Principles of neuromodelling

Computational Methods for Translational Neuroscience

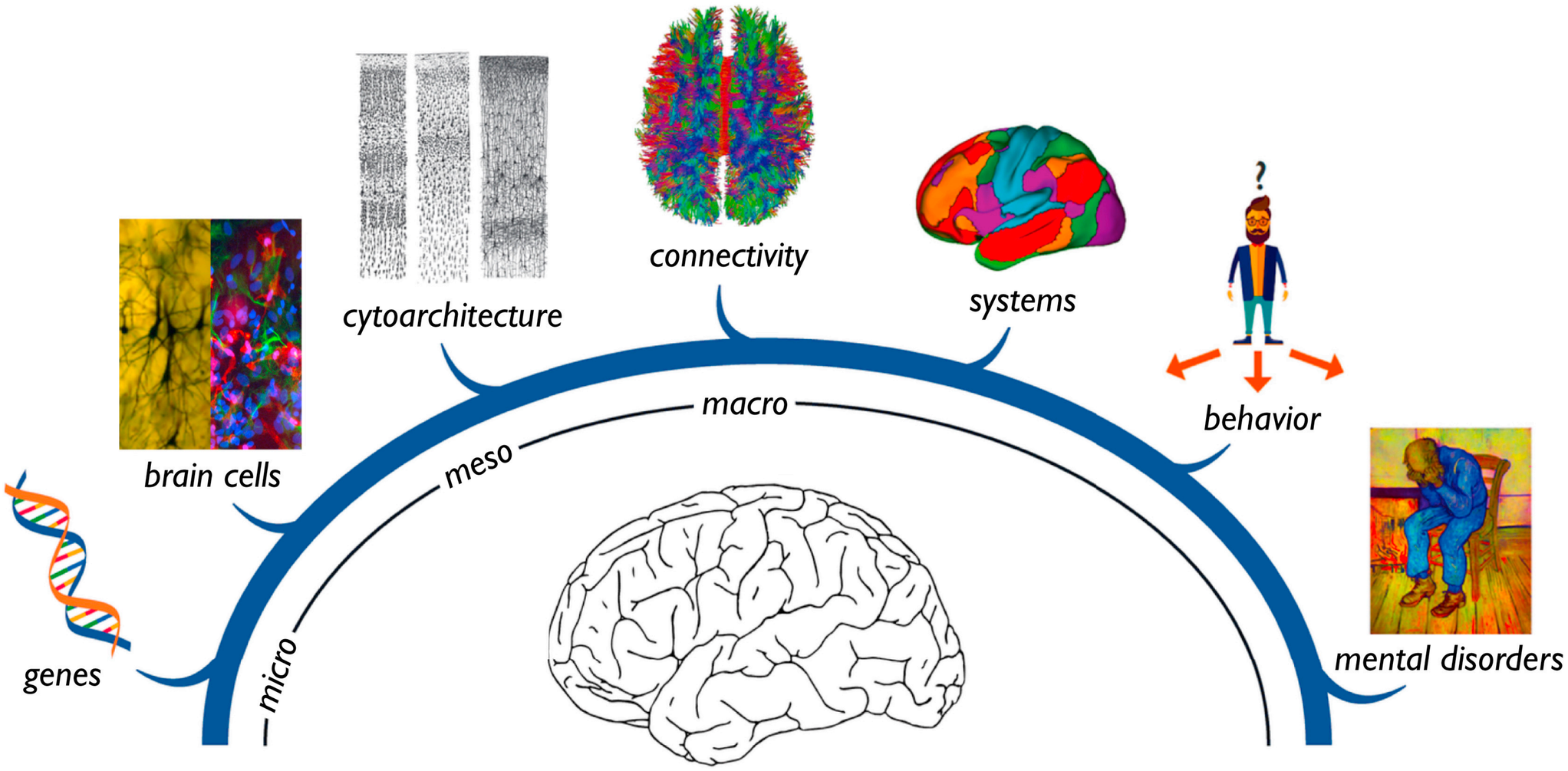
Sadra Sadeh (s.sadeh@imperial.ac.uk)

23/11/2022

Introduction

modelling the brain as a complex multi-scale system

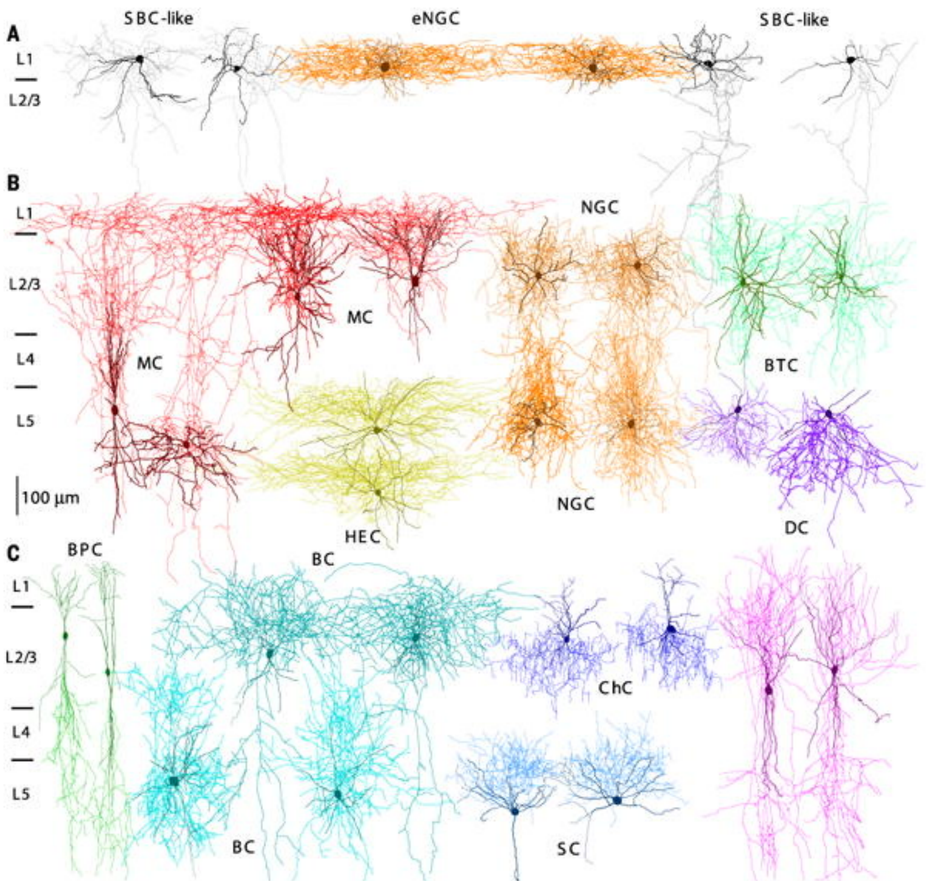
Brain as a multi-scale complex system

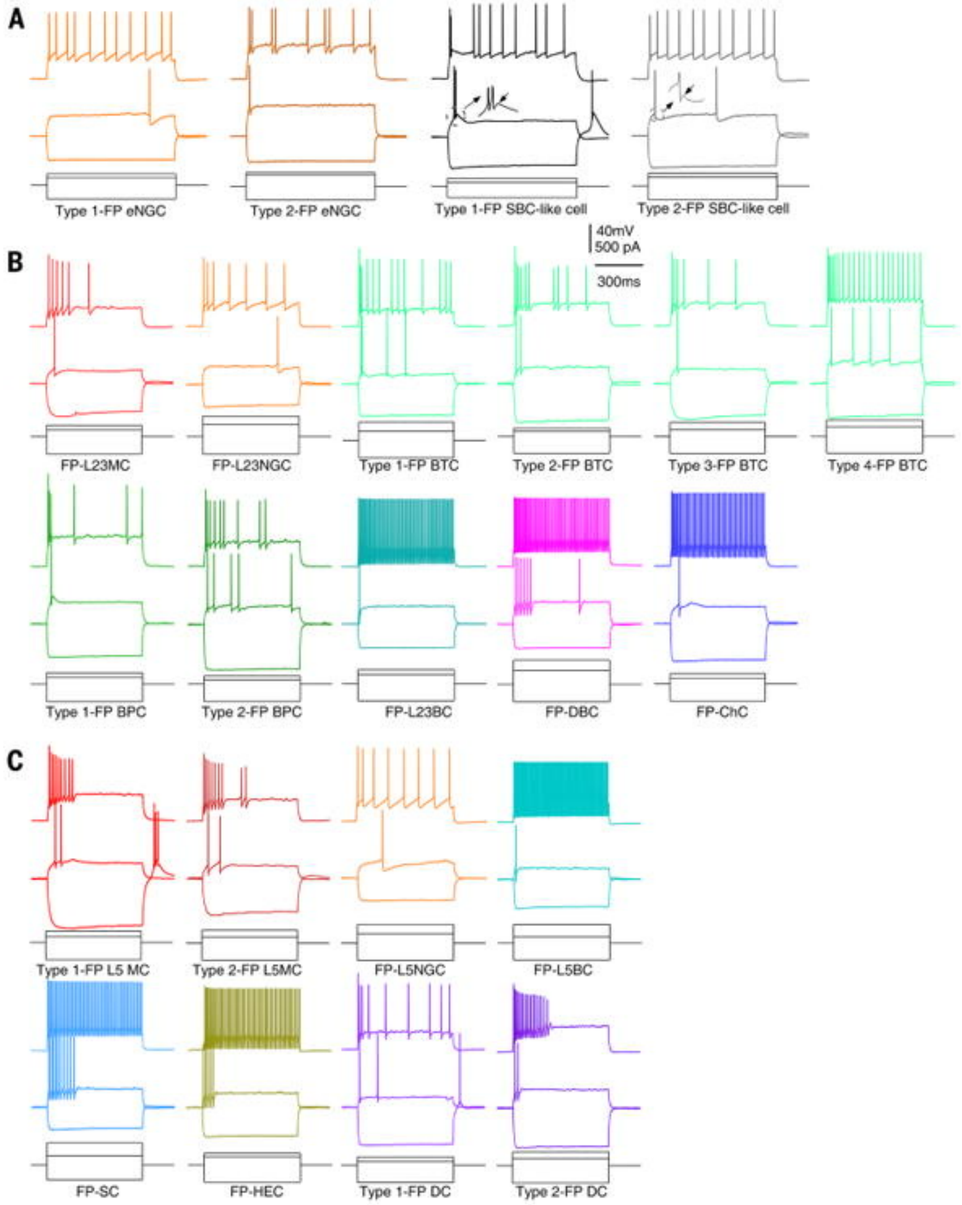


van den Heuvel et al (Biol. Psychiatry 2019)

Structural complexity

diversity of neuronal properties

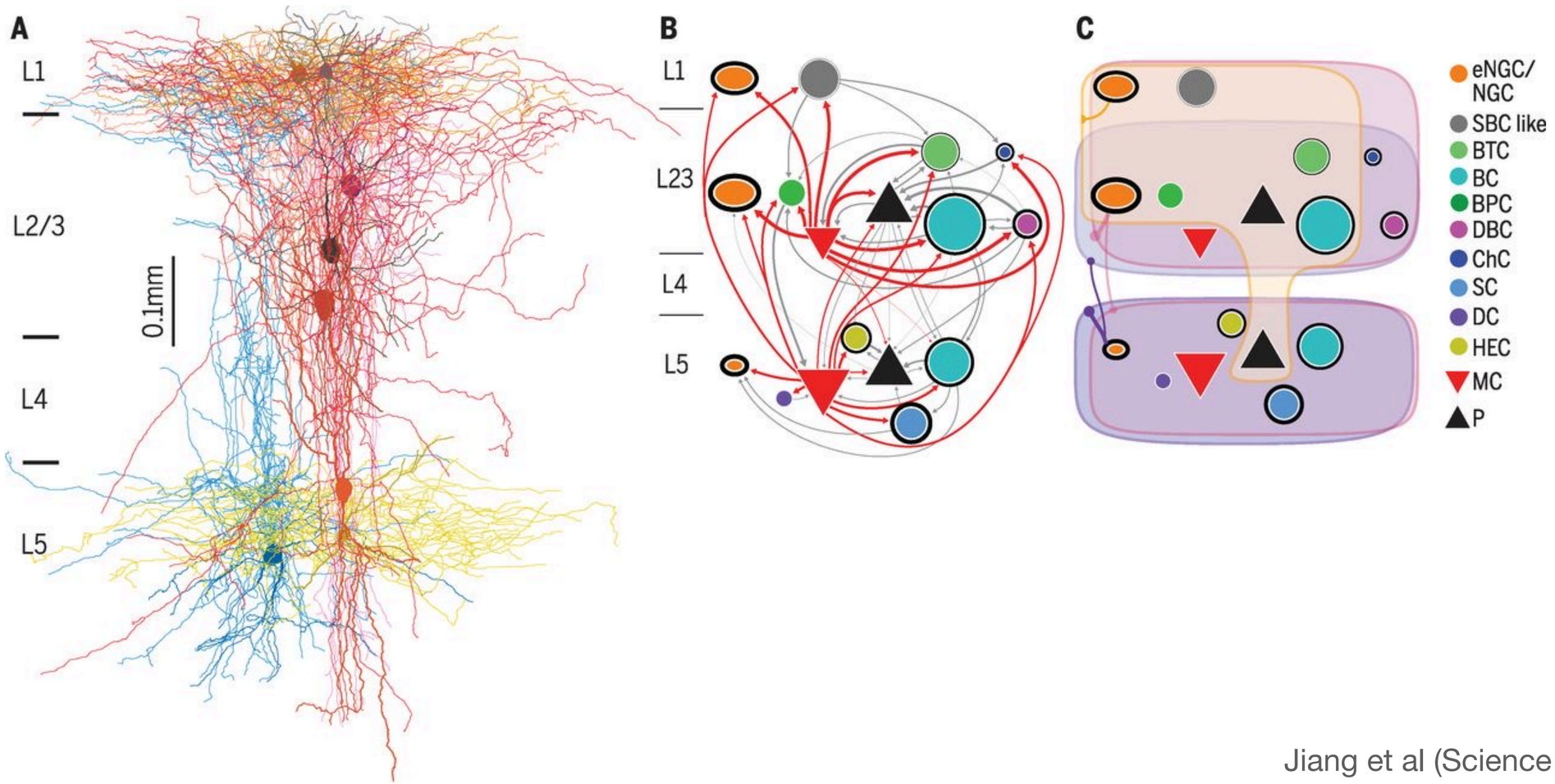




Jiang et al (Science 2015)

