PY411 / 506
Computational Physics 2

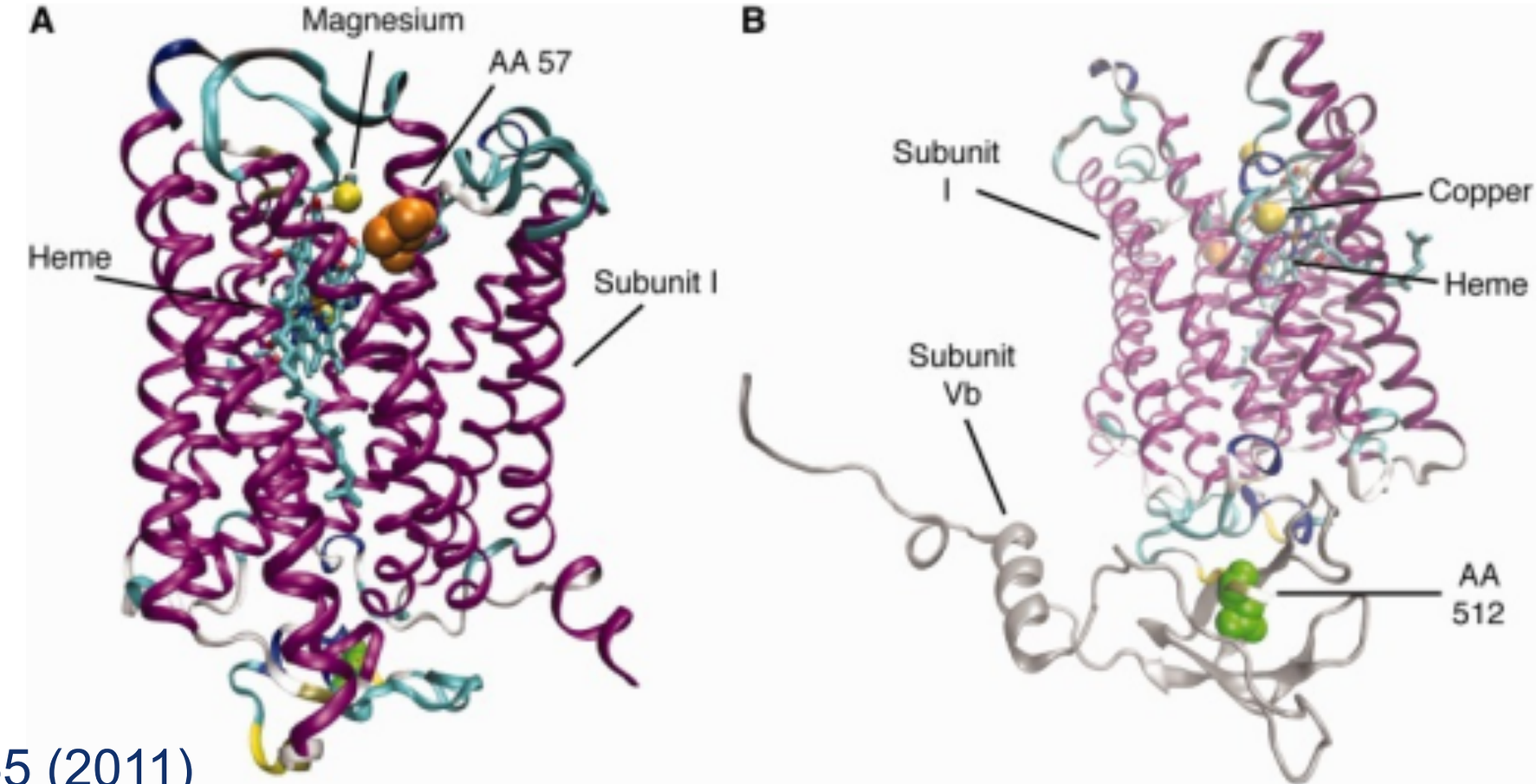
Salvatore Rappoccio

Biophysics and Neurons

- Now moving toward some methodologies involving proteins and neurons

- This will also segue into artificial neural networks for the data analysis and machine learning section later in the semester

Model of Bovine Cytochrome C

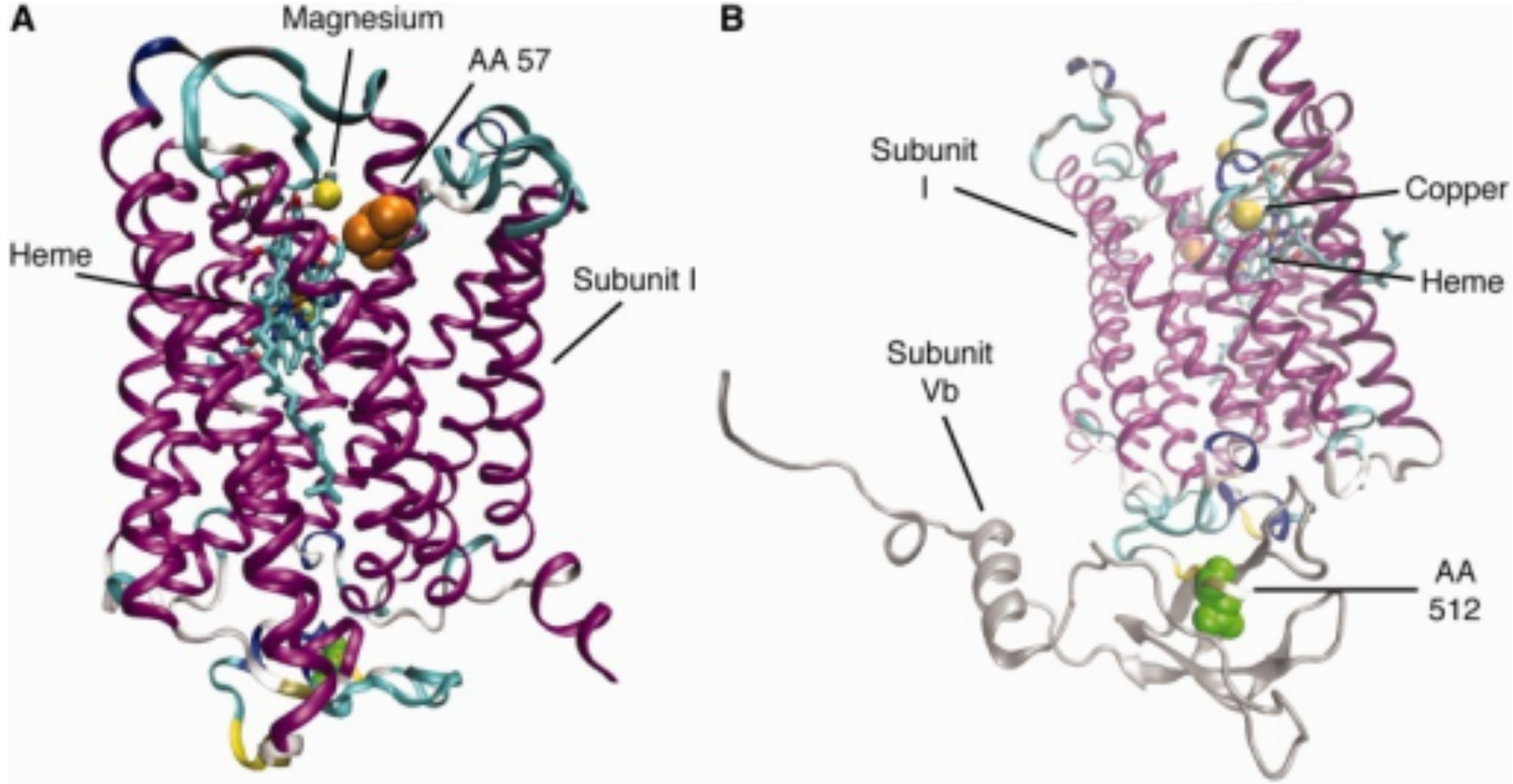


Biophys. J. 100, 1058-1065 (2011)

Biophysics and Neurons

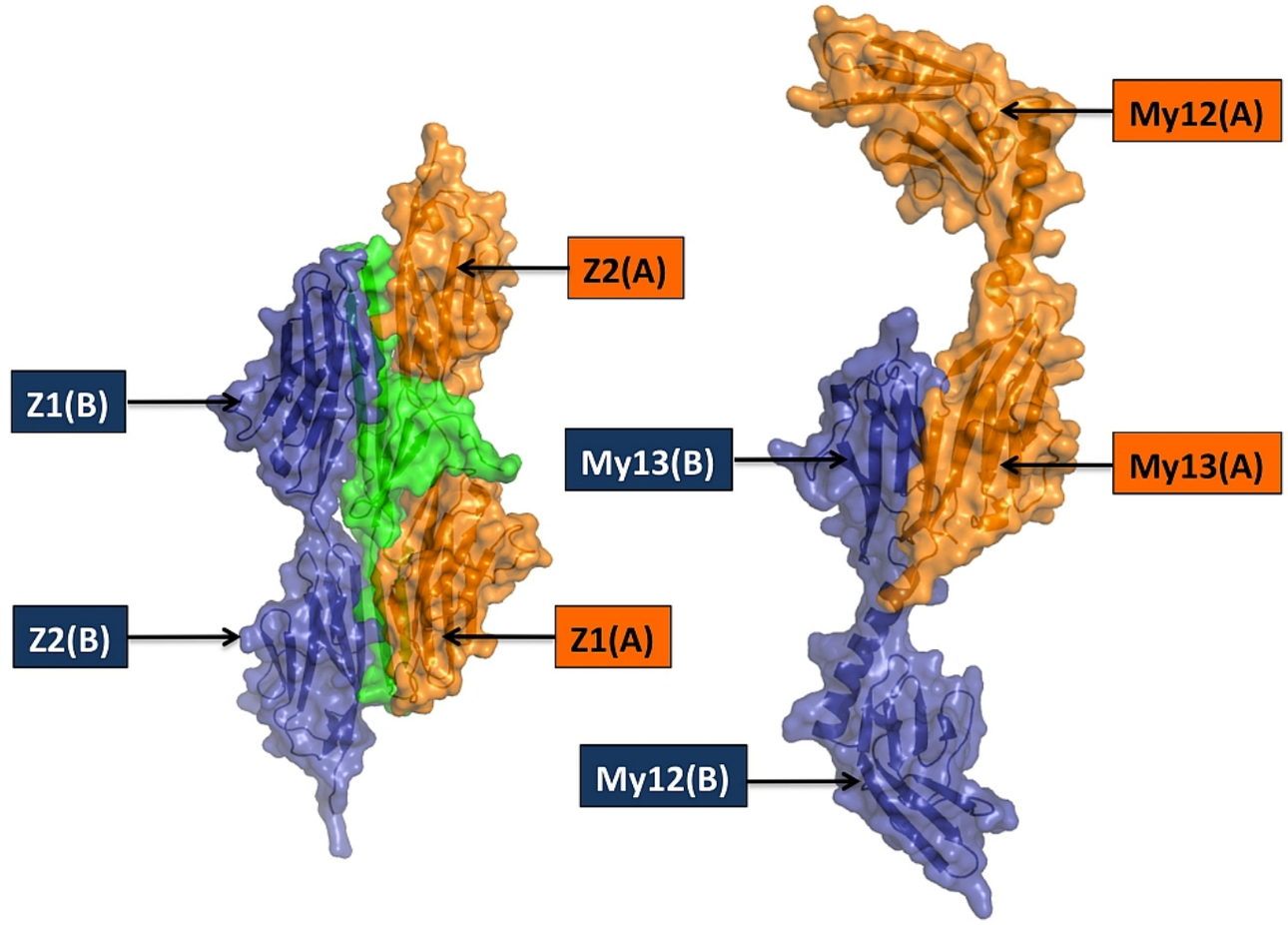
- Protein folding MC methods
  - Example: Bovine Cytochrome C
  - Biophys. J. 100, 1058-1065 (2011) (Markelz group + Zheng)
  - THz spectroscopy and molecular simuation

