py

What are they doing?

```
serca = rxd.MultiCompartmentReaction(ca[cyt], ca[er],
    gserca / ((kserca / (1000. * ca[cyt])) ** 2 + 1),
    membrane=cyt_er_membrane, mass_action=False)

leak = rxd.MultiCompartmentReaction(ca[er], ca[cyt],
    gleak, gleak, membrane=cyt_er_membrane)

minf = ip3[cyt] * 1000. * ca[cyt] / (ip3[cyt] + kip3)
      / (1000. * ca[cyt] + kact)

k = gip3r * (minf * h_gate) ** 3

ip3r = rxd.MultiCompartmentReaction(ca[er], ca[cyt], k,
    membrane=cyt_er_membrane)

ip3rg = rxd.Rate(h_gate, (1. / (1 + 1000. * ca[cyt]
    / (0.3)) - h_gate) / ip3rtau)
```

Reaction-diffusion in NEURON

Why use NEURON's rxd module?

Reduces typing

- In 2 lines: declare a domain, then declare a molecule, allowing it to diffuse and respond to flux from ion channels.

```
all = rxd.Region(h.allsec(), nrn_region='i')
ca = rxd.Species(all, name='ca', d= 1, charge= 2)
```
- Reduces the risk for errors from typos or misunderstandings.

Allows arbitrary domains

NEURON traditionally only identified concentrations just inside and just outside the plasma membrane. The rxd module allows you to declare your own regions of interest (e.g. ER, mitochondria, etc).

rxd module overview

terministic vs stochastic.



Declare a region: rxd.Region

Basic Usage

```
cyt = rxd.Region(seclist)
```

seclist may be any iterable of sections; e.g. a SectionList or a Python list.

Identify with a standard region

```
cyt = rxd.Region(seclist, nrn_region='i')
```

nrn_region may be i or o, corresponding to the locations of e.g. nai vs nao.

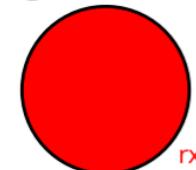
Specify the cross-sectional shape

```
cyt = rxd.Region(seclist, geometry=rxd.Shell(0.5, 1))
```

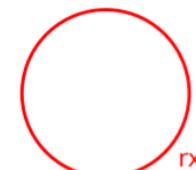
The default geometry is rxd.inside.

The geometry and nrn_region arguments may both be specified.

geometry:



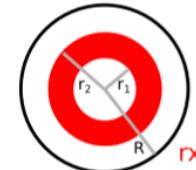
rxd.inside



rxd.membrane



rxd.FractionalVolume(
volume_fraction=f₁,
surface_fraction=f₂)



rxd.Shell(r₁/R, r₂/R)

Adapted from:
McDougal et al 2013.

rxd.Region tips

Specify nrn_region if concentrations interact with NMODL

If NMODL mechanisms (ion channels, point processes, etc) depend on or affect the concentration of a species living in a given region, that region must declare a nrn_region (typically 'i').

To declare a region that exists on all sections

```
r = rxd.Region(h.allsec())
```

Use list comprehensions to select sections

```
r = rxd.Region([sec for sec in h.allsec() if 'apical' in sec.name()])
```

Declare ions & proteins: rxd.Species

Basic usage

```
protein = rxd.Species(region, d=16)
```

d is the **diffusion constant** in $\mu\text{m}^2/\text{ms}$. region is an rxd.Region or an iterable of rxd.Region objects.

Initial conditions

```
protein = rxd.Species(region, initial=value)
```

value is in mM. It may be a constant or a function of the node.

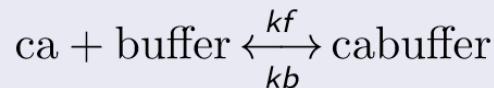
Connecting with HOC

```
ca = rxd.Species(region, name='ca', charge=2)
```

If the nrn_region of region is "i", the concentrations of this species will be stored in cai, and its concentrations will be affected by ica.

Specifying dynamics: rxd.Reaction

Mass-action kinetics

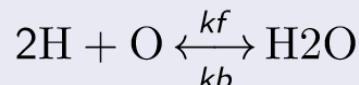


buffering = rxd.Reaction(ca + buffer, cabuffer, kf, kb)

kf is the forward reaction rate, kb is the backward reaction rate. kb may be omitted if the reaction is unidirectional.

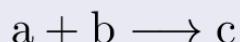
In a mass-action reaction, the reaction rate is proportional to the product of the concentrations of the reactants.

Repeated reactants



water_reaction = rxd.Reaction(2 * H + O, H2O, kf, kb)

Arbitrary reaction formula, e.g. Hill dynamics



hill_reaction = rxd.Reaction(a + b, c, a ^ 2 / (a ^ 2 + k ^ 2), mass_action=False)

Hill dynamics are often used to model cooperative reactions.

rxd.Rate and rxd.MultiCompartmentReaction

rxd.Rate

Use rxd.Rate to specify an explicit contribution to the rate of change of some concentration or state variable.

$$\text{ip3degradation} = \text{rxd.Rate(ip3, -k * ip3)}$$

rxd.MultiCompartmentReaction

Use rxd.MultiCompartmentReaction when the dynamics span multiple regions; e.g. a pump or channel.

$$\text{ip3r} = \text{rxd.MultiCompartmentReaction(ca[er], ca[cyt], kf, kb, membrane=cyt_er_membrane)}$$

The rate of these dynamics is proportional to the membrane area.

Manipulating nodes

Getting a list of nodes

- `nodelist = protein.nodes`

Filtering a list of nodes

- `nodelist2 = nodelist(region)`
- `nodelist2 = nodelist(0.5)`
- `nodelist2 = nodelist(section)(region)(0.5)`

Other operations

- `nodelist.concentration = value`
- `values = nodelist.concentration`
- `surface_areas = nodelist.surface_area`
- `volumes = nodelist.volume`
- `node = nodelist[0]`

Concentration pointers

To get a pointer to a concentration, use `node._ref_concentration`:

Recording traces

```
v = h.Vector()  
v.record(ca.nodes[0]._ref_concentration)
```

Plotting

```
g = h.Graph()  
g.addvar('ca[er][dend](0.5)', ca.nodes(er)(dend)(0.5)[0]._ref_concentration)  
h.graphList[0].append(g)
```