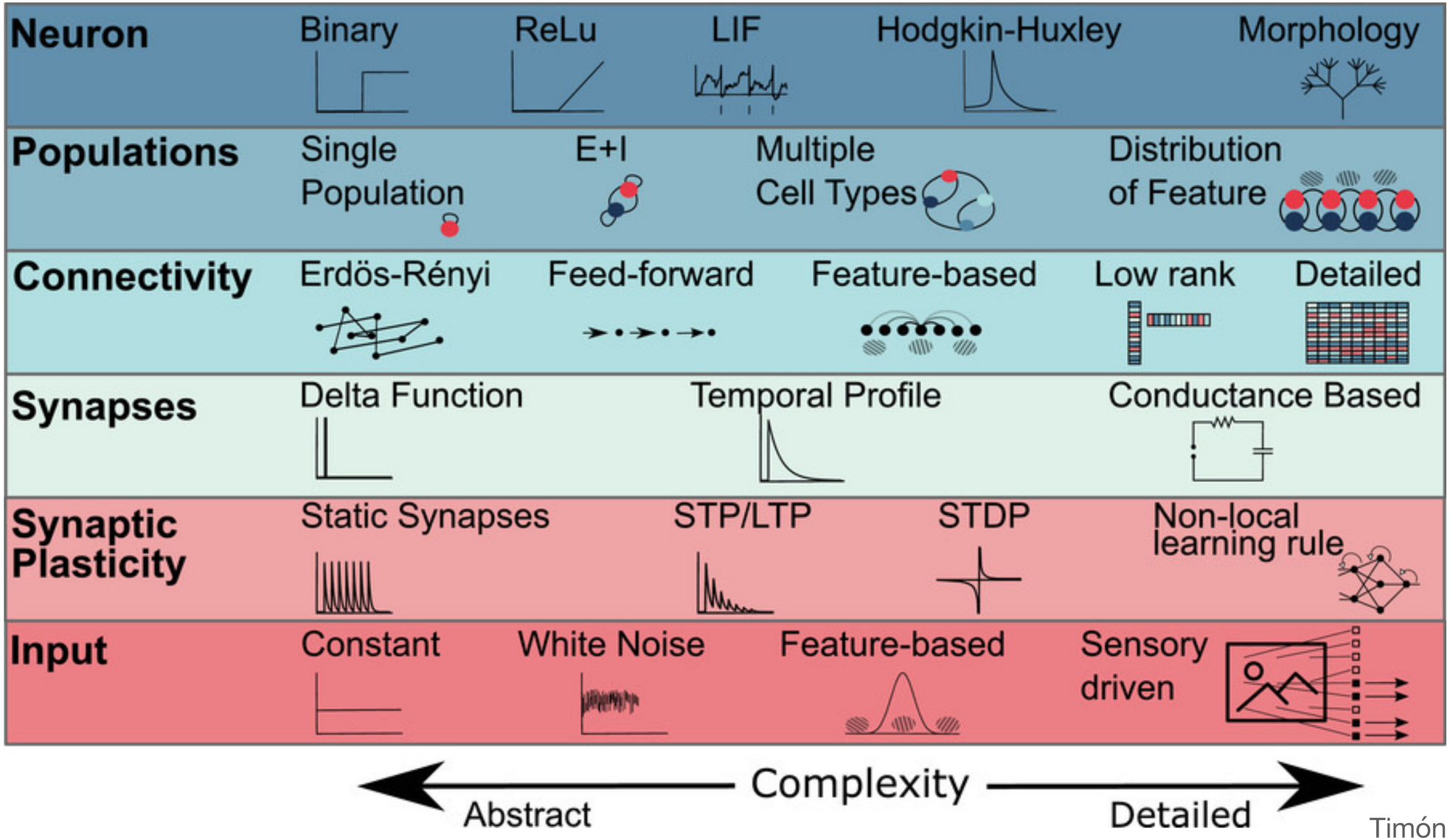
Structural complexity

complex and specific connectivity



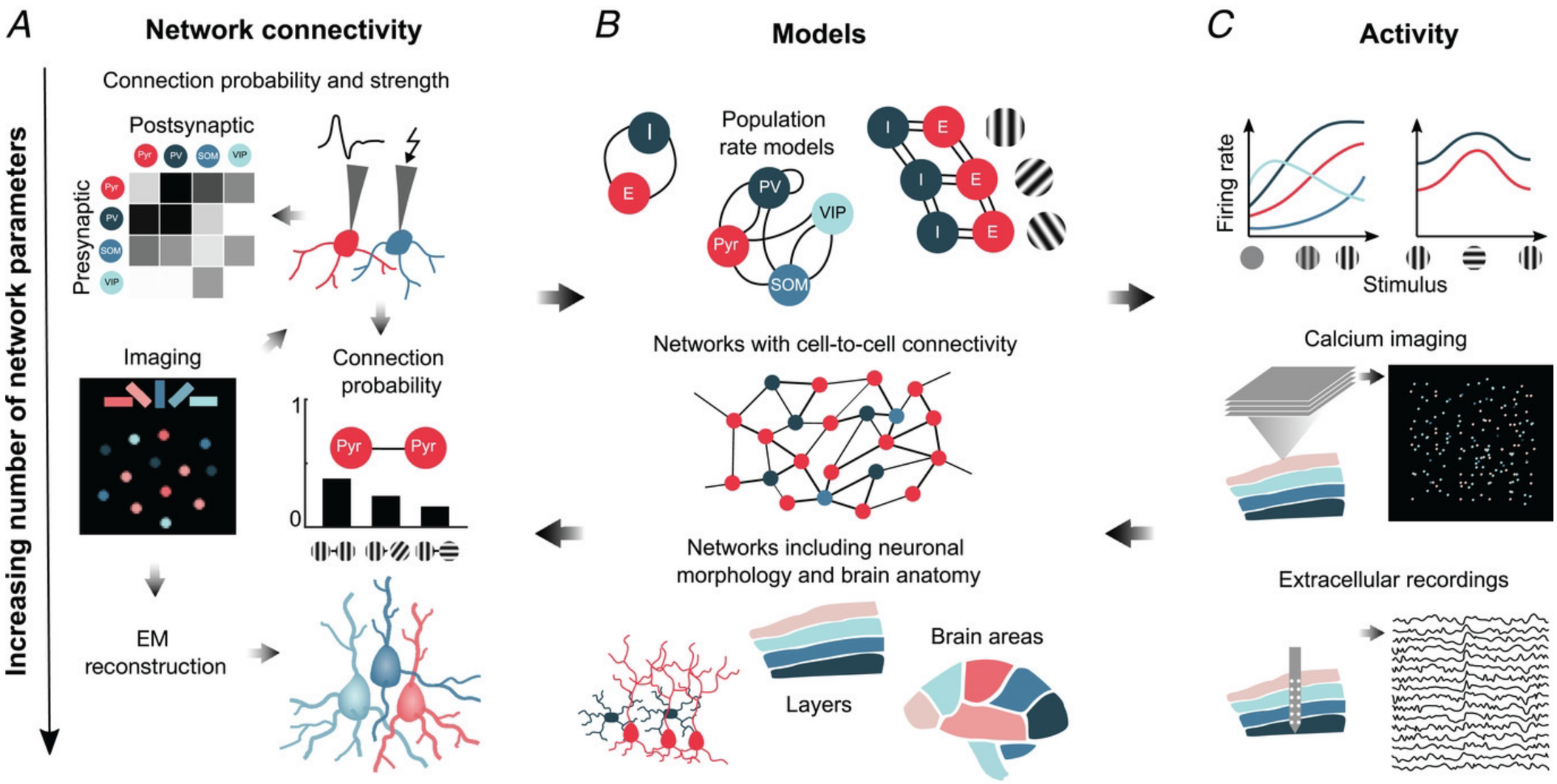
Jiang et al (Science 2015)

Models with different levels of complexity



Timón et al (J. Physiol. 2022)

Models as bridges at different levels

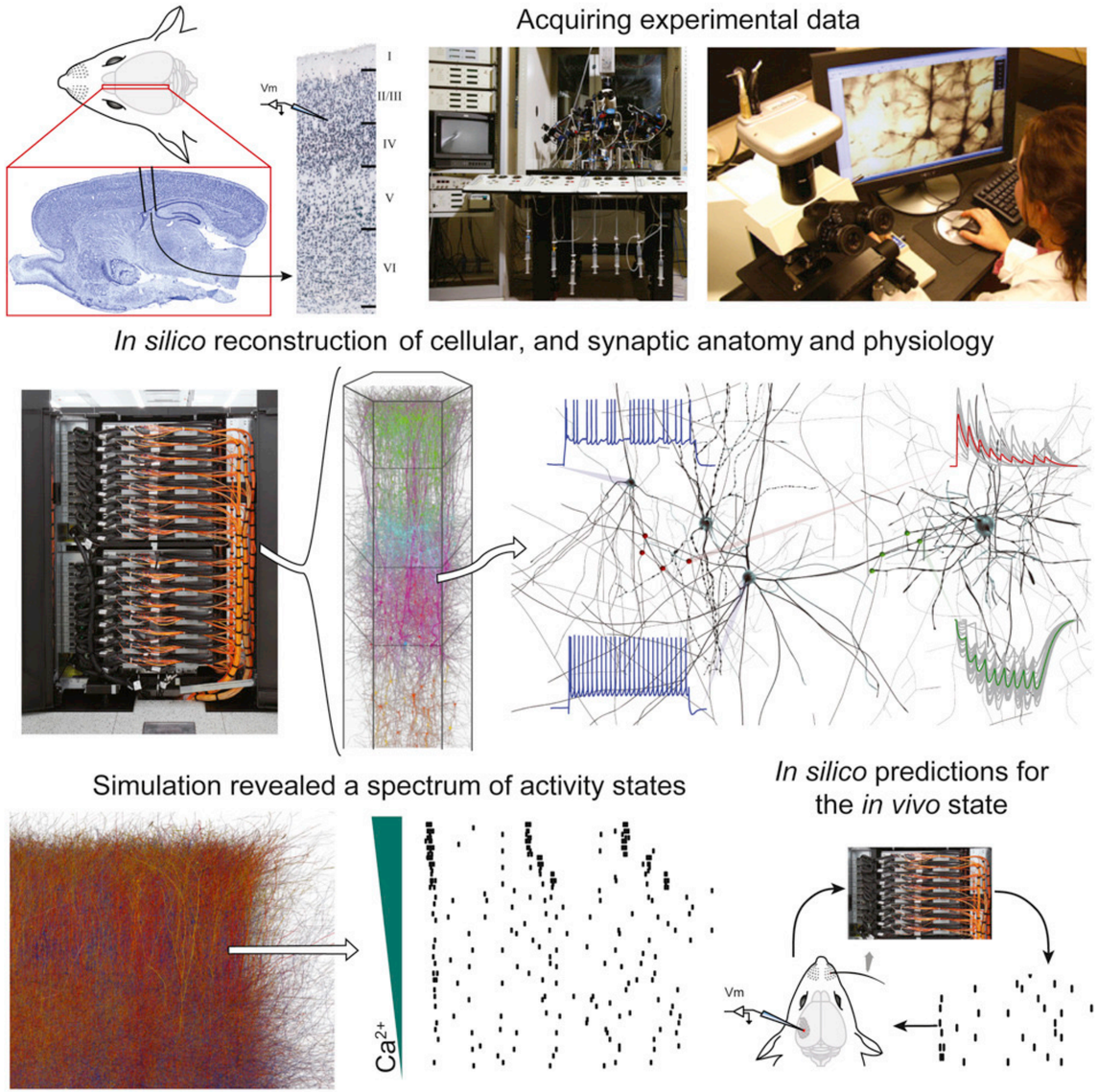


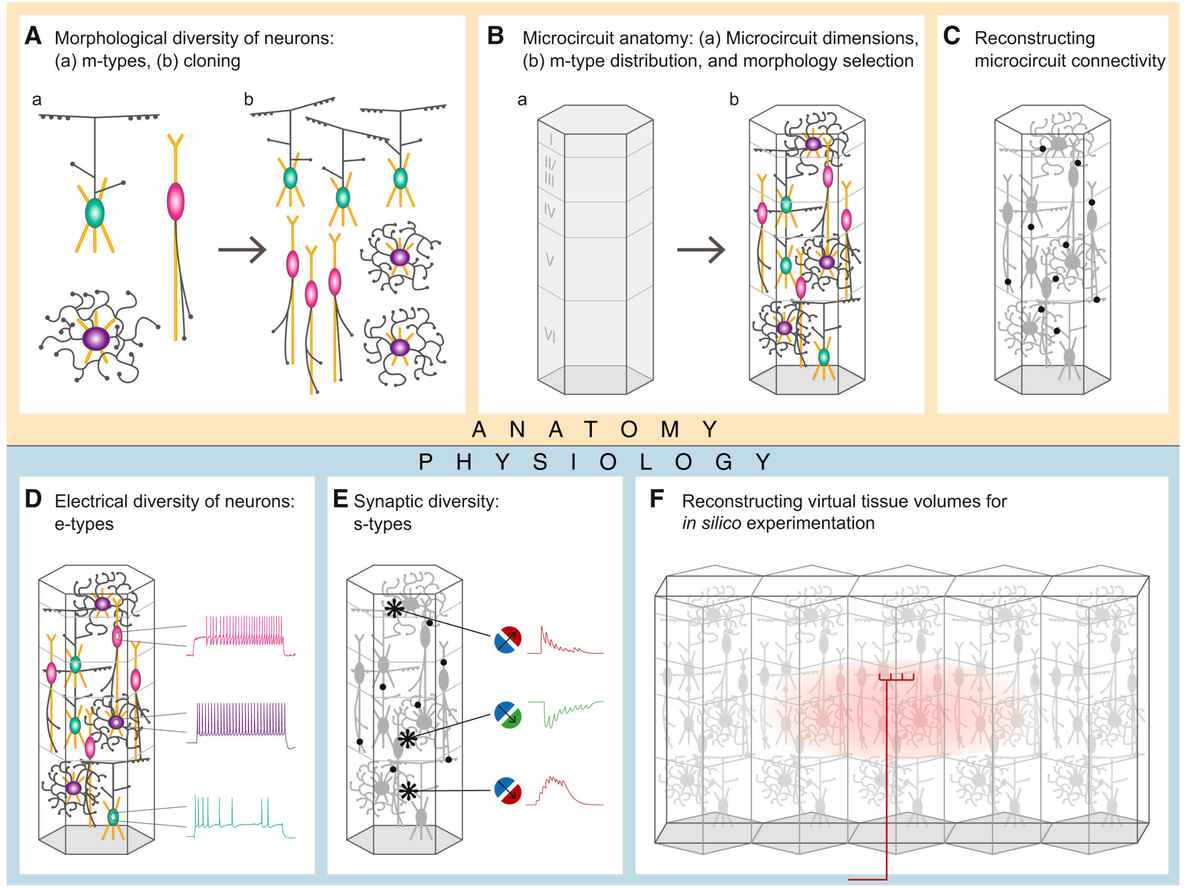
Timón et al (J. Physiol. 2022)

Where is the right level?

- Computational models try to be as “realistic” as possible
- But how much detail do we need / can we put in our models?
- It is not possible – at least yet – to have a “full” brain simulation
- Human Brain Project, Allen Institute, etc
- We still can’t build a computer model of a cubic millimetre of mouse cortex

Detailed computational models

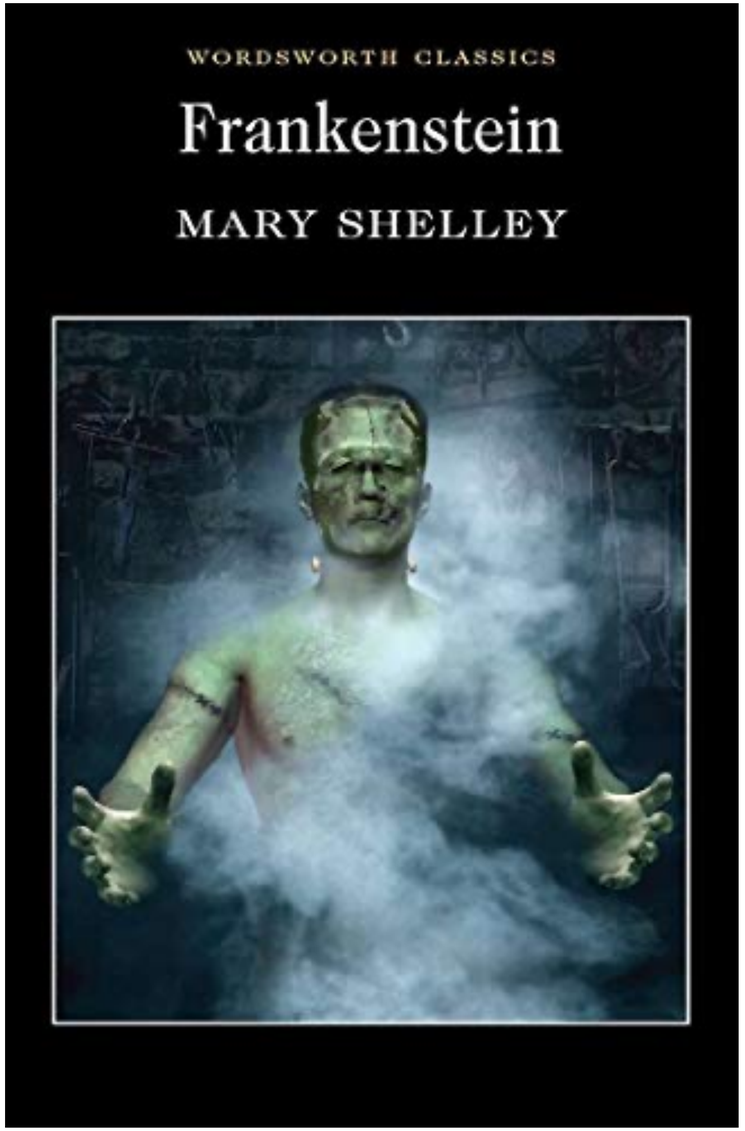




Markram et al. (Cell 2015)

Good computational model?

- ‘Enough’ detail, but not too much
  - *A good computational model is not always the most complex*
- We can’t reproduce a real brain – yet!
  - *Frankenstein*
- What is a good model aeroplane?
  - *The most complex one or a simple one that flies?*



Bottom up versus top-down approaches

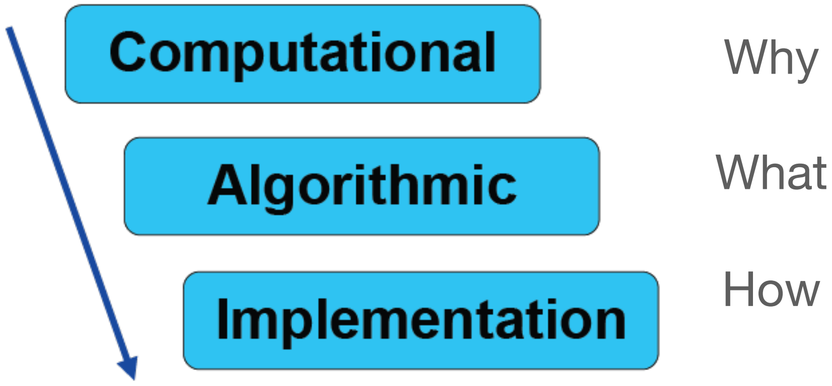
Bottom-up approach:

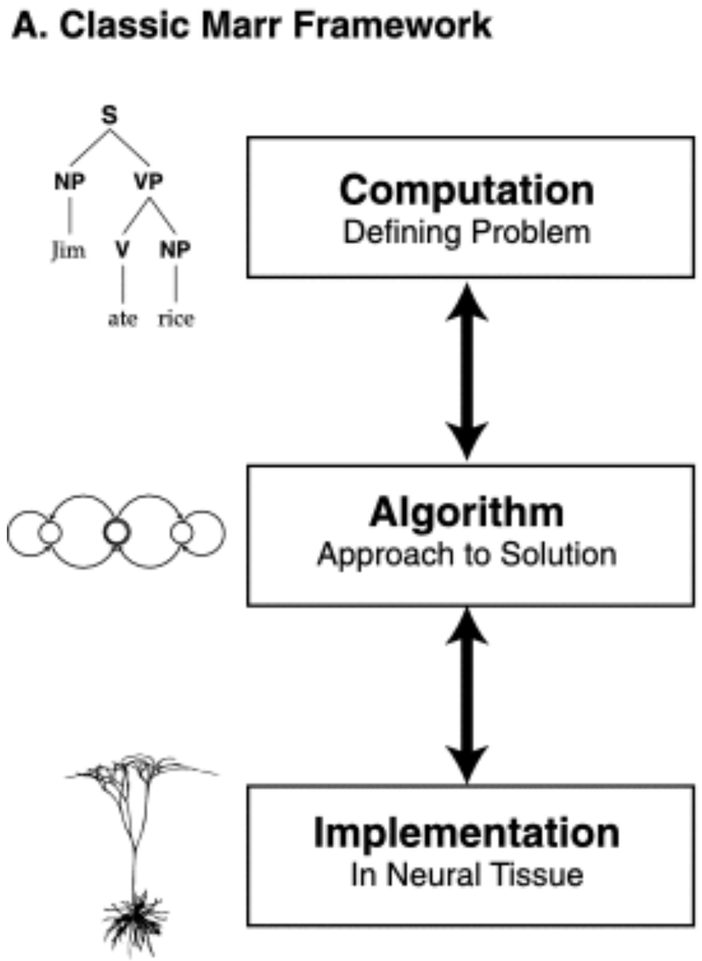
- Trying to build a detailed model, then seeing what it does
- Understanding the system by constraining the model
- A constrained model cannot do everything

Top-down approach:

- How neurons are implementing/enabling a specific function/behaviour?
- Which type of receptive fields do you need to perform a visual or navigation task?