Model of Bovine Cytochrome C



Biophysics and Neurons

- Polymers can range from 20 amino acids (Glycine) up to 30,000 (Titin)
- 6 degrees of freedom for each amino acid, so this becomes quickly intractable

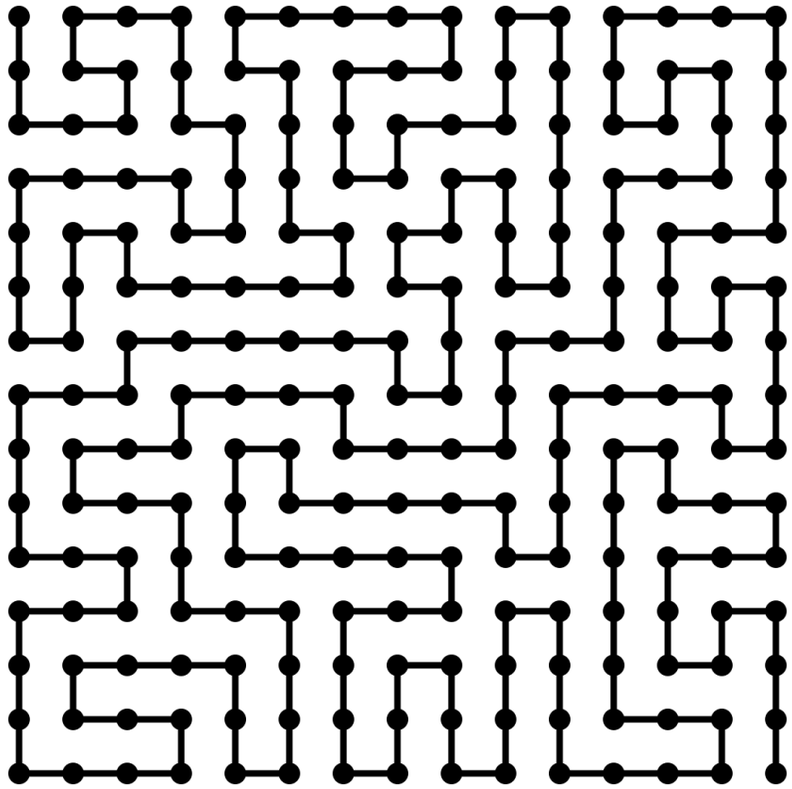


Titin(N)-telethonin-titin(N)

Myomesin(C)-Myomesin(C)

Biophysics and Neurons

- Shape of the protein impacts the interactions, so look into that
- Investigate protein folding in a simpler way first
- Model as n+1 monomers joined by n strong covalent bonds
- But! The monomers cannot occupy the same place in space:
  - **Self-avoiding!**
- So this means it is not a Markov chain



Self-avoiding Walk

- 1d: Trivial. Everything on a line, and you walk the entire chain left or right.
  - Ballistic transport: $|x| = n \sim t$
- 1d simple walk:
  - Diffusive motion: $\sqrt{< x_n^2 >} = \sqrt{n} \sim \sqrt{t}$
- 2-d Self-avoiding walk: $\sqrt{< r^2 >} \sim A t^{\nu}$
  - "Flory exponent": $\frac{1}{2} < \nu < 1$

Self-avoiding Walk

- How to generate a SAW?
- Try trivial example: N walkers, if you collide, discard
- Look at “sawalk.py” in Lecture 41

Self-avoiding Walk

- Extremely inefficient! Nstep = 10, Nwalk = 10 tractable, move to Nstep = 100, Nwalk = 100, extremely long!

- Fraction of discarded walks increases exponentially! (Whoa! Awful!)

$$\frac{\text{Walks Generated}}{\text{Total Number of Attempts}} \sim e^{-\lambda n}$$

Attrition constant

Self-avoiding Walk

- To enumerate this:

```
Exact Enumeration of Self-Avoiding Walks on a Square Lattice
------------------------------------------------------------
Enter maximum number of steps: 20
```

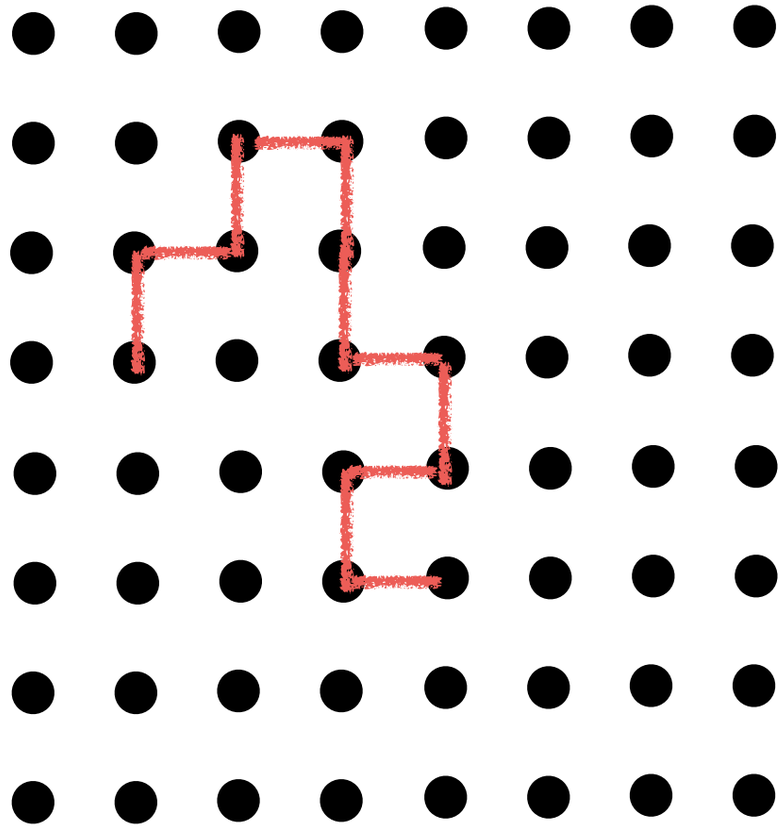
Steps	<r^2>	Std. Dev.	S.-A. Walks	Cul-de-sacs	Double Cds	CPU secs
1	1	0	4	0	0	0
2	2.66667	0.942809	12	0	0	0
3	4.55556	2.26623	36	0	0	0
4	7.04	3.53814	100	0	0	0
5	9.56338	5.21302	284	0	0	0
6	12.5744	6.90435	780	0	0	0
7	15.5562	8.9282	2172	16	0	0
8	19.0128	10.9428	5916	32	0	0
9	22.4114	13.2495	16268	176	0	0
10	26.2425	15.5487	44100	400	0	0
11	30.0177	18.1032	120292	1520	0	0
12	34.187	20.6544	324932	3648	16	0.016
13	38.3043	23.434	881500	12032	8	0.016
14	42.7864	26.2132	2374444	29760	152	0.031
15	47.2177	29.2016	6416596	91856	176	0.078

Self-avoiding Walk

- Instead, use the reptation method
  - Developed by Pierre-Gilles de Gennes (Nobel Prize in 1991)
  - Assumes polymers are confined to a tube
  - Tube “snakes” through the tunnel
  - Thermal fluctuations cause polymer to “reptate” (like a snake)
- Slight adaption: Wall and Mandel, J. Chem. Phys. 63, 4592 (1975)
  - “Slithering snake” model instead
  - Uses MC techniques

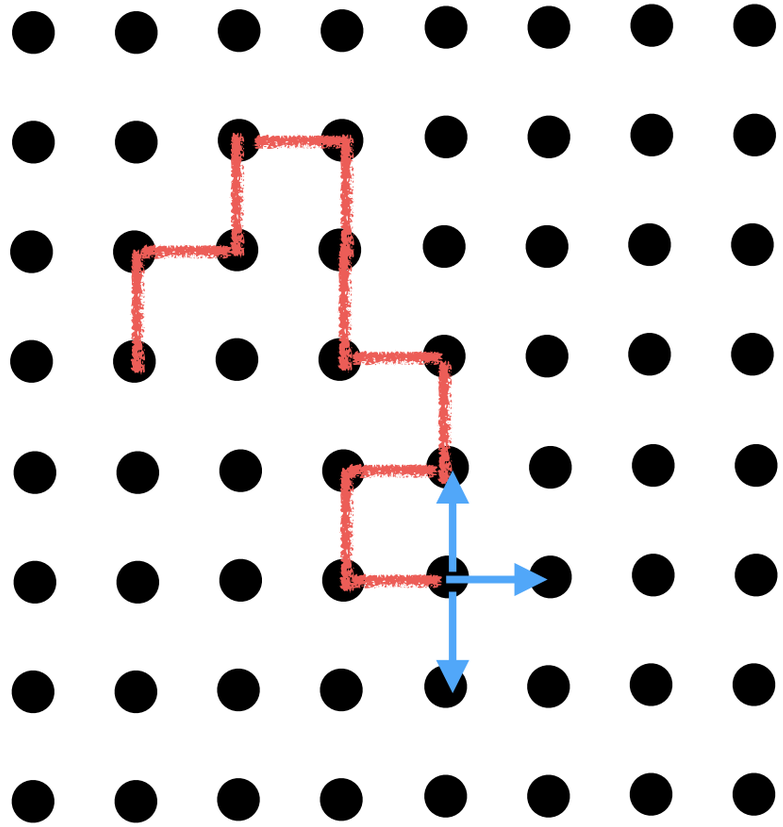
Self-avoiding Walk

- Start with chain of n links
- Iterate:
  - Choose an end of the chain at random
  - Choose 3 reptation directions (forward, left, right)
  - If site in randomly chosen direction is not one of interior sites of chain:
    - Remove the site at the other end of the chain
    - Add this site to the chain
    - Count new config as next in ensemble
- Else:
  - Retain config as next in ensemble



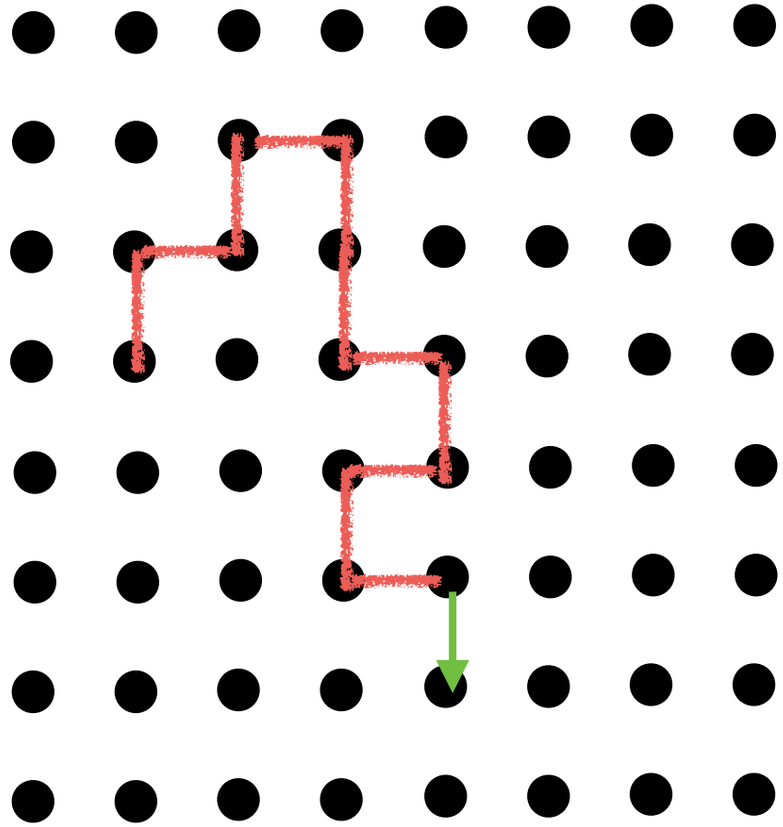
Self-avoiding Walk

- Start with chain of n links
- Iterate:
  - Choose an end of the chain at random
  - Choose 3 reptation directions (forward, left, right)
  - If site in randomly chosen direction is not one of interior sites of chain:
    - Remove the site at the other end of the chain
    - Add this site to the chain
    - Count new config as next in ensemble
- Else:
  - Retain config as next in ensemble



Self-avoiding Walk

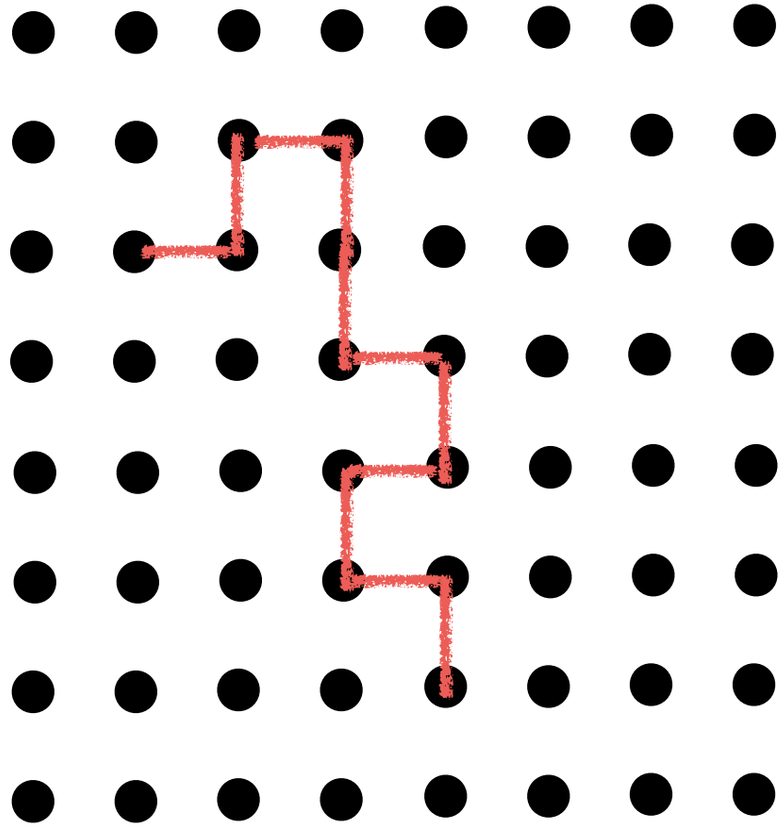
- Start with chain of n links
- Iterate:
  - Choose an end of the chain at random
  - Choose 3 reptation directions (forward, left, right)
  - If site in randomly chosen direction is not one of interior sites of chain:
    - Remove the site at the other end of the chain
    - Add this site to the chain
    - Count new config as next in ensemble
- Else:
  - Retain config as next in ensemble



Gets “down”

Self-avoiding Walk

- Start with chain of n links
- Iterate:
  - Choose an end of the chain at random
  - Choose 3 reptation directions (forward, left, right)
  - If site in randomly chosen direction is not one of interior sites of chain:
    - Remove the site at the other end of the chain
    - Add this site to the chain
    - Count new config as next in ensemble
  - Else:
    - Retain config as next in ensemble



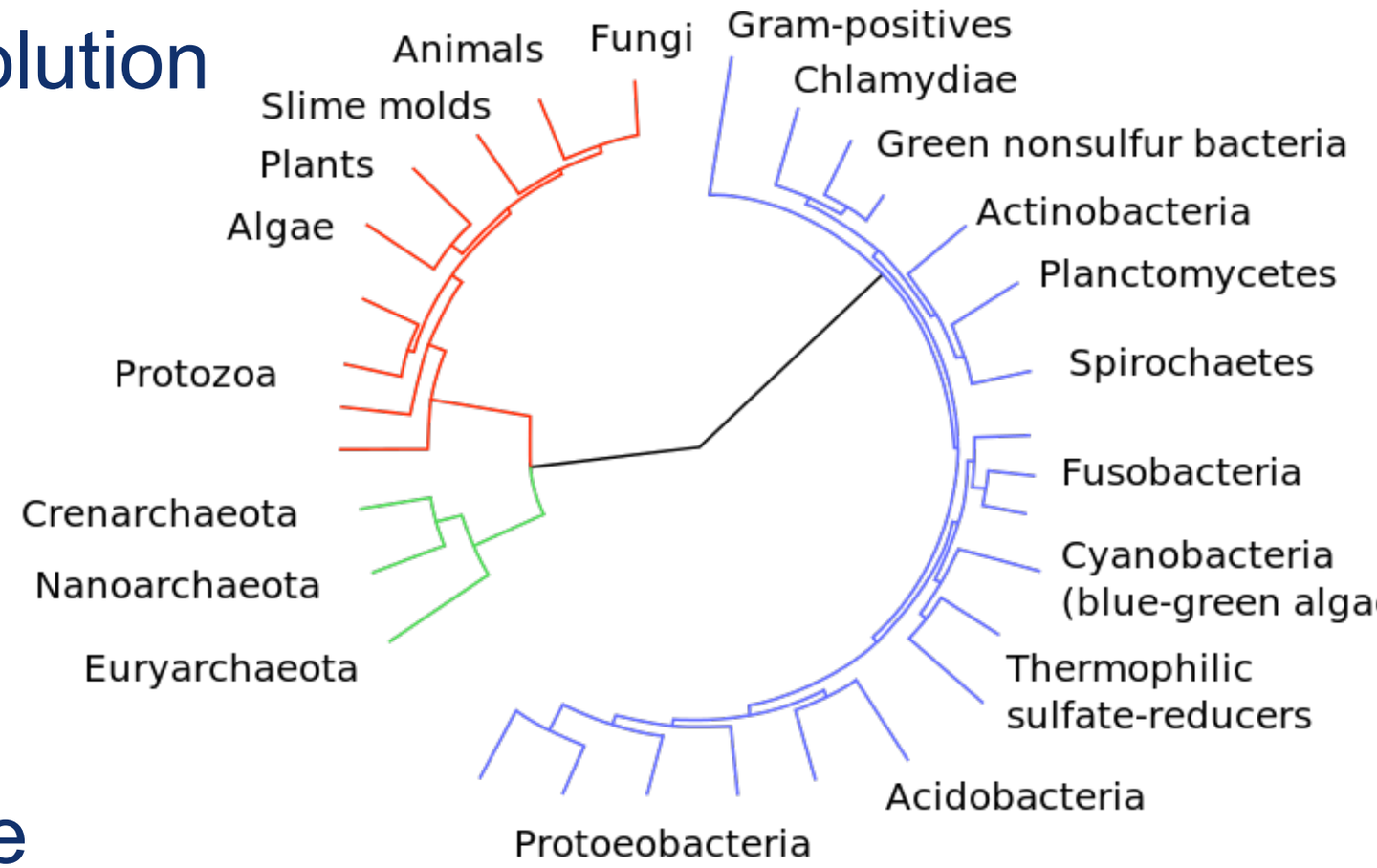
Remove the end, add a new node to front

Genetic Algorithms

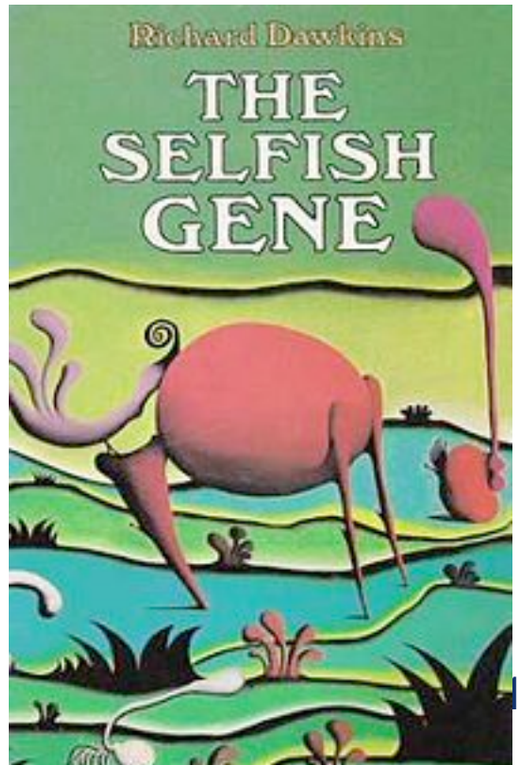
- In recent history, genetic algorithms are becoming very popular, so we can look into them
  - http://en.wikipedia.org/wiki/Genetic_algorithm