Different levels of analysis/modelling





Tecumseh Fitch (2014)

Summary

- Building computational models can shed light on the **mechanistic** operation of the brain: function and dysfunction; health and disease

- But the brain is complex – structural, functional, regions, scales (spatiotemporal):

  - we do not know all the parameters (yet and maybe never) --> so we need to make **approximations** and educated guesses

  - the art of finding the right ones to **solve the puzzle** at hand

- A good model provides a good quantitative *and* qualitative understanding of the experimental data/question

  - …but a better model would go beyond that and "**predicts**"

- Different levels of analysis – and hence modelling

Modelling neurons

Modelling neuronal networks

Modelling learning and plasticity

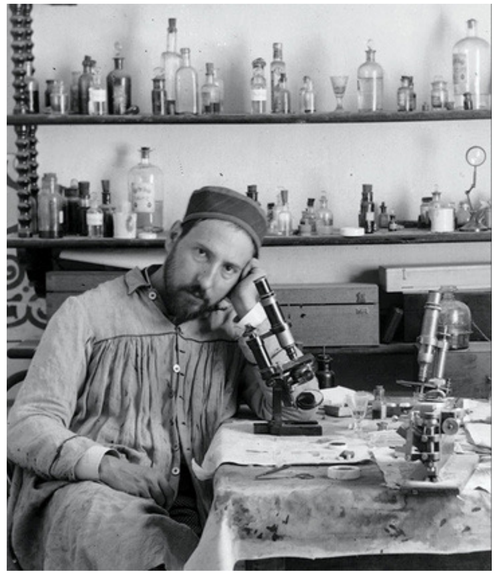
Modelling neurons

Modelling neuronal networks

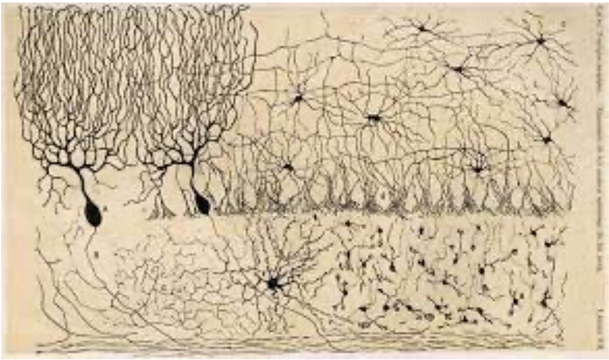
Modelling learning and plasticity

Where to start to model the brain?

Its main “elements”: Neurons – might be a good start



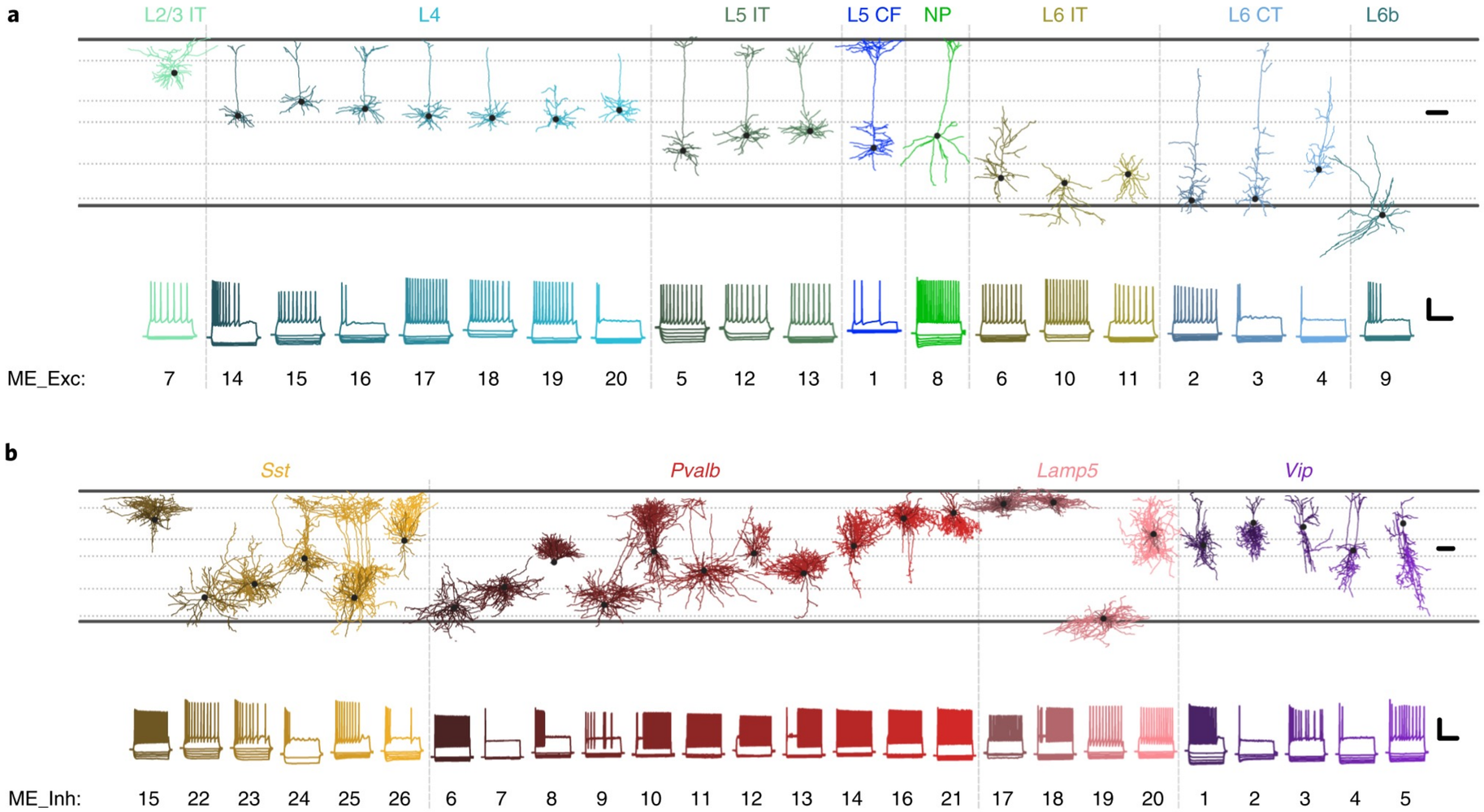
Cajal's Butterflies of the Soul: Science and Art
Javier DeFelipe (Oxford Univ Press)



“Butterflies of the soul” according to Santiago Ramón y Cajal

Neurons can have dramatically different activity patterns

Defined transcriptomic subclasses linked to electrical and morphological properties



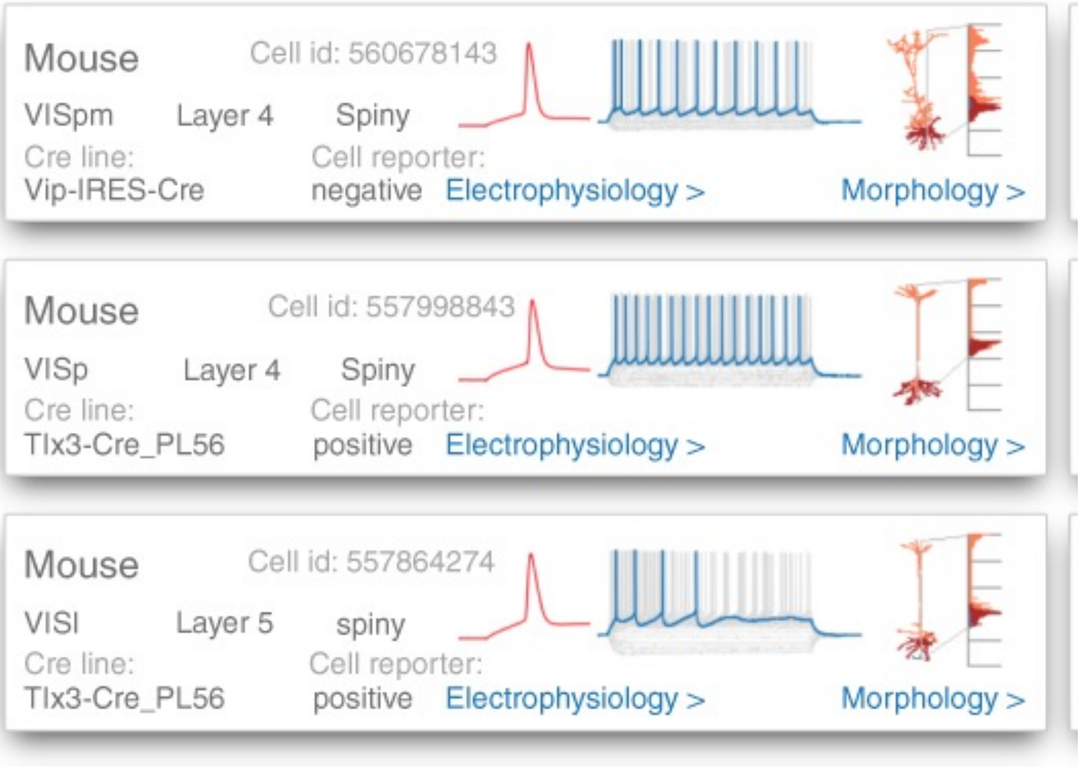
Classification of electrophysiological and morphological neuron types in the mouse visual cortex
Gouwens … Koch (Nature Neuroscience 2019)

Determinants of neuronal identity and activity

Neuronal features:

1. Genetic markers (transcriptomics)
2. Morphology (neuroanatomy)
3. **Electrophysiology (Biophysics)**

Shape + type and distribution of ion channels:
Response patterns of neurons

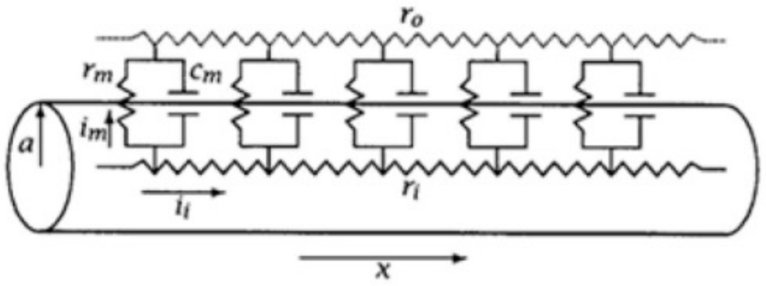


Modelling neurons

- Cable theory
- Neuron active properties
- Hodgkin-Huxley equations

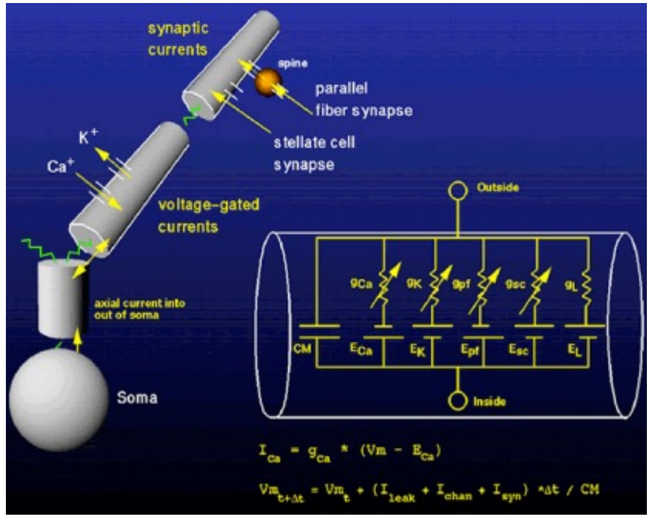
Reduced models:

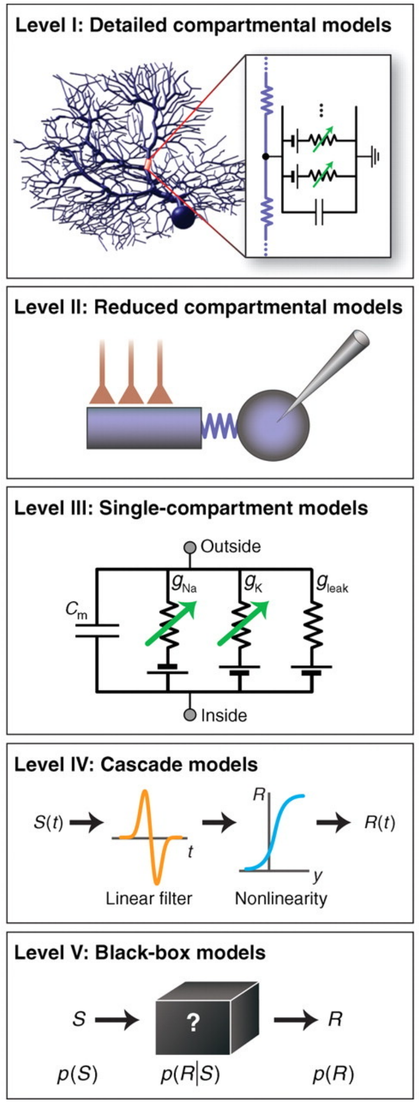
- Morris-Lecar model
- **Leaky integrate-and-fire neurons**
- Izhikevich model
- …



passive properties of neurons as "stretched-out" RC circuits

Compartmental modelling

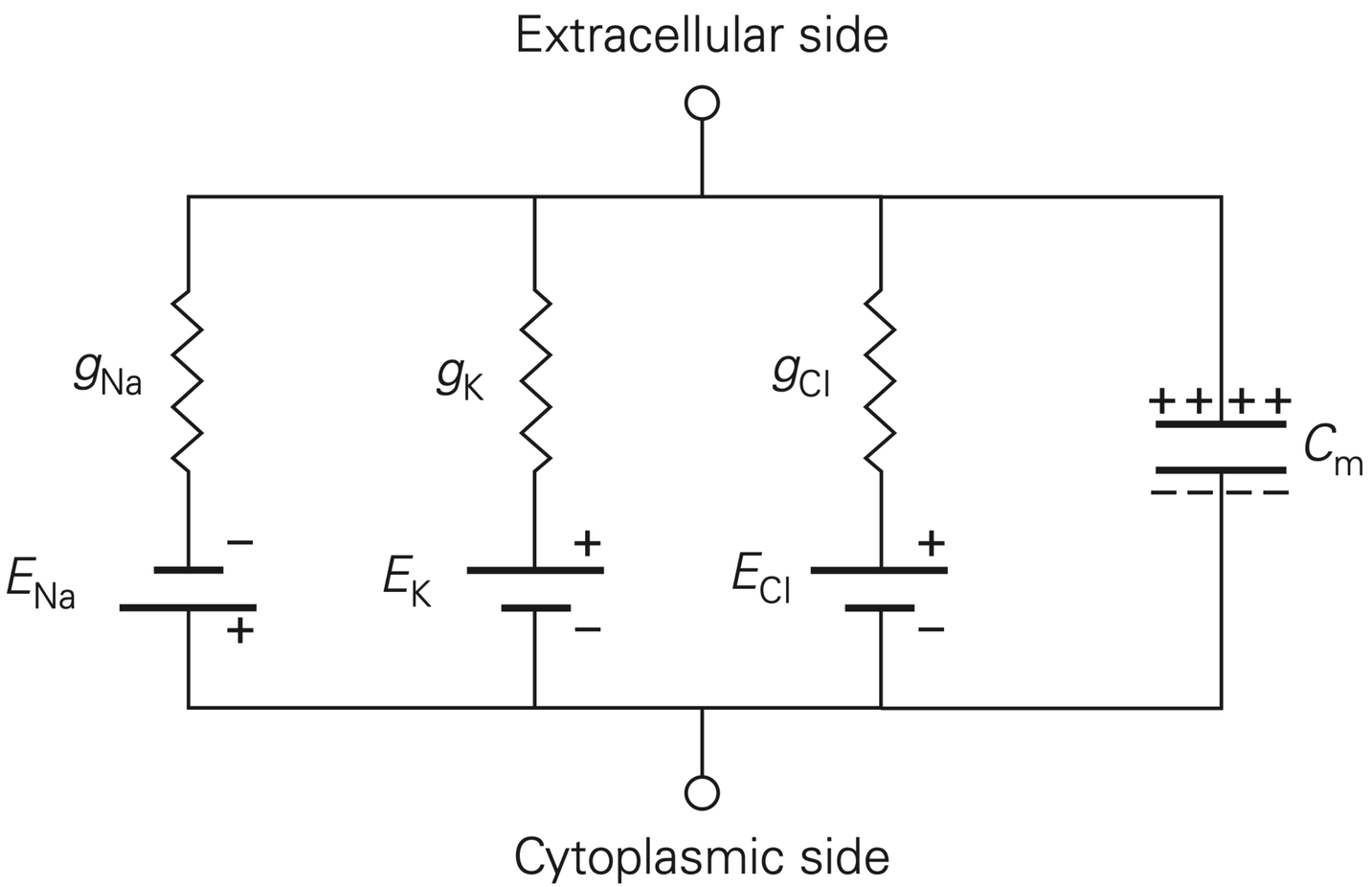




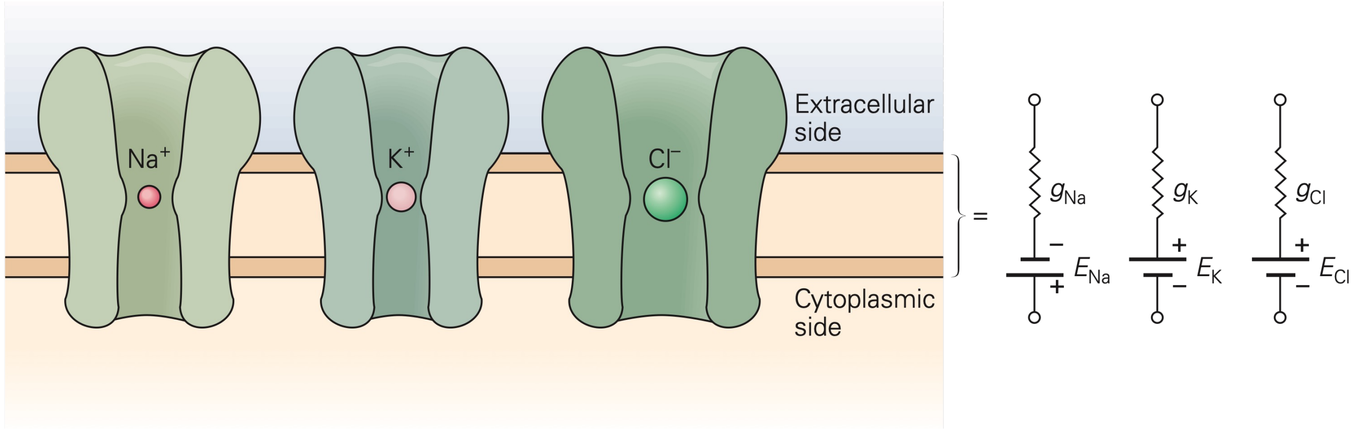
(Herz et al., Science 2006)