Genetic Algorithms

- Based on biological evolution and natural selection



- Idea is exactly the same for genetic algorithms
- “Species” “compete” for “resources” and the “fittest genes” propagate through the generations



Genetic Algorithms

- How does evolution work?
  - Have a genome
  - Mutations occur
  - Some are more beneficial than others
  - Organisms with more beneficial genes reproduce more

- Exactly the same way for CS, but the mapping is :

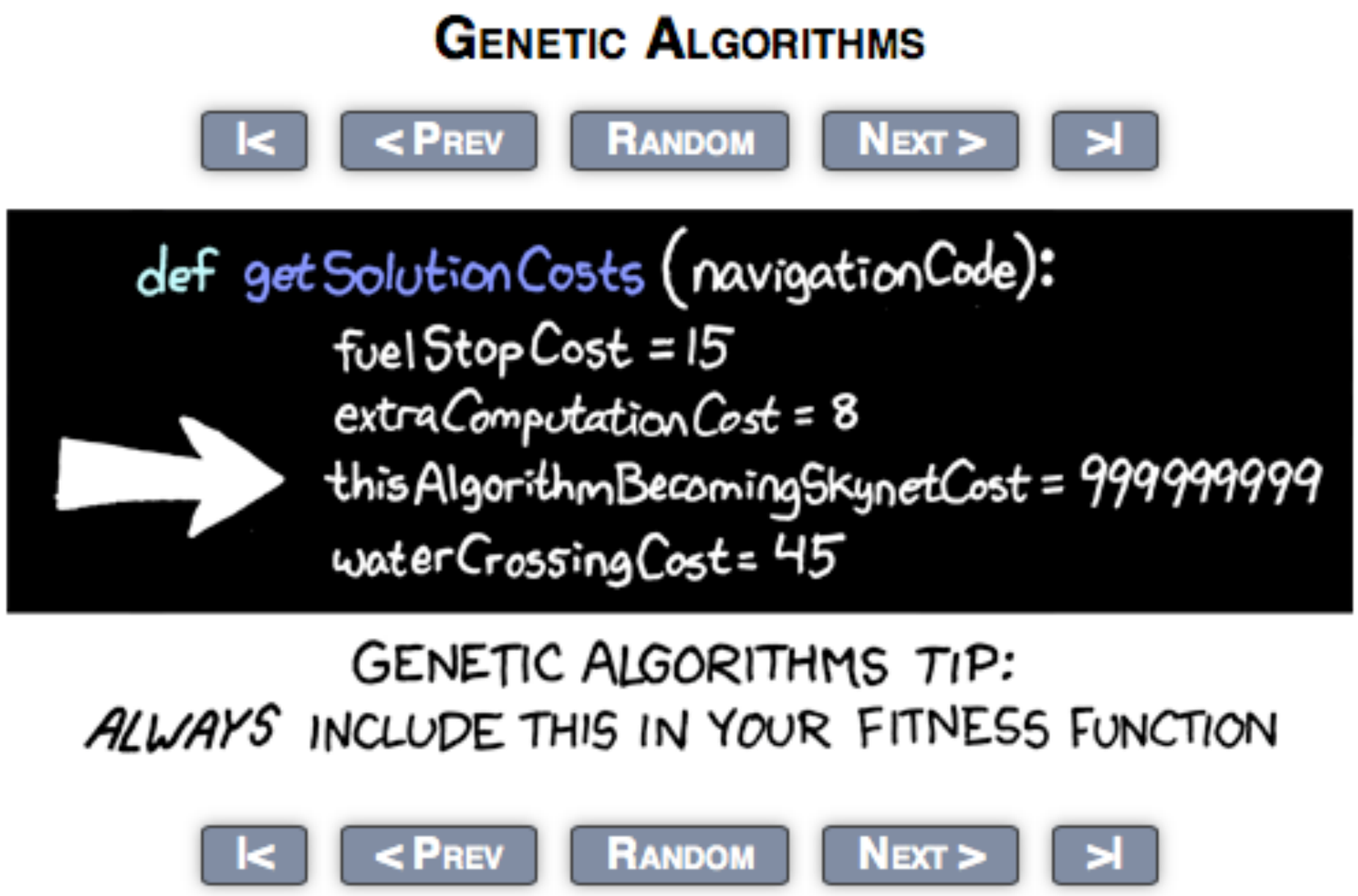
Biological evolution	Genetic algorithms
Species	Parameters for a code (“state”)
Competition for survival in nature	Competition for survival against a fitness function
Death	Iteration

Genetic Algorithms

- How to represent “DNA” / chromosomes?
- Lots of ways, but for us we can just use a sequence of bits (i.e. 100101101)
- We can “evolve” these bitstreams and select the ones that maximize the fitness function
- Once we’re close enough, we’re done!

Genetic Algorithms

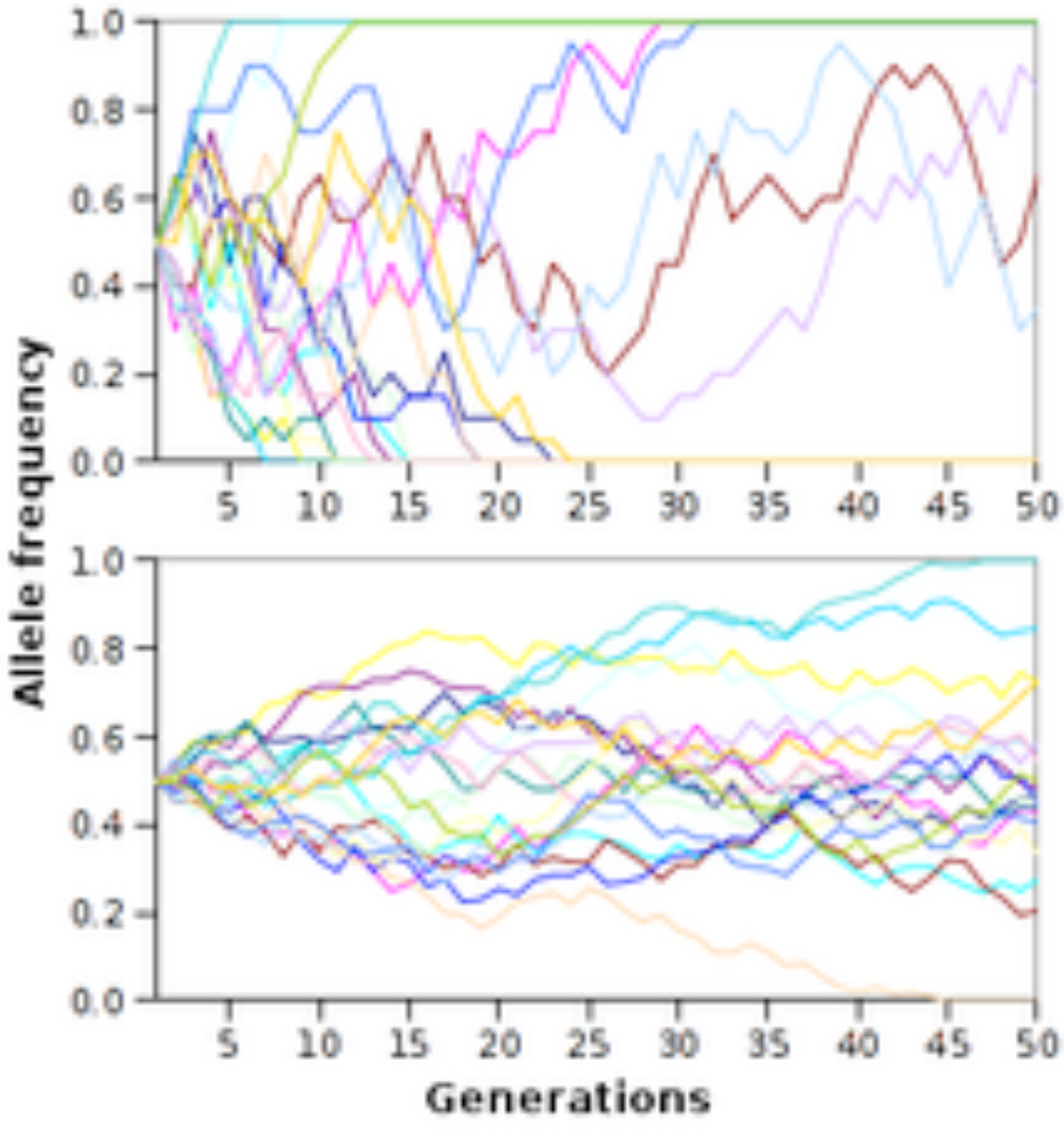
- But, be very, very, very careful here.