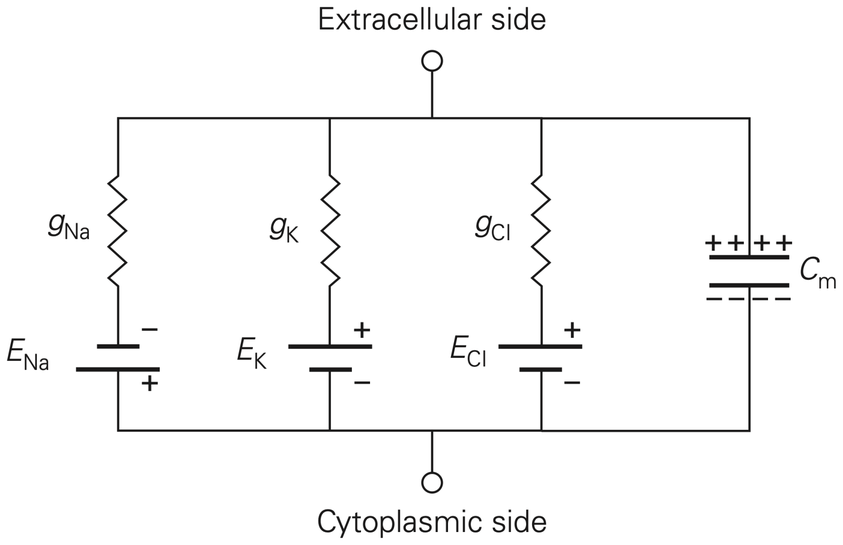
Modeling Single-Neuron Dynamics and Computations: A Balance of Detail and Abstraction

A circuit model of neuronal membrane



Differential equation describing the membrane potential evolution





$$i_c = -C\frac{dV}{dt}$$

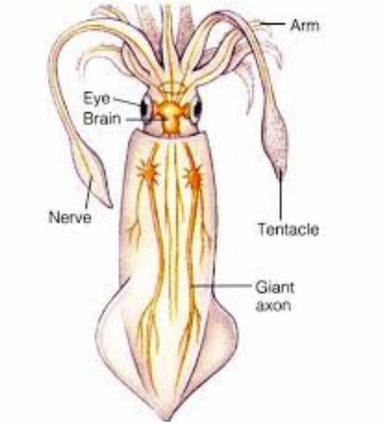
$$i_c = i_K + i_{Na} + i_{Cl}$$

$$i_C = g_K(V - E_K) + g_{Na}(V - E_{Na}) + g_{Cl}(V - E_{Cl})$$

$$\boxed{C\frac{dV}{dt} = g_K(E_K - V) + g_{Na}(E_{Na} - V) + g_{Cl}(E_{Cl} - V)}$$

The differential equation describing the dynamics of the membrane potential

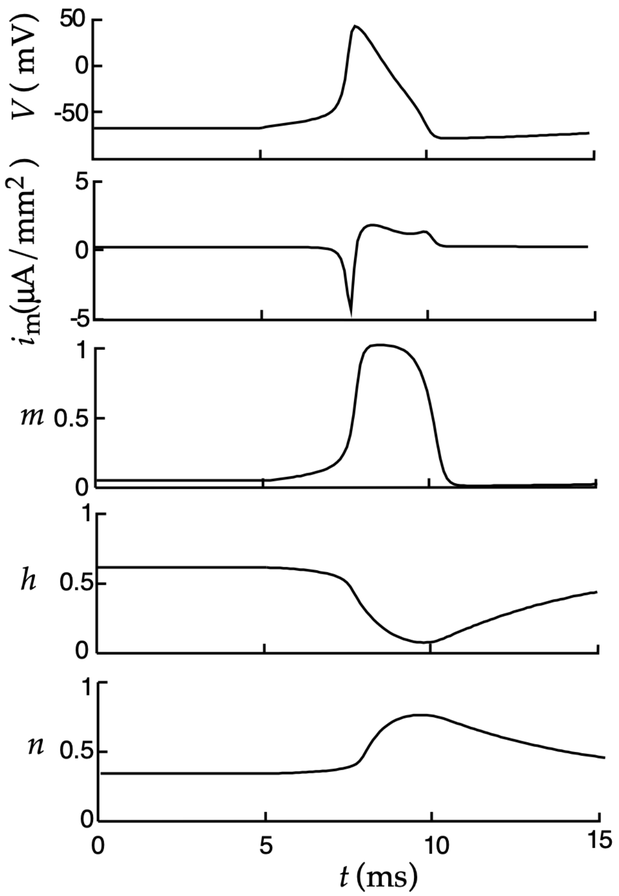
Hodgkin-Huxley model of an action potential



$$I = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + g_K(V_m - V_K) + g_{Na}(V_m - V_{Na}) + g_l(V_m - V_l)$$

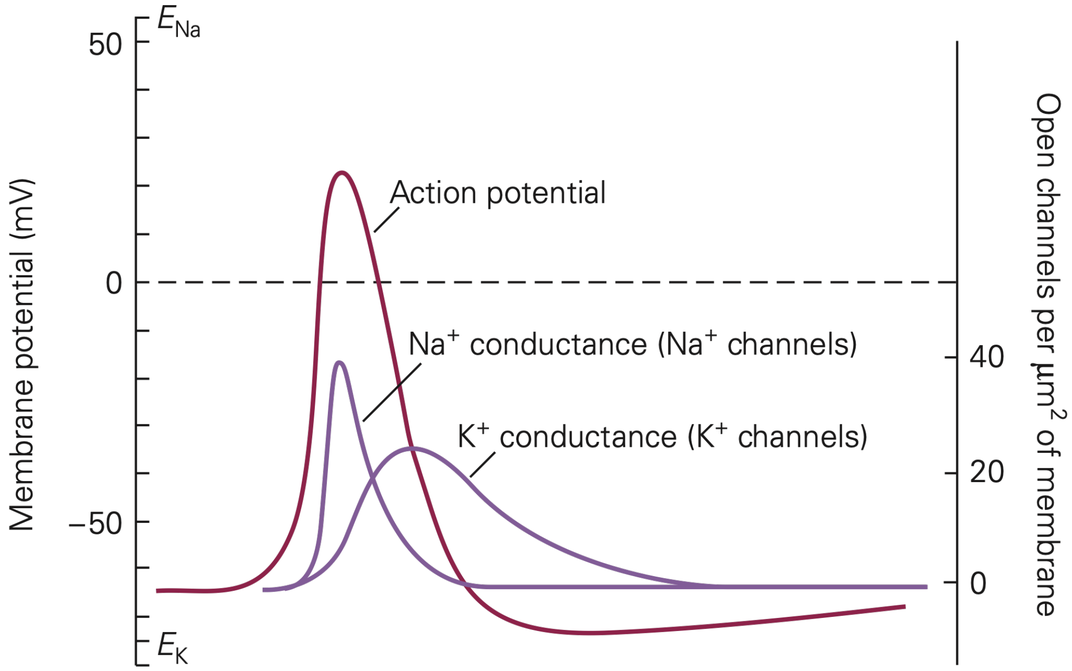
The time dependent elements of this equation are Vm, gNa, and gK, where the last two conductances depend explicitly on voltage as well.

$$i_\mathrm{m} = \overline{g}_\mathrm{L}(V - E_\mathrm{L}) + \overline{g}_\mathrm{K} n^4 (V - E_\mathrm{K}) + \overline{g}_\mathrm{Na} m^3 h (V - E_\mathrm{Na})$$



Hodgkin and Huxley (1952)

Interplay of Sodium and Potassium conductances during an action potential



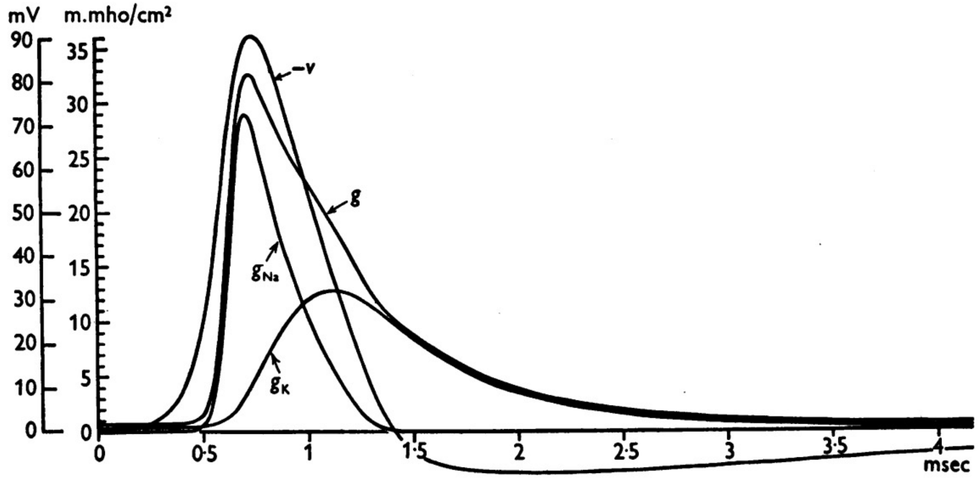


Fig. 17. Numerical solution of eqn. (31) showing components of membrane conductance (g) during propagated action potential ($-V$). Details of the analysis are as in Fig. 15.

Hodgkin and Huxley (1952)

Hodgkin-Huxley equations

$$i_{\mathrm{m}} = \overline{g}_{\mathrm{L}}(V - E_{\mathrm{L}}) + \overline{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}}) + \overline{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}})$$

4-dimensional differential equation:

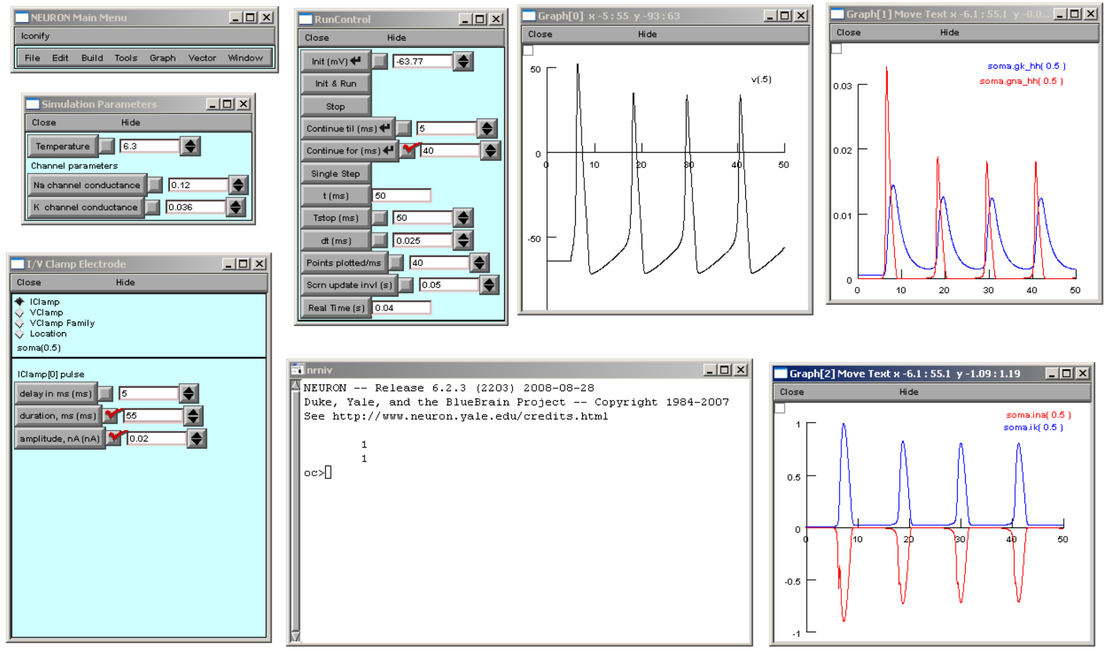
$$I = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + \bar{g}_{\mathrm{K}} n^4 (V_m - V_K) + \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l),$$

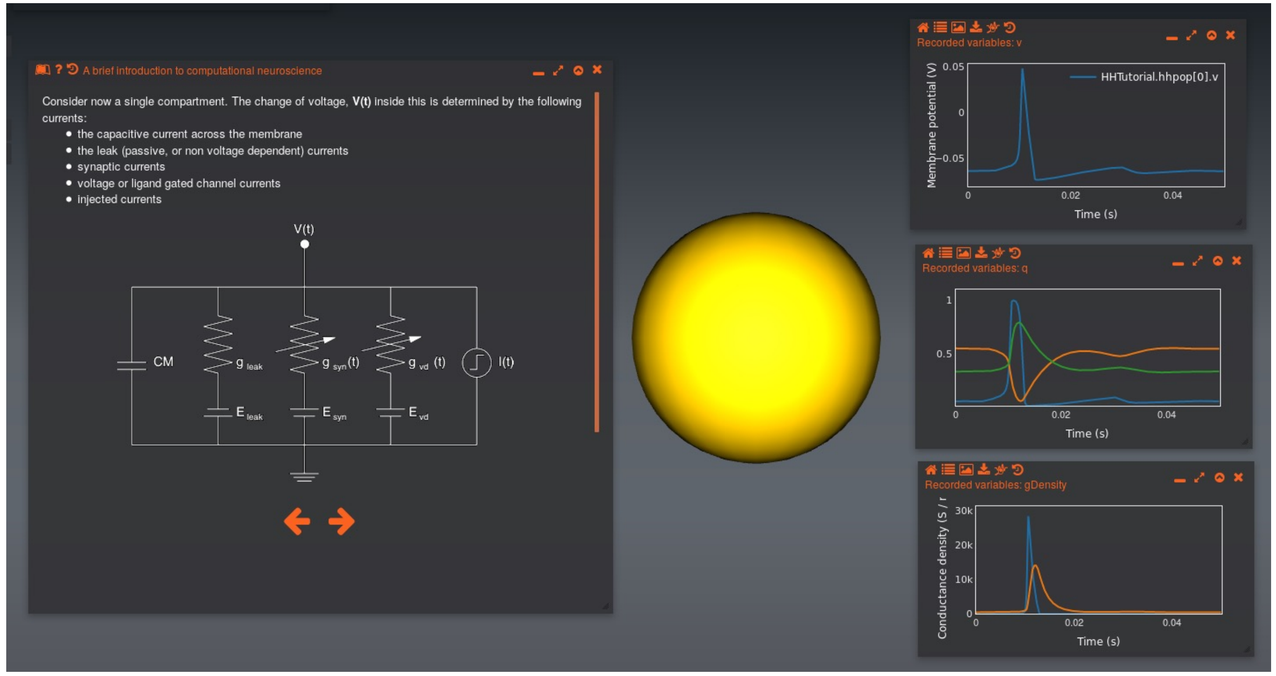
$$\frac{dn}{dt} = \alpha_n(V_m)(1 - n) - \beta_n(V_m) n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1 - m) - \beta_m(V_m) m$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1 - h) - \beta_h(V_m) h$$

Simulation



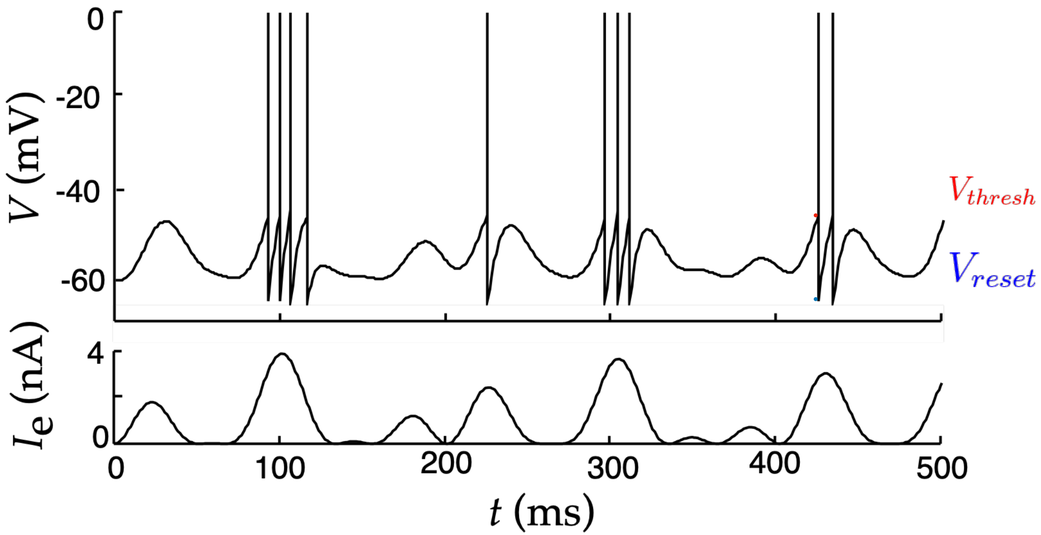


NEURON simulator
(https://www.neuron.yale.edu/neuron/)

Open Source Brain
(https://www.opensourcebrain.org/projects/hodgkin-huxley-tutorial)

Reduced model: Leaky integrate and fire

$$\tau \frac{dV}{dt} = (E_{rest} - V) + R_m I_{ext}$$



Solving differential equations

```
dX = f(X)

X(n+1) = X(n) + dX(n)

X(n+1) = X(n) + f(X(n))

for i in range(N_steps):
    X[i+1] = X[i] + f(X([i])
```

$$\tau \frac{dV}{dt} = (E_{rest} - V) + R_m I_{ext}$$

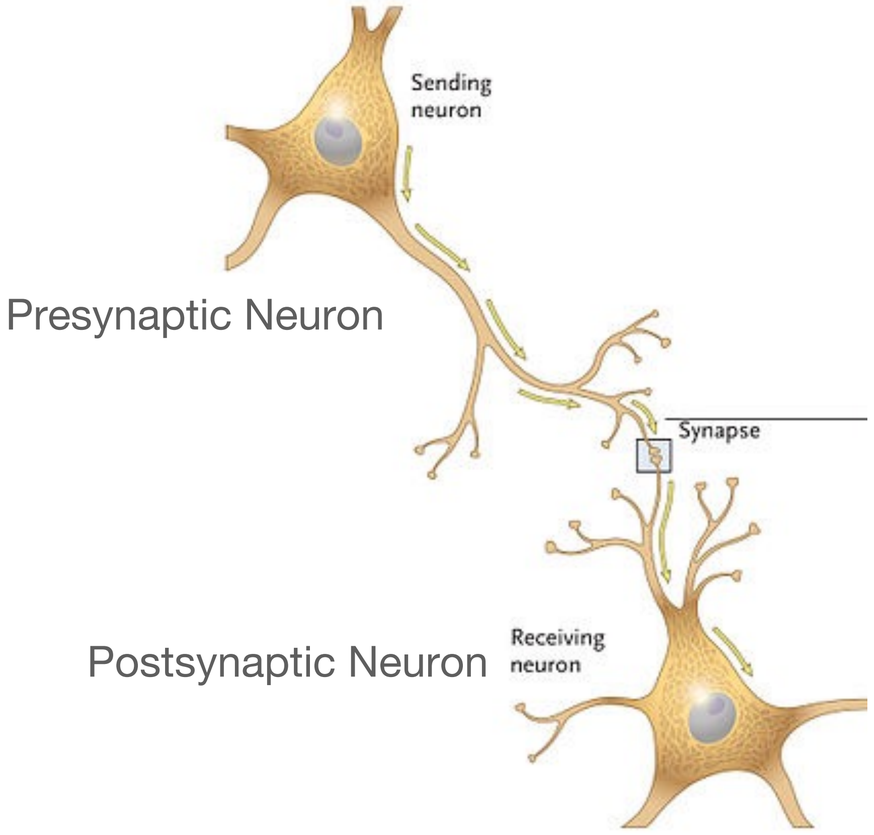
```
dt = 0.1;
T = np.arange(0, 1000, dt);

for i in range(len(T)-1):
    dv = dt/tau*((E_r – v[i]) + R*I[i])
    v[i+1] = v[i] + dv
```

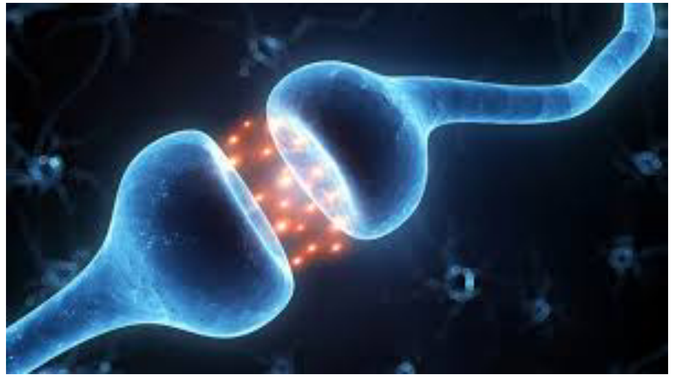
Modelling neurons

Modelling neuronal networks

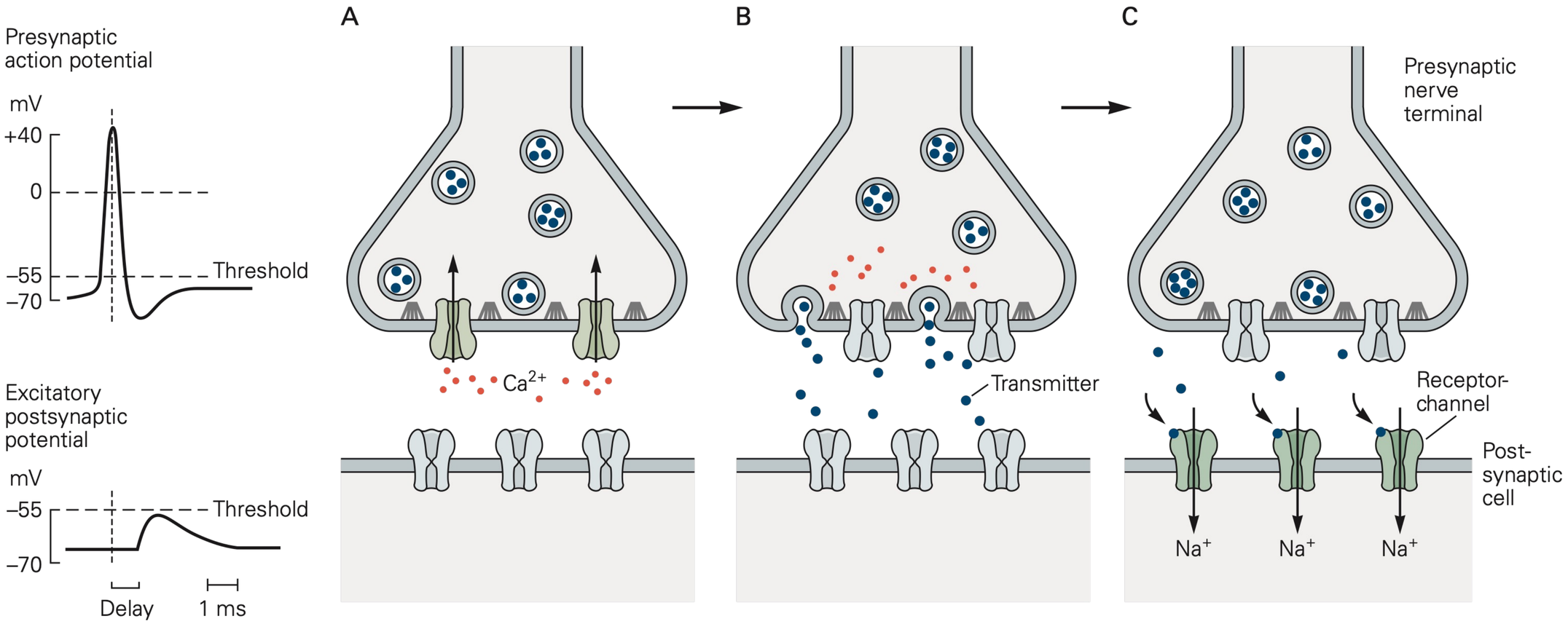
Modelling learning and plasticity

Neuronal communication and synaptic transmission

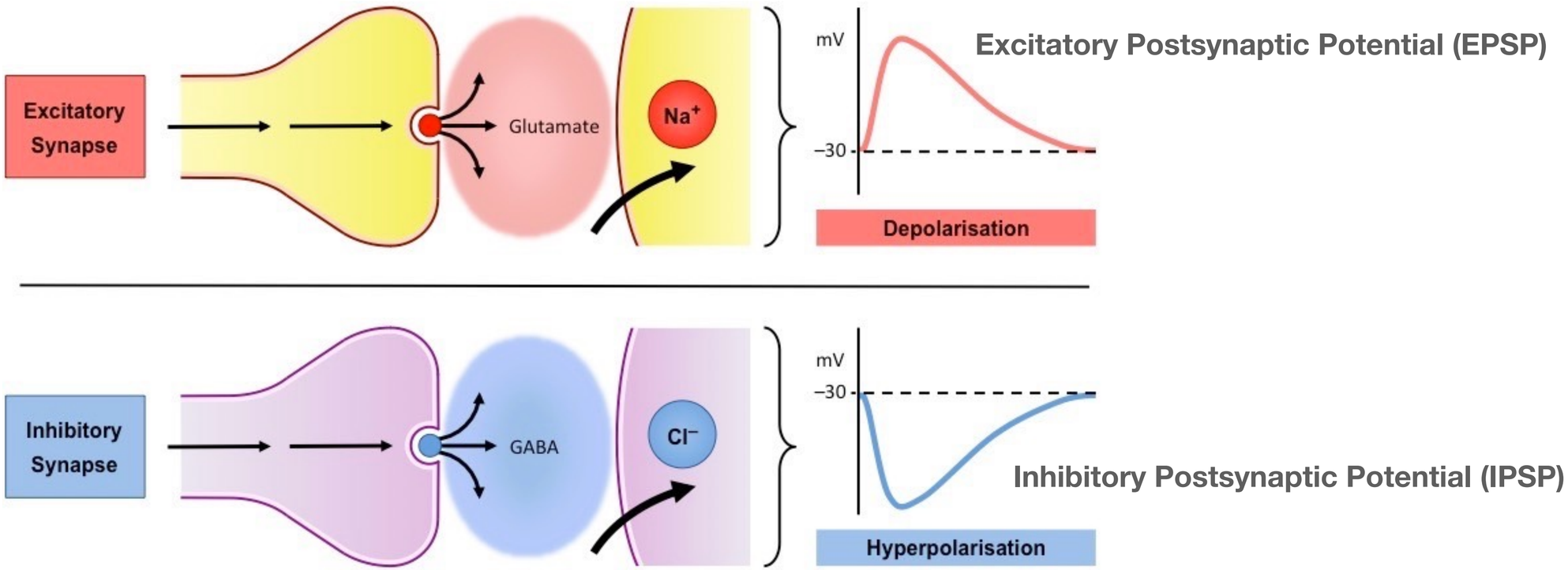




Synaptic transmission: Chemical synapses



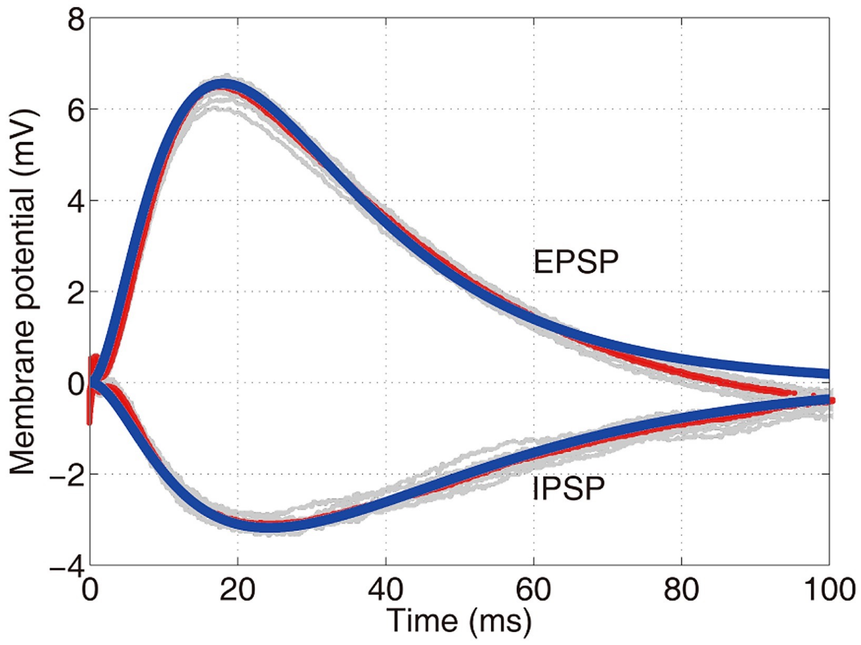
Excitatory vs inhibitory synapses

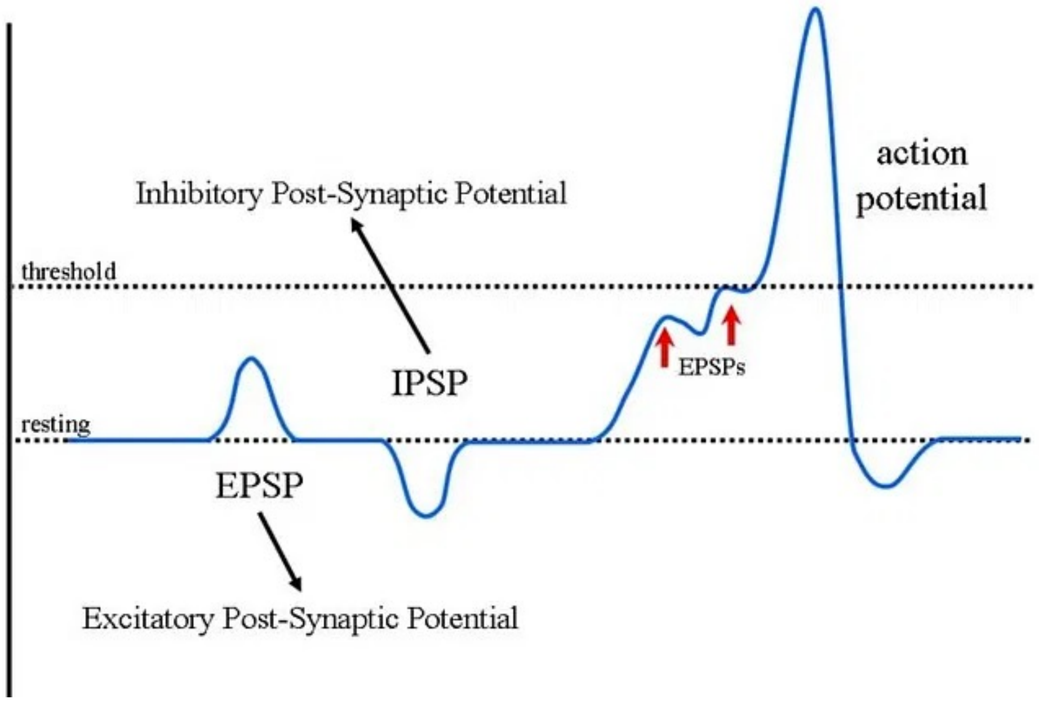


Species of ion	Concentration in cytoplasm (mM)	Concentration in extracellular fluid (mM)
K^+	400	20
Na^+	50	440
Cl^-	52	560
A^- (organic anions)	385	none