http://xkcd.com/534/

Genetic Algorithms

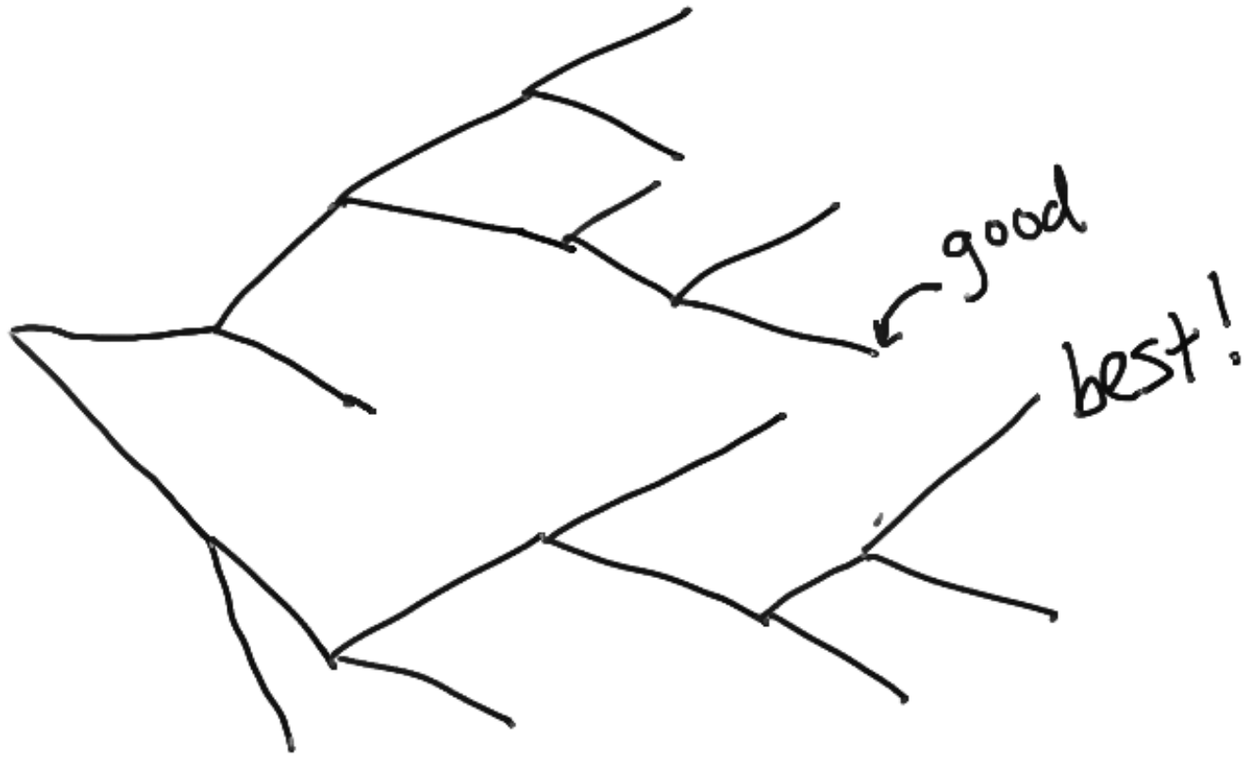
- All of the intricacies of biological evolution apply here :
  - Natural selection (what we want)
  - Genetic drift (which can occur in GA's and bio. evo.)



- Can't necessarily always find a solution!
- Also can't let it drift forever
- So, we also have to be practical and put cutoff conditions
  - Too many iterations or organisms

Genetic Algorithms

- Also just like real evolution, you are only able to deal with "reachable" mutations
  - A bacterium growing an arm will never happen in either bio. evo. or GA's.
- "Good" solutions are only relatively so given the previous history
  - There may be better ways to get to the end, but you are only given the immediate organisms to work with

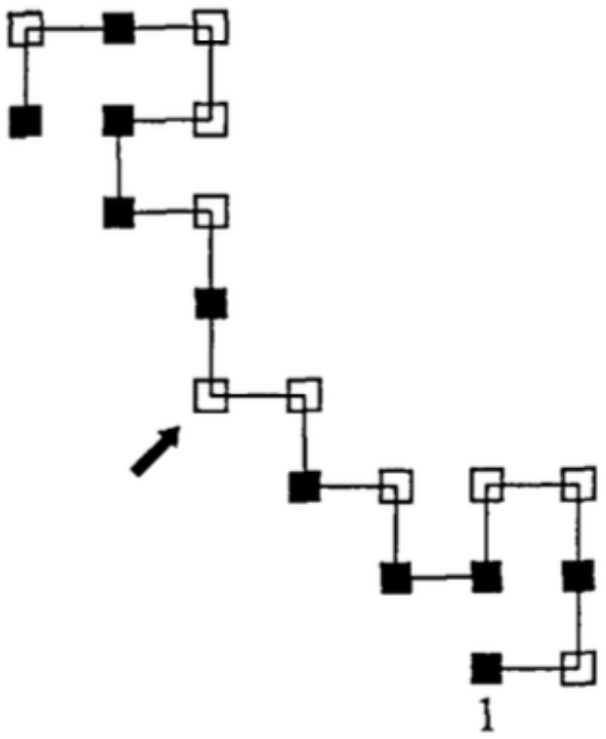


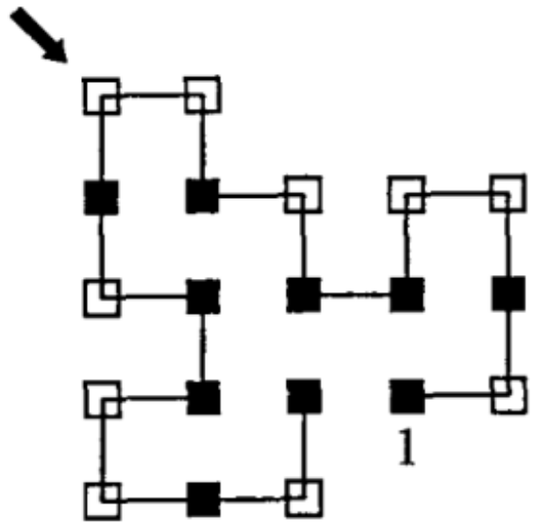
Genetic Algorithms

- Example from https://gist.github.com/bellbind/741853
- Basic algorithm :
  - Start : generate random population of n chromosomes
  - Fitness : evaluate fitness of each
  - Get a new population : Mutate or otherwise generate a new population
  - Replace the population
  - Test against desired accuracy / etc
  - Loop

Protein Folding

- Unger and Moult, “Genetic algorithms for protein folding simulations”, J. Mol. Biol. 231, 75-81 (1993)
- Looking for configurations of 2-d proteins hydrophobic (black) and hydrophilic (white) amino acids

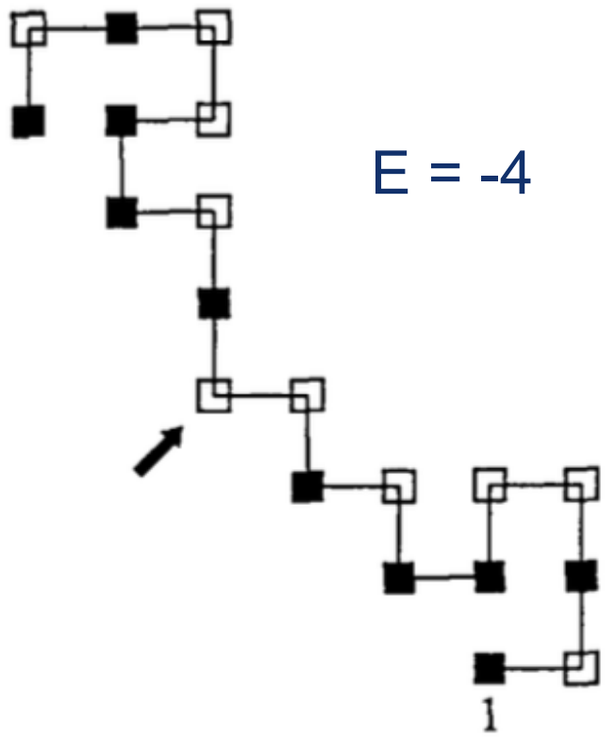




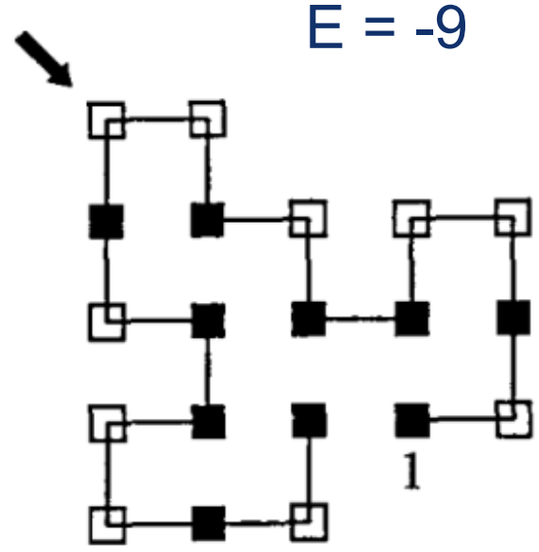
Protein Folding

- Energy function:

$$V = -\sum_{\langle B_i B_j \rangle} \Delta(\mathbf{r}_i - \mathbf{r}_j) \,, \qquad \Delta(\mathbf{r}) = \begin{cases} 1 & \text{if } |\mathbf{r}| = 1 \text{ and bond is not covalent} \\ 0 & \text{otherwise} \end{cases}$$



E = -4



E = -9

Protein Folding

- Simulated annealing (Metropolis MC):

- Start from straight line
- Trial: Select monomer at random, rotate
- If new config is self-avoiding, compute ΔE
  - If negative: accept
  - else: accept if $w = \exp\left[-\Delta E / c_k\right] > u\ , \qquad c_k \equiv k_{\mathrm{B}} T$
    - (u = random deviate)

- If stopping criterion is not met, repeat trial after reducing T
  - In original paper, start at T=2K, reduce by 1% every 200,000 steps to a minimum of 0.15K.