Effect on membrane potential

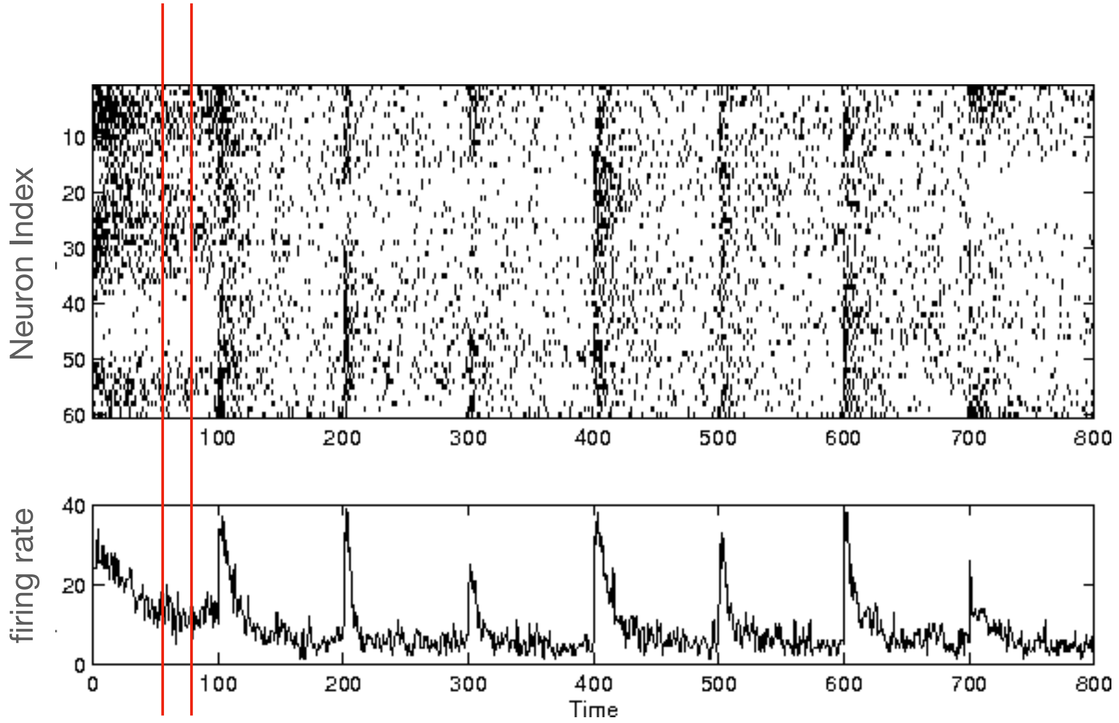
IPSP vs EPSP

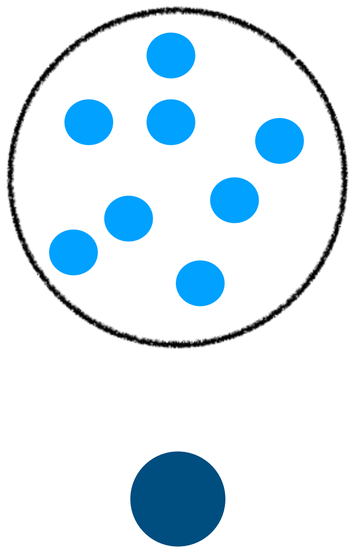




Spiking vs rate-based representation of neuronal populations

average firing rate of all the population





population
of spiking neurons

represented by one
rate neuron

- In the brain many neurons can code for the same feature (e.g. to overcome noisy spiking or spiking failure)
- One rate neuron can be used as a representation of the whole population of neurons “doing the same thing”
- The rate of this “rate neuron” describes the average firing rate of all the neurons in the population

How to simulate a simple leaky rate neuron?

leaky rate neuron

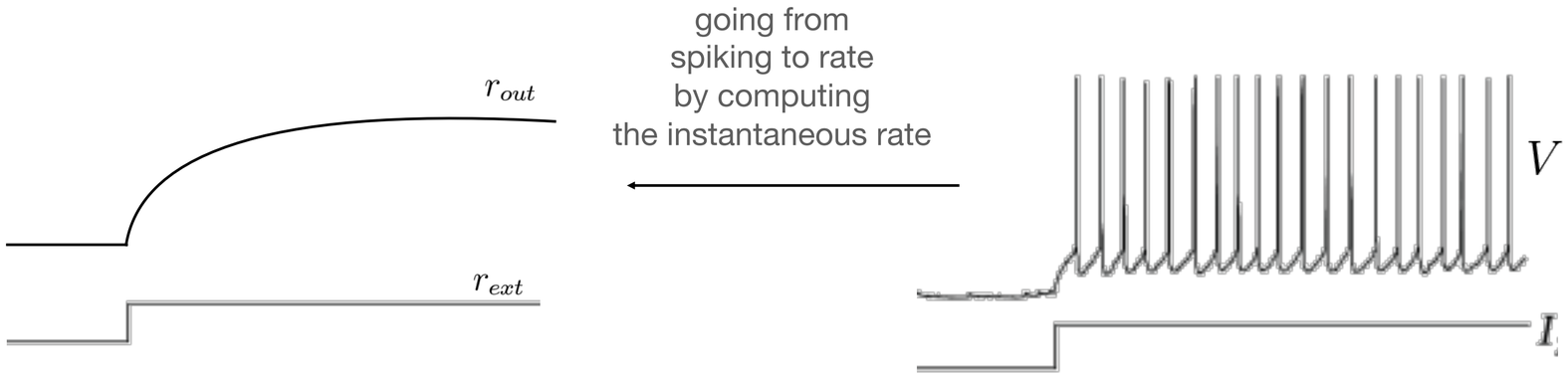
$$\frac{dr_{out}}{dt} = -r_{out} + r_{ext}$$

leaky integrate-and-fire neuron

$$\tau \frac{dV}{dt} = (E_{rest} - V) + R_m I_{ext}$$

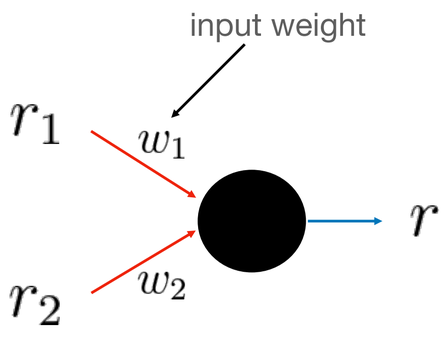
- Simulating a leaky **rate** neuron is similar to simulating a leaky integrate-and-fire **spiking** neuron:

1. Instead of simulating the evolution of the membrane potential, we simulate evolution of the firing rate,
2. No spiking mechanism.



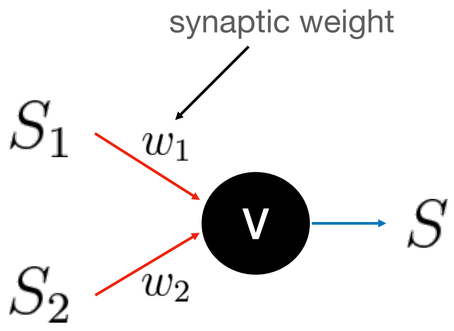
Simple feedforward network

Rate neuron



A rate neuron receives two firing rates as inputs

Spiking neuron



A spiking neuron receives two spike trains as inputs

How should we model them?

$$\tau \dot{r} = -r + w_1 r_1 + w_2 r_2$$

differential equation describing the dynamics of the output firing rate

$$\tau \dot{V} = -V + w_1 S_1 + w_2 S_2$$

differential equation describing the dynamics of the output firing rate

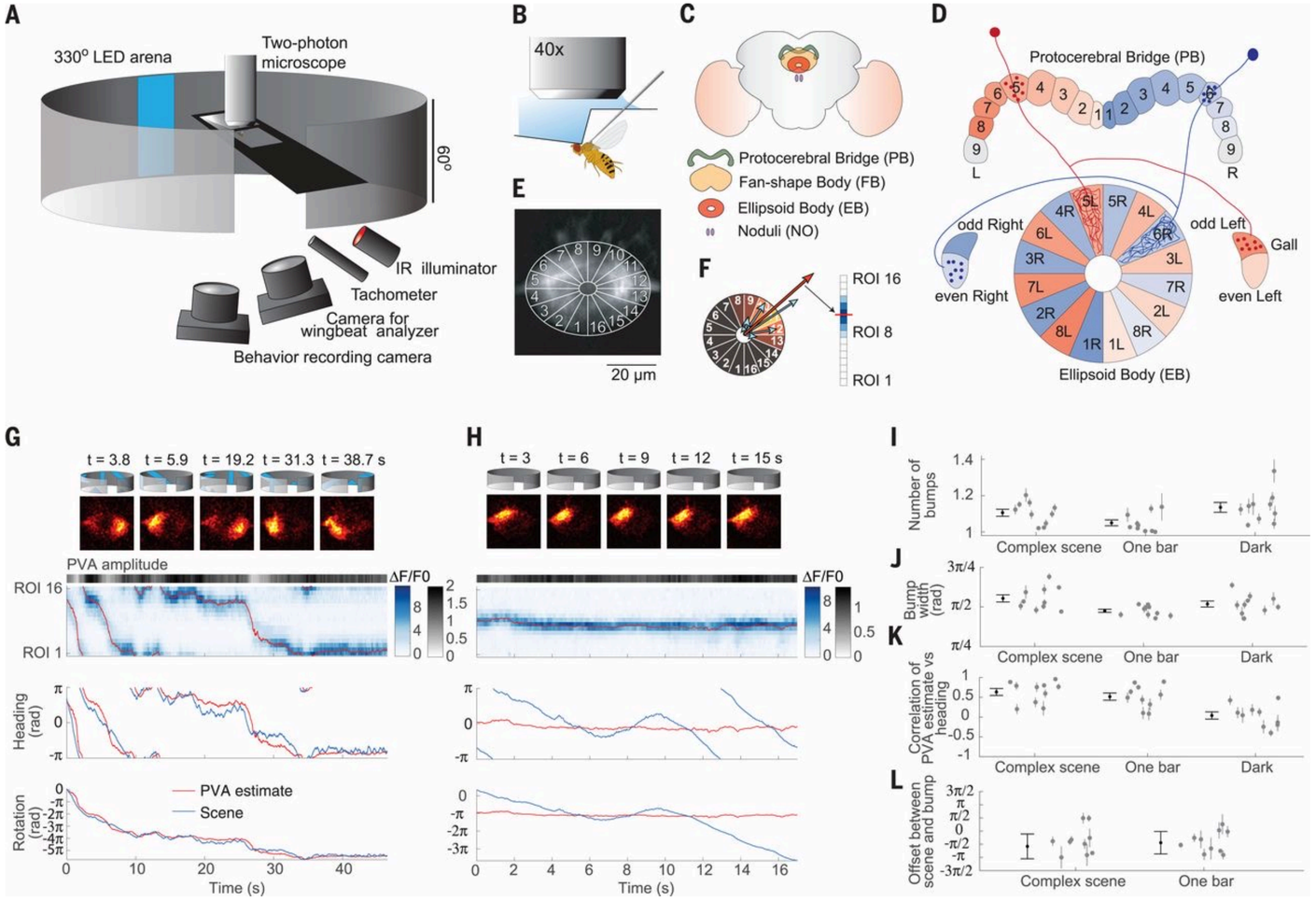
Recurrent networks and recurrent processing

Recurrent processing:

- Amplification
- Persistent activity
- Competition
- Balance of excitation and inhibition

Example: Models of persistent activity

- Attractor network models
- Persistent activity / short term memory
- Head direction in drosophila



Kim et al (Science, 2017)

Repertoire of dynamics

- From asynchronous (A) to synchronous (S) states
- From regular (R) to irregular (I) states of activity

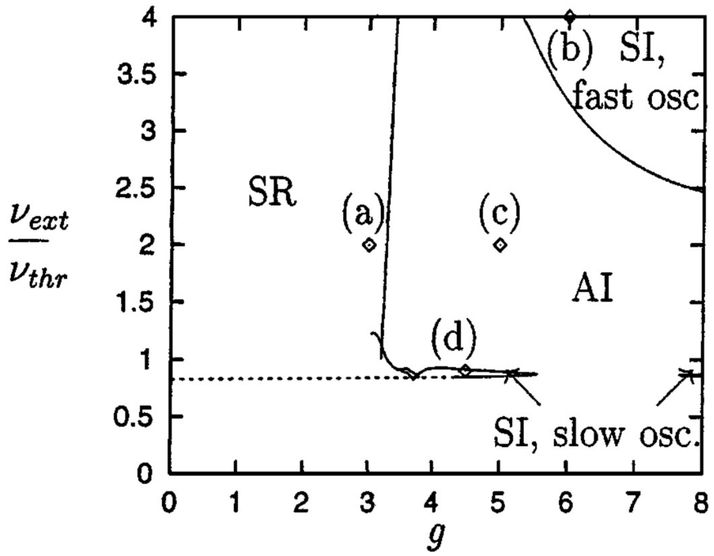
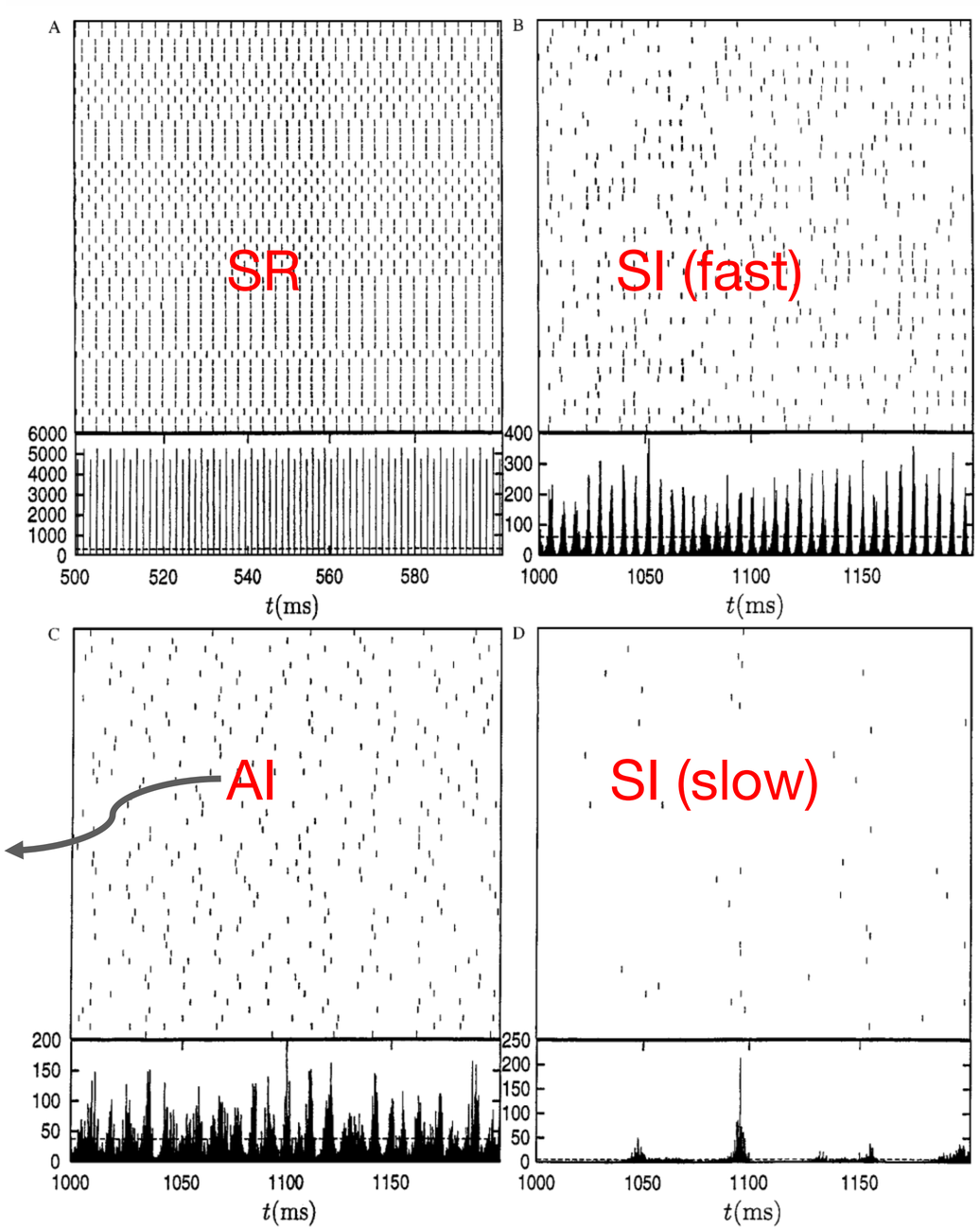


Figure 7. Phase diagram of the network characterized by the parameters of Fig. 8 ($C_E = 1000$, $C_I = 250$, $J = 0.1$ mV, $D = 1.5$ ms). Diamonds indicate the parameter sets chosen for the simulations shown in Fig. 8.

Normal regime of cortical activity (balanced)



Brunel (2000)

Pattern formation

e.g. as in hallucinations: computational modelling of visual hallucinations

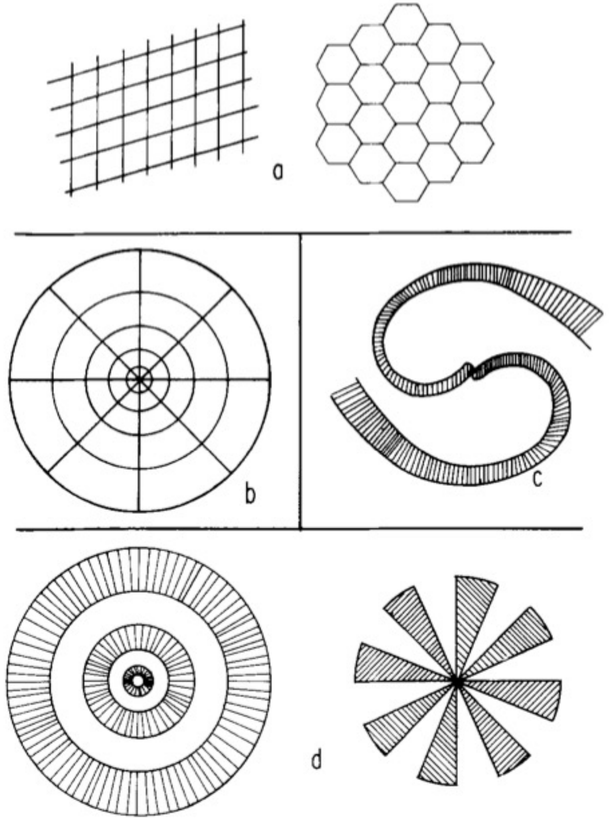


Fig. 1. **a** Typical lattice form constants. **b** Cobweb form constant. **c** Spiral form constant. **d** Tunnel and funnel constants

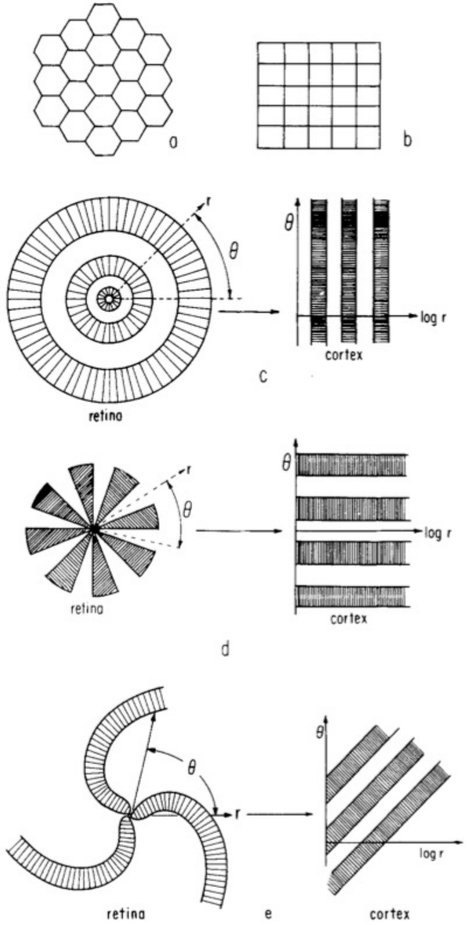
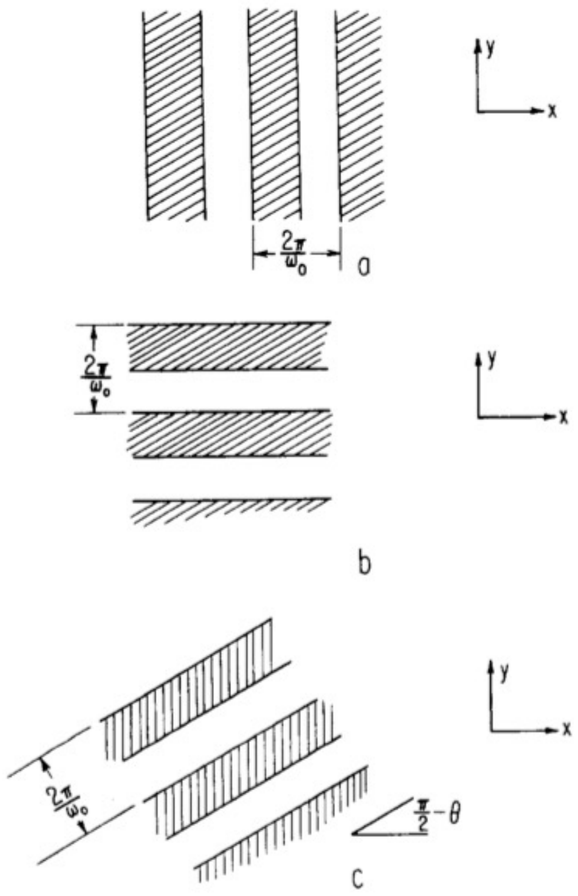


Fig. 2. **a** Pattern of a typical lattice at the cortical level. **b** Pattern of cobweb on the cortex. **c** Transformation of tunnel form constant via retino-cortical map to cortex. **d** Same, for funnel constant. **e** Same, for spiral constant

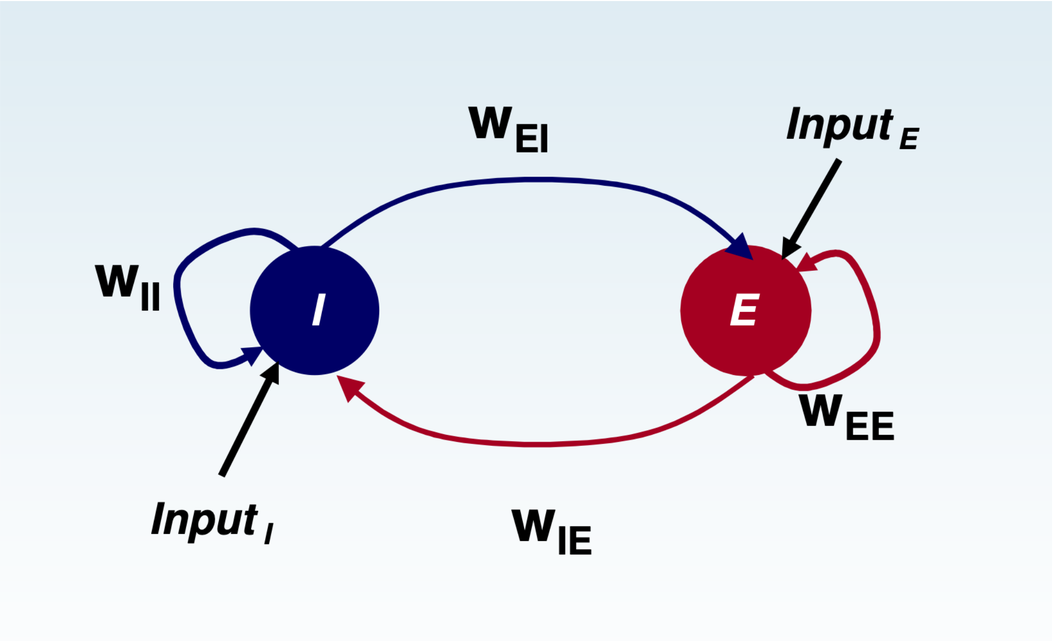


Ermentrout & Cowan (1979)

Modelling recurrent excitatory-inhibitory interactions

Wilson-Cowan Model:

- The Wilson-Cowan model describes the general dynamics of interacting **excitatory** and **inhibitory** neuronal populations.
- It describes the dynamics (transient) as well as equilibrium (steady state) states of the system.
- It can model the emergence of oscillations as a result of E-I interactions in neuronal networks.



Wilson Cowan Model

Evolution of the excitatory (E) population:

$$\tau_E \frac{dr_E}{dt} = -r_E + g(W_{EE} r_E - W_{EI} r_I + I_E)$$

Characteristic time

Leak term

Connectivity weights (recurrent coupling)

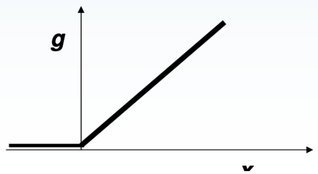
External Input

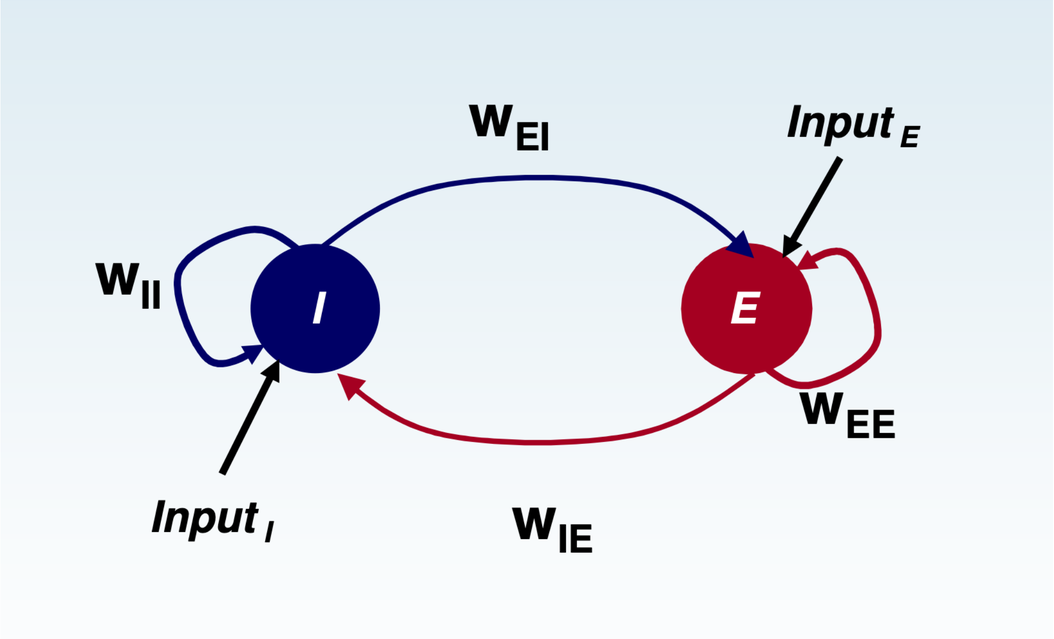
$$\tau_I \frac{dr_I}{dt} = -r_I + g(W_{IE} r_E - W_{II} r_I + I_I)$$

Evolution of the inhibitory (I) population:

activation function (e.g. linear rectified):

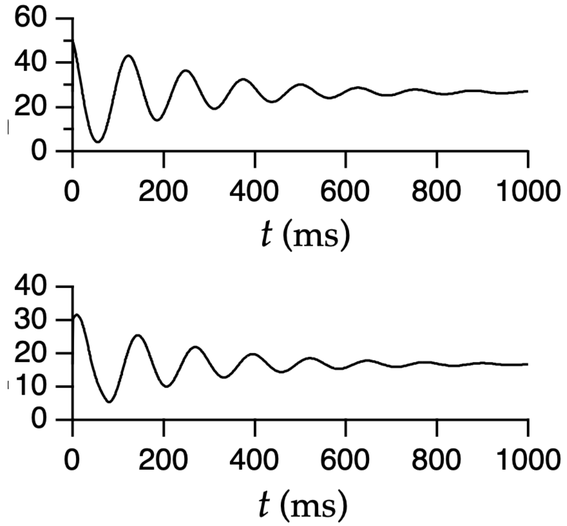
$$g(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$



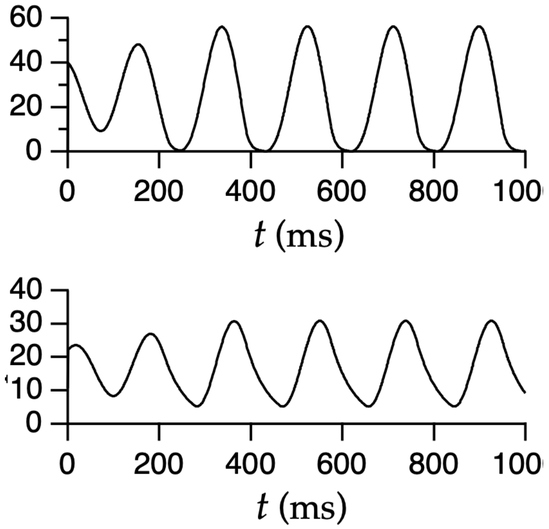


Wilson-Cowan Model – different behaviours

Depending on the combination of the parameters (the characteristic times, connection weights, …), the model can exhibit different behaviors:



The activity of the two population oscillate and each population converge to an equilibrium



The two populations settle into periodic oscillations without converging to an equilibrium

Solving differential equations

```
T = np.arange(0, T_end, dt)

r_E = np.zeros(len(T))
r_I = np.zeros(len(T))

for i in range(len(T)-1):
    dr_E = dt/tau_E * (-r_E[i] + act_fun(w_EE * r_E[i] + w_EI *r_I[i] + I_E))
    dr_I = dt/tau_I * (-r_I[i] + act_fun(w_IE * r_E[i] + w_II *r_I[i] + I_I))

    r_E[i+1] = r_E[i] + dr_E
    r_I[i+1] = r_I[i] + dr_I

   return r_E, r_I, T
```

$$\tau_E \frac{dr_E}{dt} = -r_E + g(W_{EE}r_E - W_{EI}r_I + I_E)$$

$$\tau_I \frac{dr_I}{dt} = -r_I + g(W_{IE}r_E - W_{II}r_I + I_I)$$

Modelling neurons

Modelling neuronal networks

Modelling learning and plasticity

Modelling learning: across scales

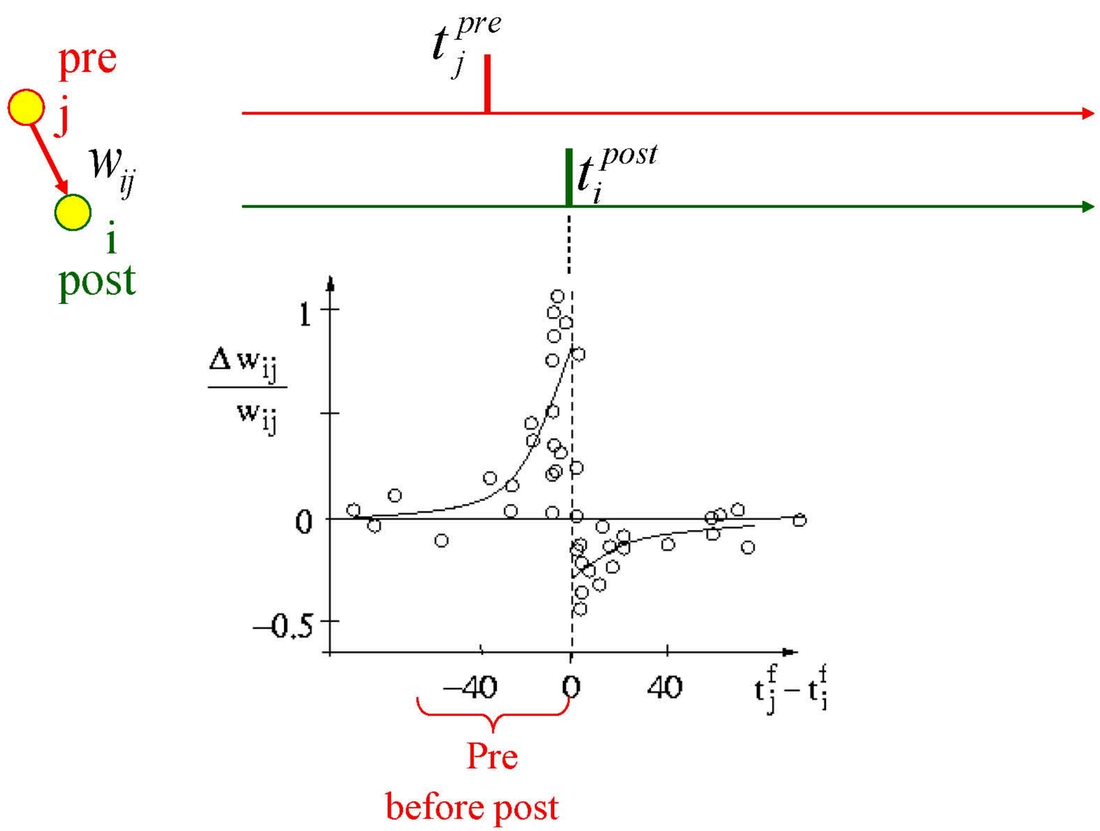
Learning spans a large range of mechanisms, across multiple scales

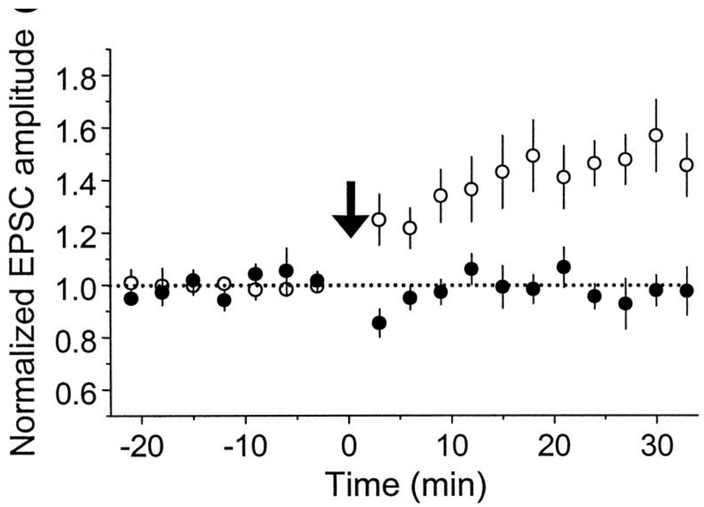
(from synapse to behaviour):

- **Spike timing dependent plasticity (STDP)**
- Voltage dependent rules (+ spiking)
- The role of dendrites → plateau potentials, back-propagating APs
- Hebbian learning / covariance rule
- BCM (phenomenological rules)
- Backpropagation in artificial networks and machine learning
- Bayesian learning
- **Reinforcement learning (RL)**

Spike timing dependent plasticity

- a temporally asymmetric form of Hebbian learning
- induced by tight temporal correlations between the spikes of pre- and postsynaptic neurons.
- Hippocampal culture





Bi and Poo (1998)

Sjöström and Gerstner (2010)