Protein Folding

- Genetic algorithm:

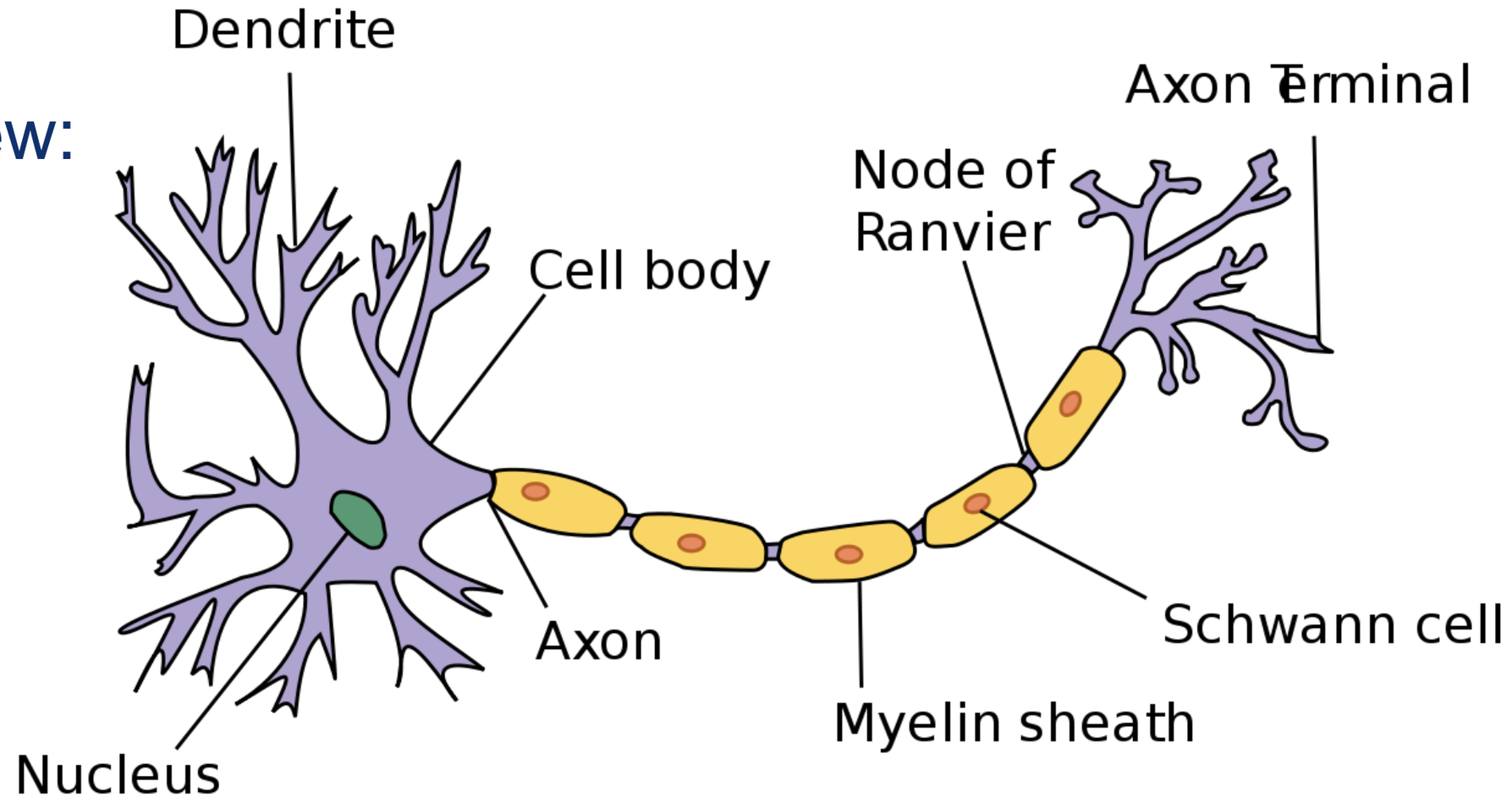
- Start from straight line
- Make N copies (mutants)
- Trial: For each mutant, select monomer at random, rotate
  - Accept if new config is self-avoiding and satisfies Metropolis Boltzmann condition:

$$w = \exp\left[-(E - \langle E \rangle)/c_k\right] > u\ , \qquad c_k \equiv k_{\mathrm{B}}T$$

- Compute best two mutants based on lowest energy
- Randomly select point, splice the two together
  - If avoiding: replace N copies with new mutant
- Repeat

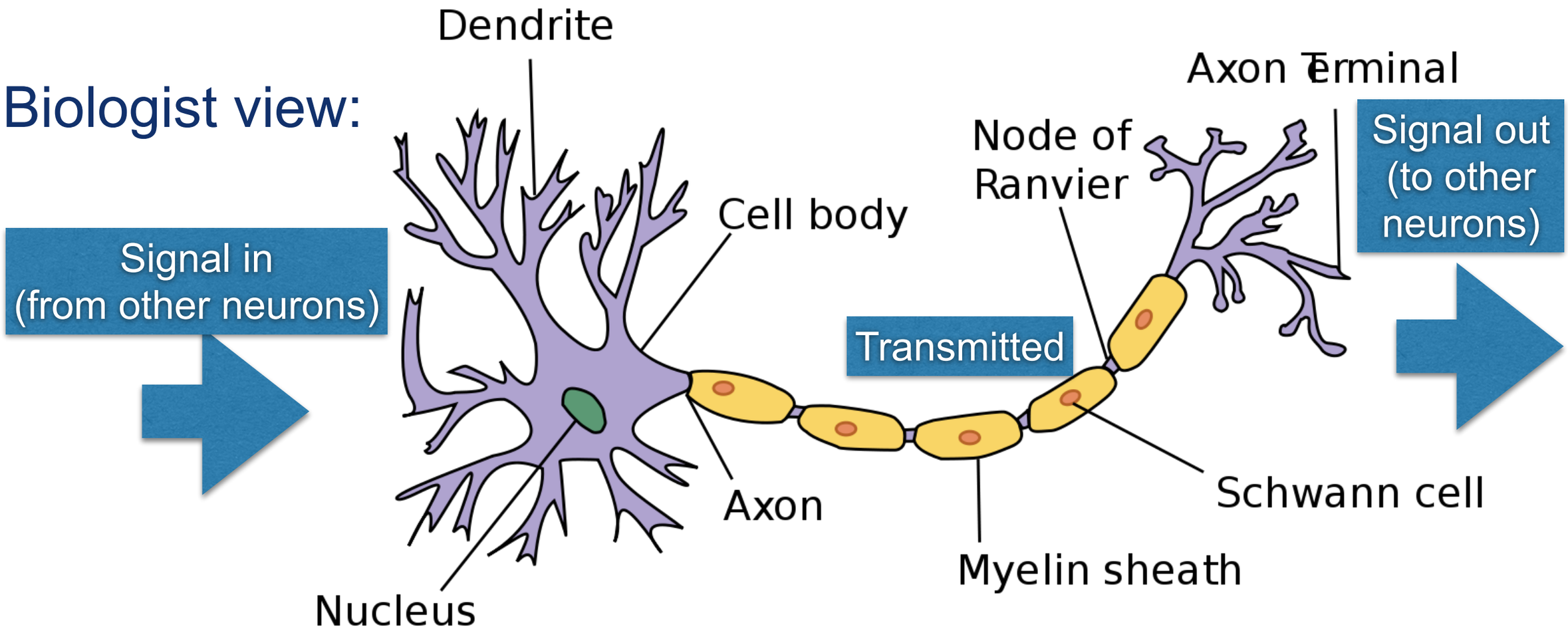
Neurons

- They are what make you, you!
- Form the central nervous system and brain
- ATP drives ion pumps to maintain ~ -70 mV voltage bias
- Excitations cause action potential across the axon
- Biologist view:



Neurons

- They are what make you, you!
- Form the central nervous system and brain
- ATP drives ion pumps to maintain ~ -70 mV voltage bias
- Excitations cause action potential across the axon
- Biologist view:



Neurons

- They are what make you, you!
- Form the central nervous system and brain
- ATP drives ion pumps to maintain ~ -70 mV voltage bias
- Excitations cause action potential across the axon

- Physicist view:

"Spherical cow" approach

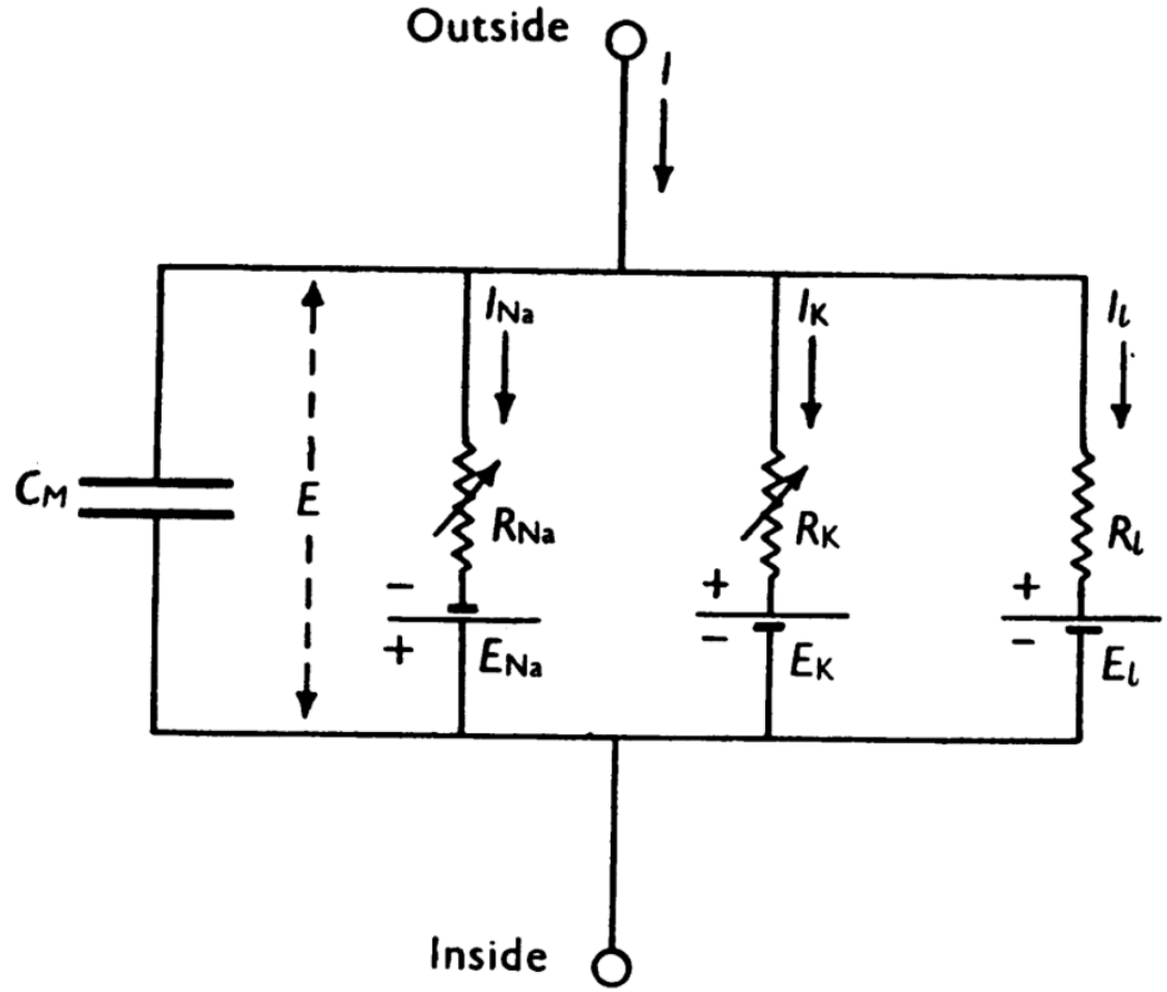
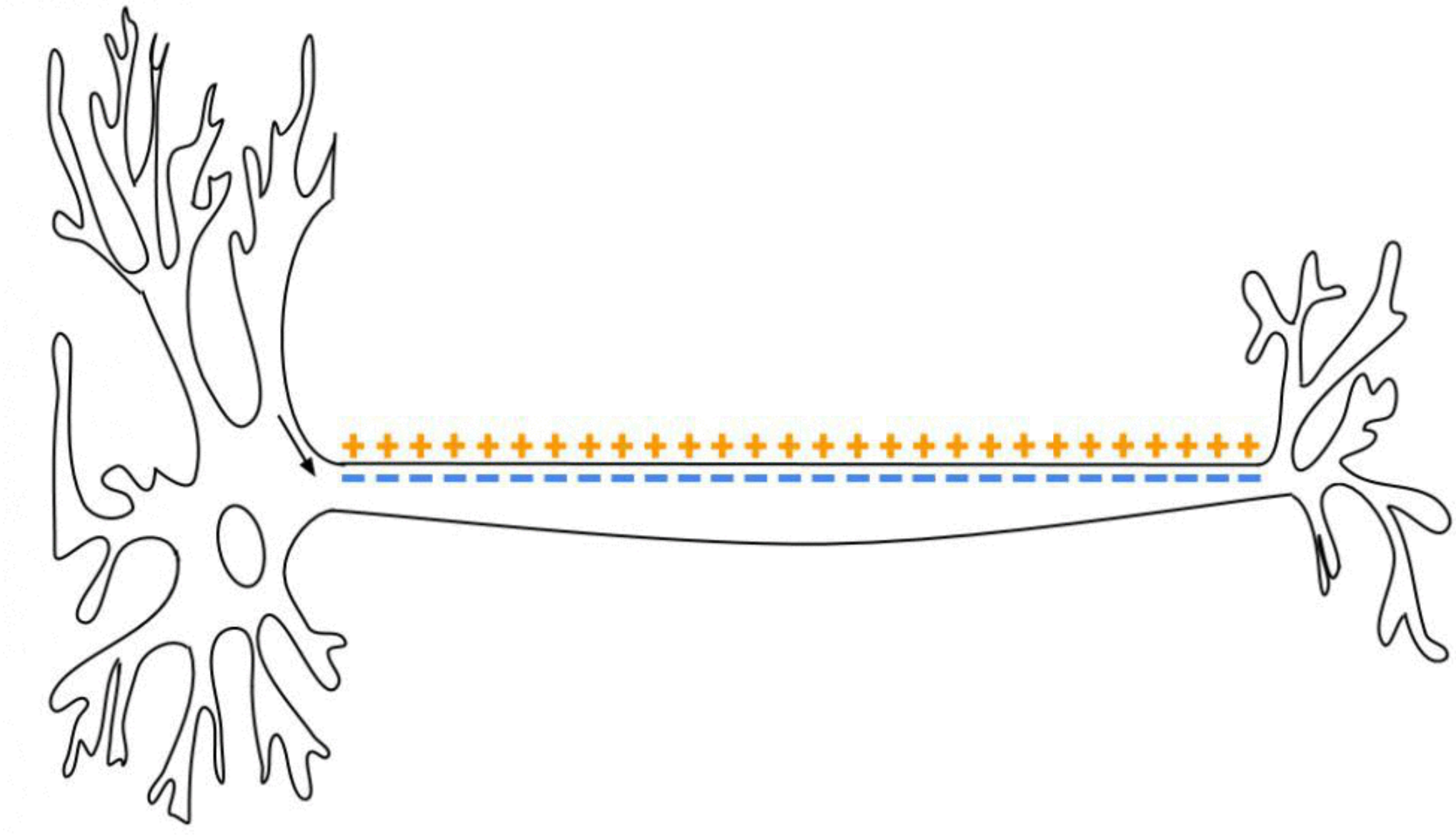


Fig. 1. Electrical circuit representing membrane. $R_{Na}=1/g_{Na}$; $R_K=1/g_K$; $R_l=1/\bar{g}_l$. R_{Na} and R_K vary with time and membrane potential; the other components are constant.

Neurons

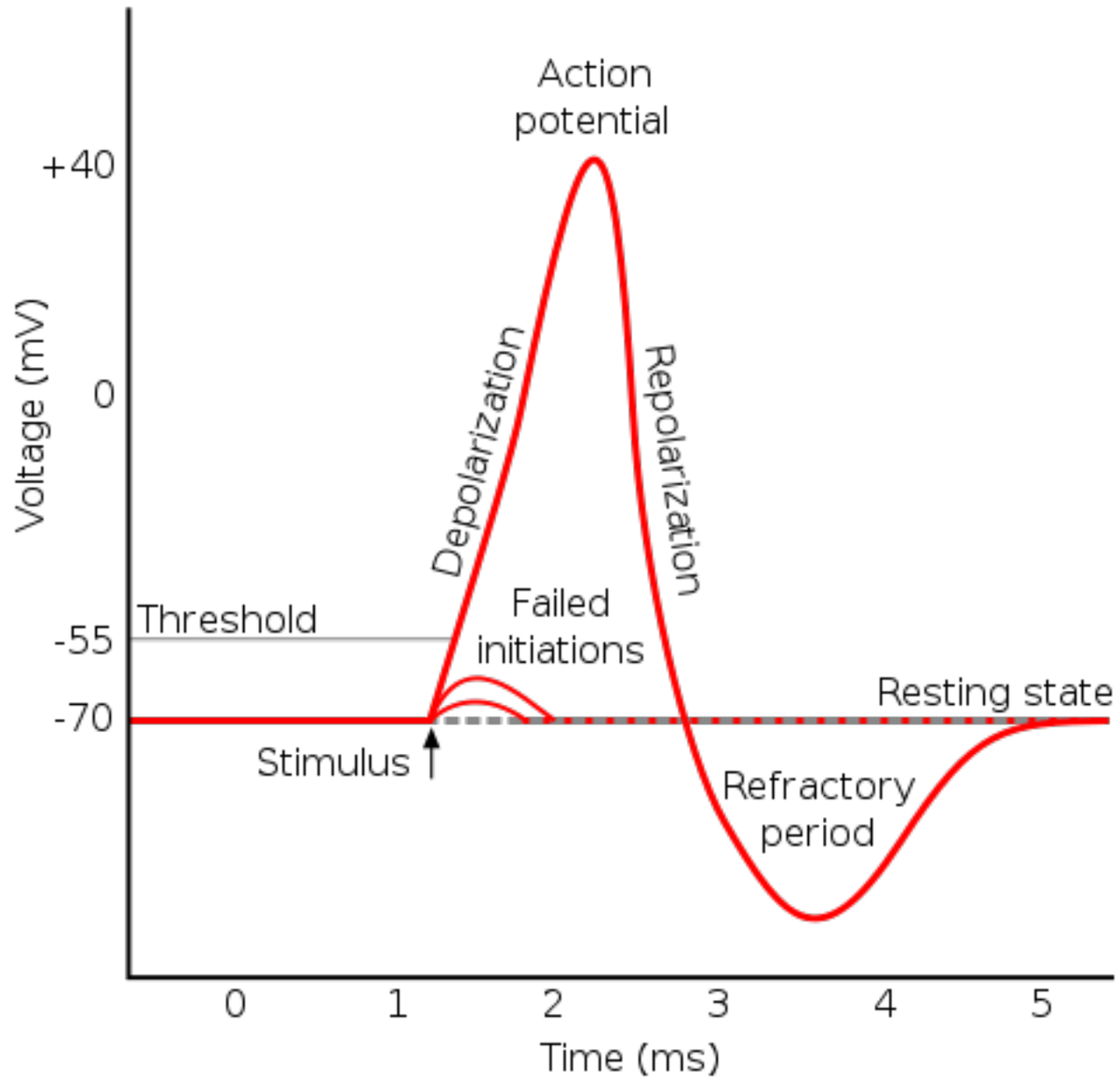
- Action potential



https://en.wikipedia.org/wiki/Action_potential

Neurons

- Above some threshold voltage, potential grows to a pulse that is propagated as a soliton
- Has a “cooloff” refractory period
- Returns to steady state



Hodgkin-Huxley Equations

- Derived in 1952 by two physician / biophysicists :
  J. Physiol. 117(4), 500-544 (1952)
- Phenomenological equations for the current and potentials:

$$I = C_M \frac{dV}{dt} + \bar{g}_{\mathsf{K}} n^4 \left(V - V_{\mathsf{K}}\right) + \bar{g}_{\mathsf{Na}} m^3 h \left(V - V_{\mathsf{Na}}\right) + \bar{g}_l \left(V - V_l\right)$$

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

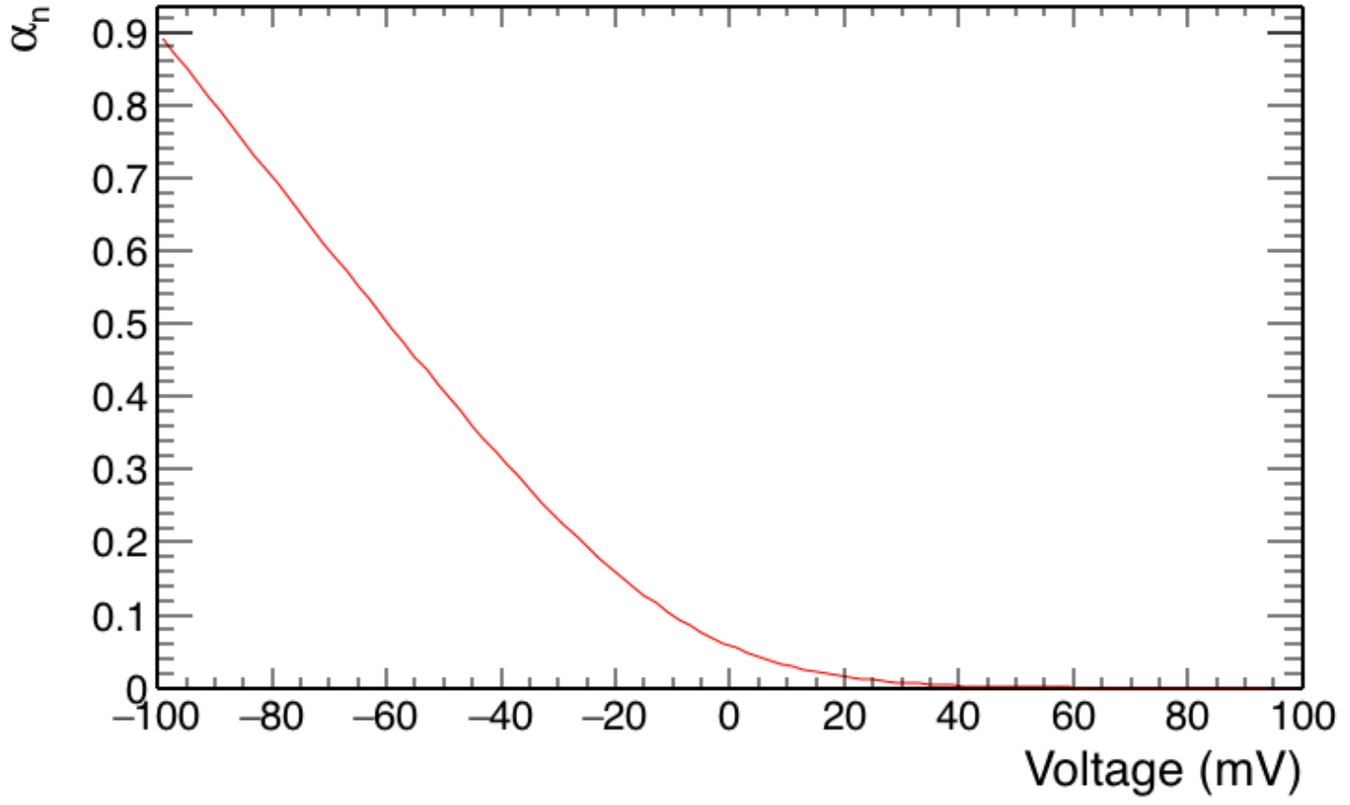
$$\frac{dm}{dt} = \alpha_m (1 - m) - \beta_m m$$

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

$$\alpha_n = \frac{0.01(V + 10)}{\exp\left[\frac{V+10}{10}\right] - 1} \qquad \beta_n = 0.125 \exp\left(\frac{V}{80}\right)$$

$$\alpha_m = \frac{0.1(V + 25)}{\exp\left[\frac{V+25}{10}\right] - 1} \qquad \beta_m = 4 \exp\left(\frac{V}{18}\right)$$

$$\alpha_h = 0.07 \exp\left(\frac{V}{20}\right) \qquad \beta_h = \frac{1}{\exp\left[\frac{V+30}{10}\right] + 1}$$



Hodgkin-Huxley Equations

- Experimentally measured values:

TABLE 3

Constant (1)	Value chosen (2)	Experimental values: Mean (3)	Experimental values: Range (4)	Reference (5)
C_M (μF/cm^2)	1·0	0·91	0·8 to 1·5	Table 1, Hodgkin *et al.* (1952)
V_{Na} (mV)	−115	−109	−95 to −119	p. 455, Hodgkin & Huxley (1952*a*)
V_K (mV)	+ 12	+ 11	+ 9 to + 14	Table 3, values for low temperature in sea water, Hodgkin & Huxley (1952*b*)
V_l (mV)	−10·613*	− 11	− 4 to − 22	Table 5, Hodgkin & Huxley (1952*b*)
$\bar{g}_{Na}$ (m.mho/cm^2)	120	80	65 to 90	Fully analysed results, Table 2† } Hodgkin & Huxley (1952*a*)
		160	120 to 260	Fresh fibres, p. 465† } Hodgkin & Huxley (1952*a*)
$\bar{g}_K$ (m.mho/cm^2)	36	34	26 to 49	p. 463, Hodgkin & Huxley (1952*a*)
$\bar{g}_l$ (m.mho/cm^2)	0·3	0·26	0·13 to 0·50	Table 5, Hodgkin & Huxley (1952*b*)

* Exact value chosen to make the total ionic current zero at the resting potential ($V = 0$).

† The experimental values for $\bar{g}_{Na}$ were obtained by multiplying the peak sodium conductances by factors derived from the values chosen for α_m, β_m, α_h, and β_h.

Hodgkin-Huxley Equations

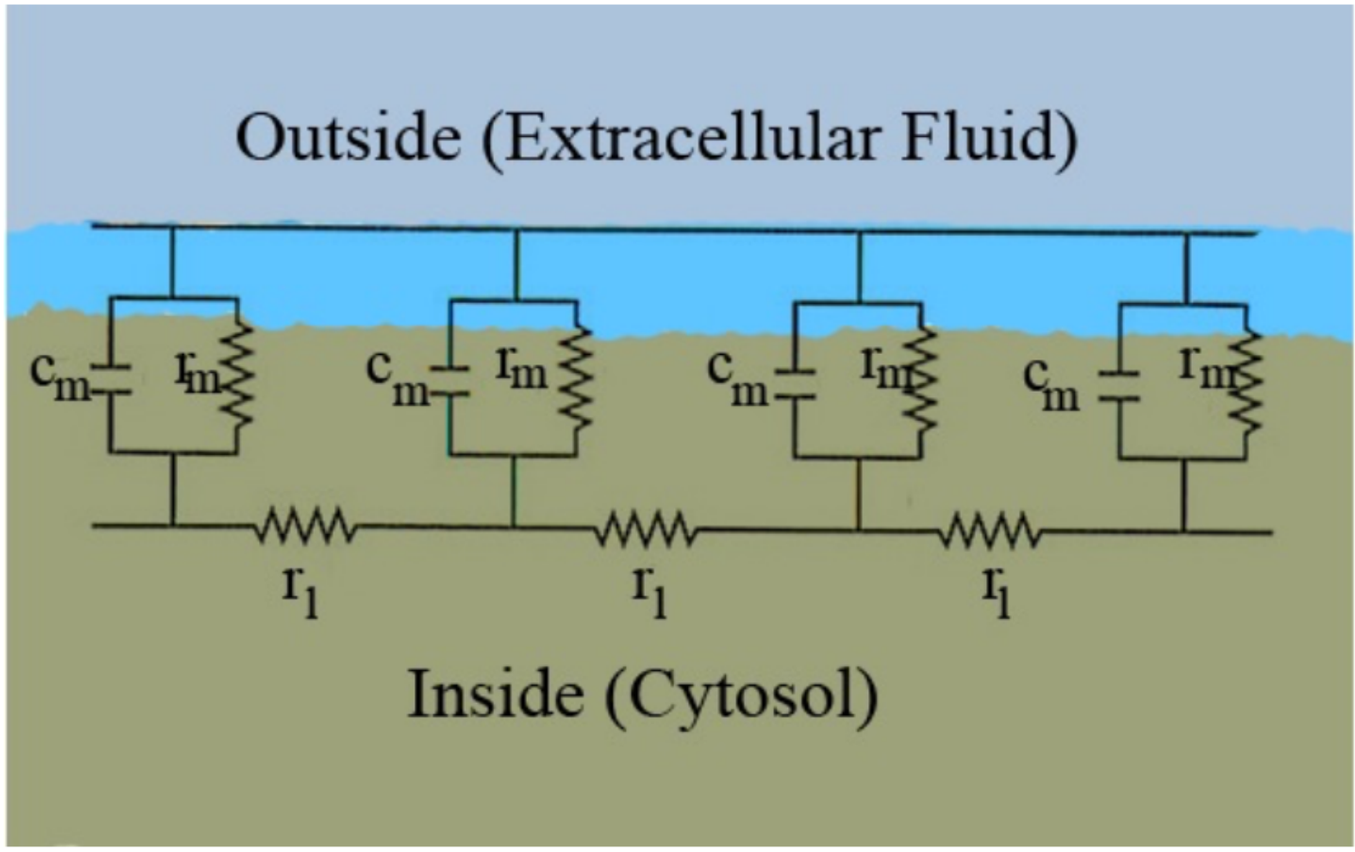
- Two cases solved in HH paper
  - Constant uniform membrane potential
  - Propagated action potential

Hodgkin-Huxley Equations

- Constant uniform membrane potential
  - Inserted a wire through length of axon, held at constant V
  - No current along cylinder axis, so I = 0 except during stimulus (short shock ~ delta function at t=0)
  - Equation has I = 0, V = V0 and m,n,h are at steady state resting values

Hodgkin-Huxley Equations

- Propagated action potential
  - Axons in living organism excited at junction with cell body
  - Impulse from excitation propagates down length
  - Simulate this with HH circuit elements connected in series by longitudinal resistors
  - Continuum limit:



$$\frac{a}{2R_2}\frac{\partial^2 V}{\partial x^2} = C_M \frac{\partial V}{\partial t} + \bar{g}_{\mathsf{K}} n^4 \left(V - V_{\mathsf{K}}\right) + \bar{g}_{\mathsf{Na}} m^3 h \left(V - V_{\mathsf{Na}}\right) + \bar{g}_l \left(V - V_l\right)$$

Length along axon

Hodgkin-Huxley Equations

- There is a soliton solution to this diff. eq.:

$$\frac{\partial^2 V}{\partial x^2} = \frac{1}{\theta^2}\frac{\partial^2 V}{\partial t^2}$$

Propagation speed

- Assuming the shape is independent of x (i.e. wave propagation) it becomes an ODE:

$$\frac{a}{2R_2\theta^2}\frac{d^2V}{dt^2} = C_M\frac{dV}{dt} + \bar{g}_{\mathsf{K}}n^4\left(V - V_{\mathsf{K}}\right) + \bar{g}_{\mathsf{Na}}m^3h\left(V - V_{\mathsf{Na}}\right) + \bar{g}_l\left(V - V_l\right)$$

  - H+H found that they could measure theta by requiring V tend to zero as time goes on (bad solutions go to +- infinity)
    - Used root finder to solve this

Hodgkin-Huxley Equations

- Going to reproduce Fig 12 in their paper
  - (first case with I = 0)