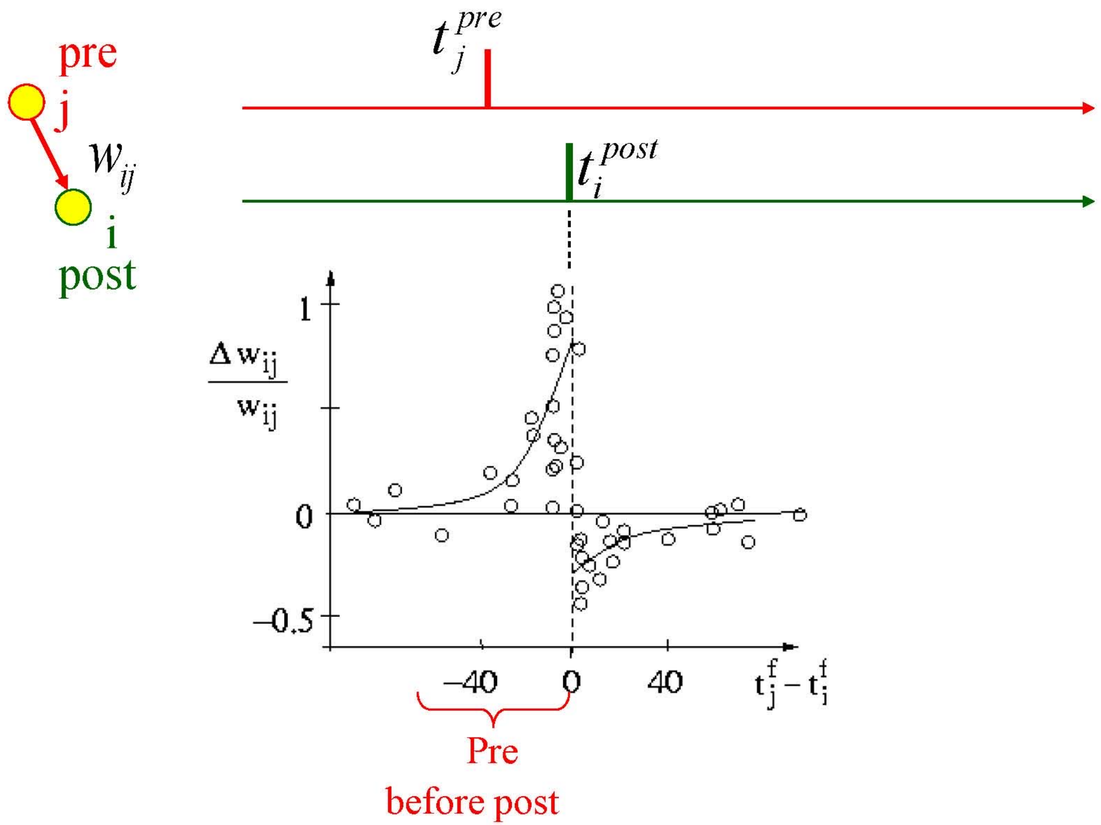
Implementation

Sum of all weight changes resulting from all pairs of pre- and postsynaptic spikes:

$$\Delta w_j = \sum_{f=1}^{N} \sum_{n=1}^{N} W(t_i^n - t_j^f)$$

$$W(x) = A_+ \exp(-x/\tau_+) \qquad for \quad x > 0$$

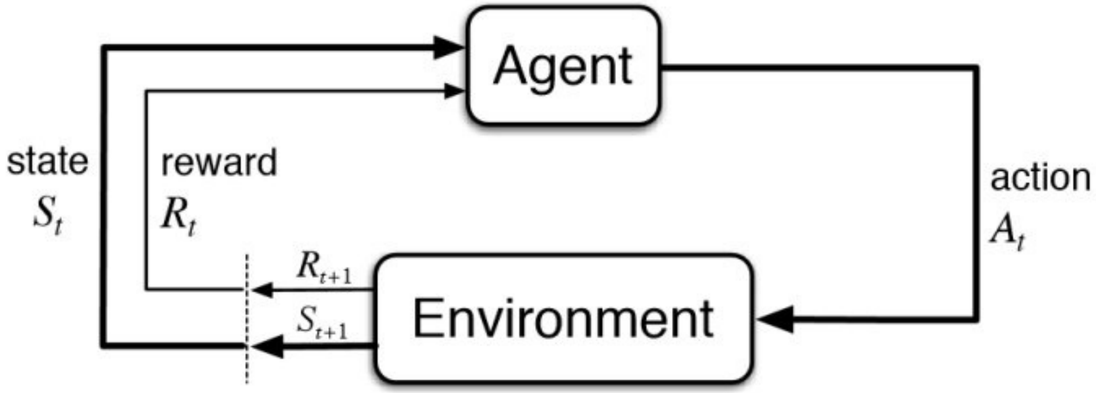
$$W(x) = -A_- \exp(x/\tau_-) \qquad for \quad x < 0$$



Sjöström and Gerstner (2010)

Reinforcement Learning

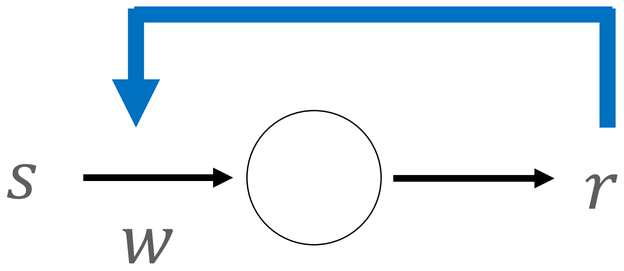
- Reward driven learning
- Learning from your actions – output
- Try; see the outcome; adjust the weights / actions accordingly
- Feedback learning?

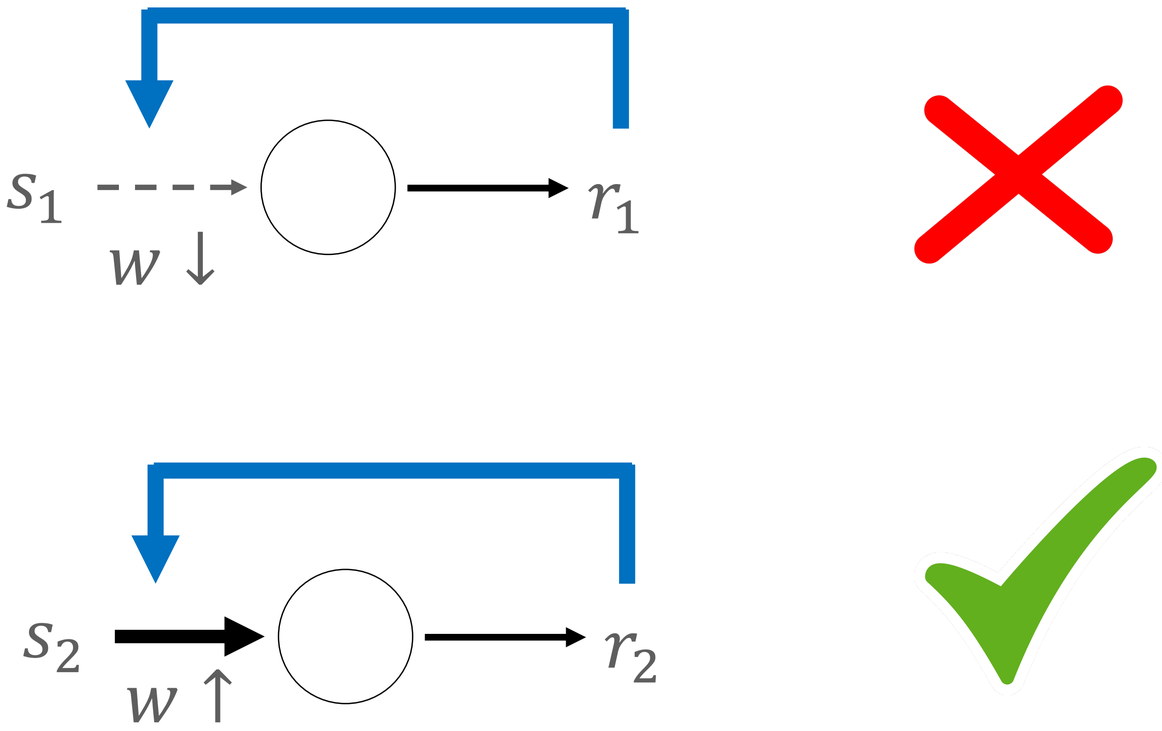


RL and feedback

Reinforcing the input / weights

as a result of learning from the output / results





Multiple states

Reinforcing the input / weights as a result of learning from the output / results

Multiple inputs

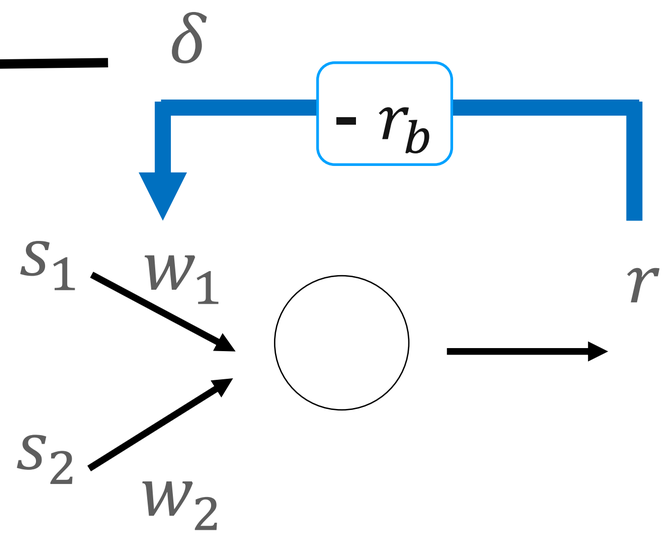
Error signal

Feedback from: not the absolute value …

Comparing the results / output with expectations

Learning / feedback from the error: deviation from expectations

Learning from the “error signal”



Updating the weights (update rule):

$$w \leftarrow w + \eta\, \delta$$

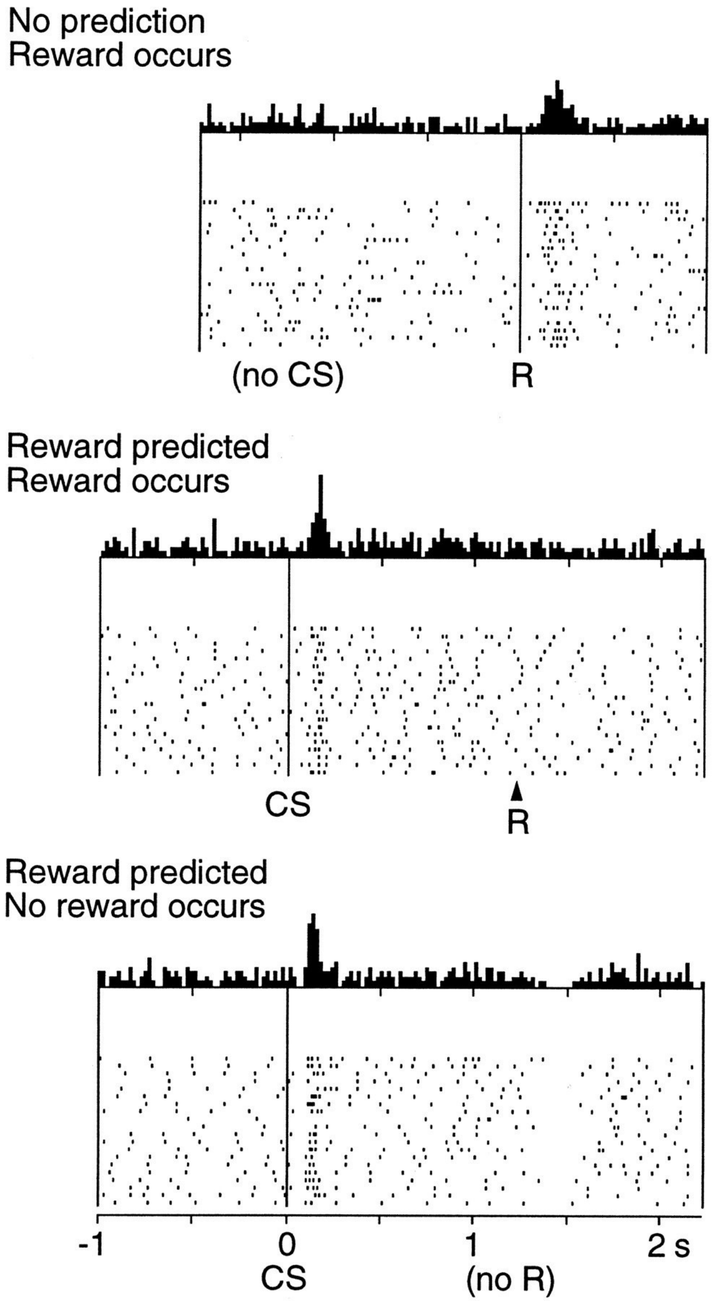
$$\delta = (r - r_b)$$

η: learning rate

Error signal in the brain

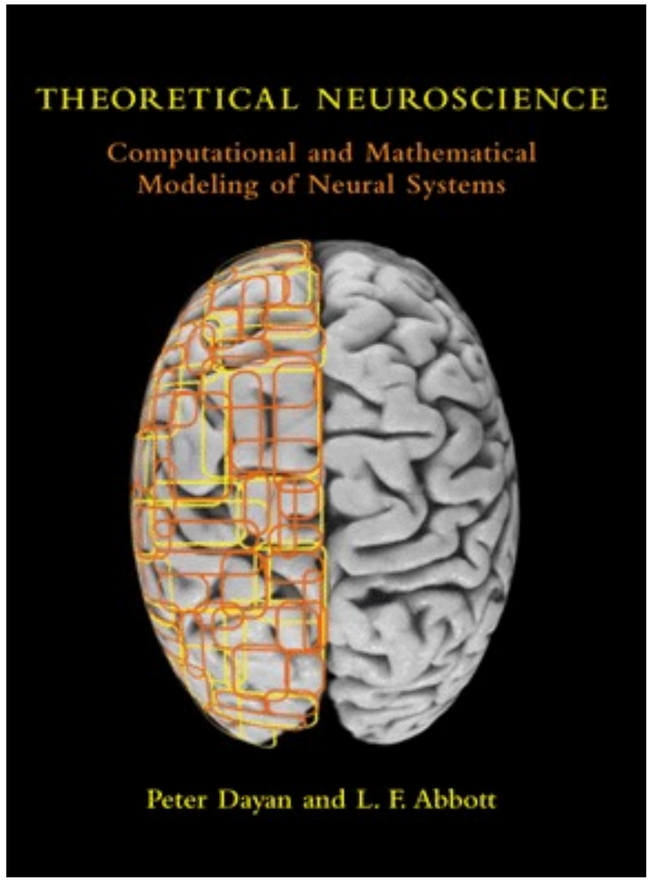
- **Behaviour**:
- Learning is driven by changes in the **expectations** about future salient events such as rewards

- **Physiology**:
- Encoding of changes or errors in the predictions of future reward in **dopaminergic neurons**

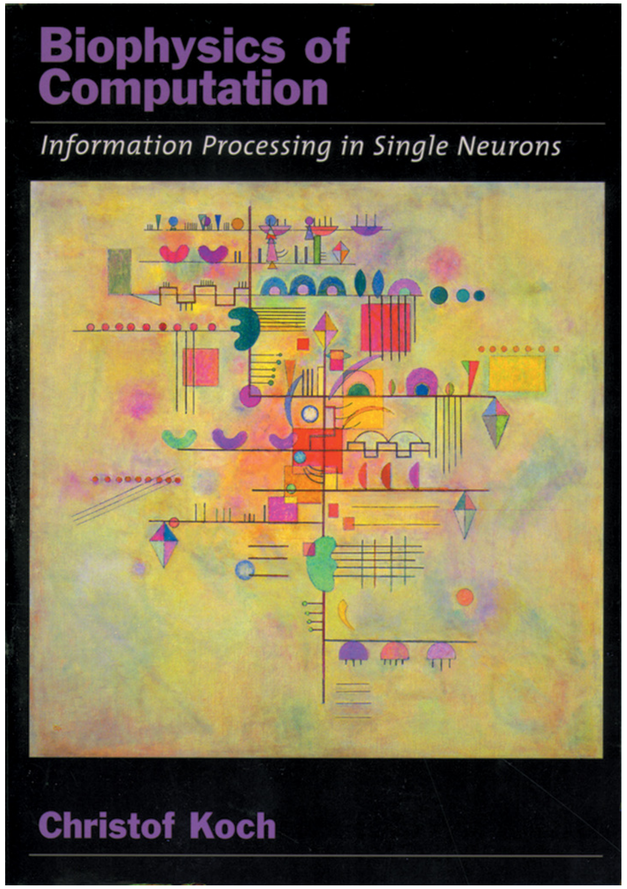


Schultz (1998)

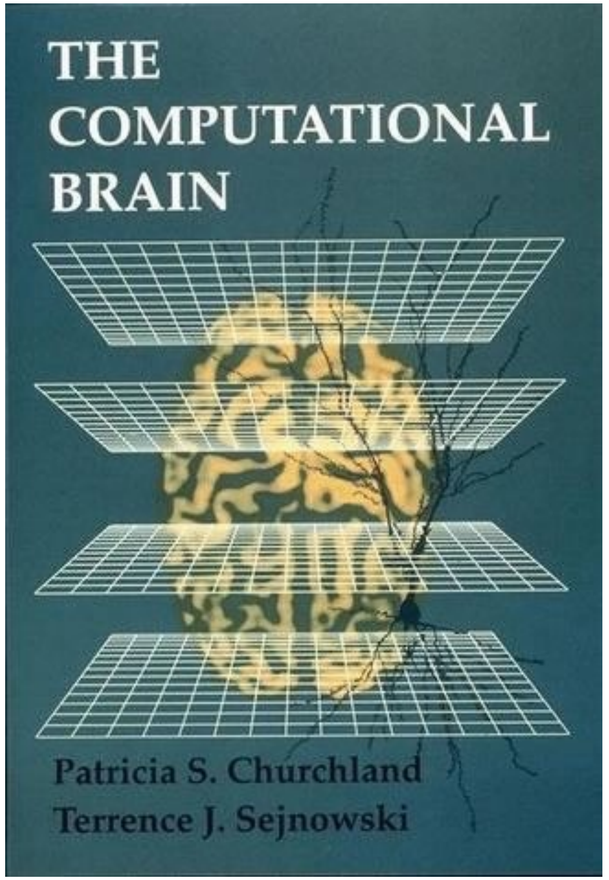
Text books



Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems (Dayan and Abbott)
http://www.gatsby.ucl.ac.uk/~dayan/book/



Biophysics of Computation: Information Processing in Single Neurons (Christof Koch)



The Computational Brain (Churchland and Sejnowski)