University of Washington Computer Science & Engineering

CSE590b: Molecular and neural computation

Georg Seelig

Administrative

Georg Seelig
Office: CSE 228
gseelig@uw.edu

TA: Kevin Oishi

koishi@uw.edu

Grading:
30% class participation: ask questions. It's more fun if it's interactive.

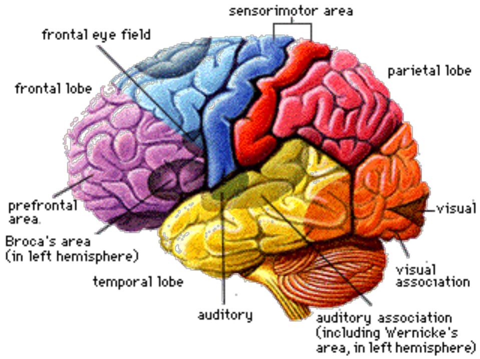
30% Homework: Due at the end of class one week after they are handed out. Late policy: -10% each day for the first 3 days, then not accepted

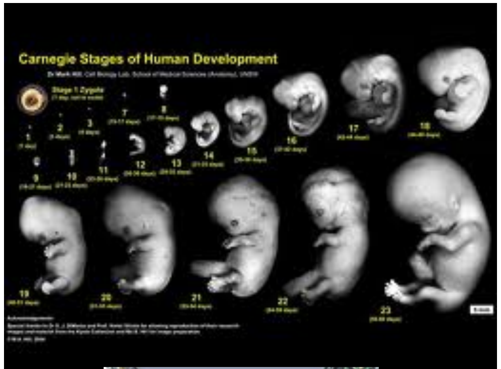
40% Class project: A small design project using one (or several) of the design and simulation tools that will be introduced in class

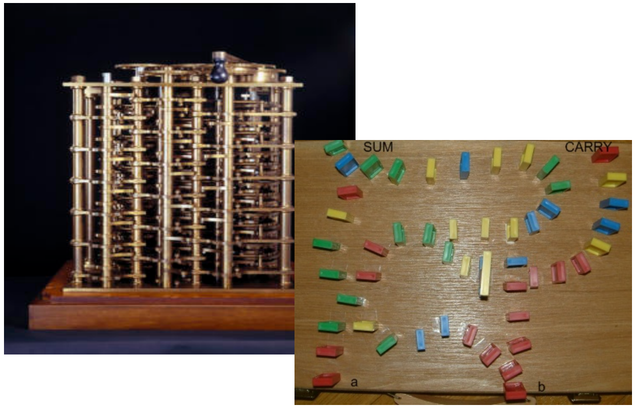
Books: There is no single book that covers the material in this class. Any book on molecular biology and on neural computation might be helpful to dig deeper.

https://www.coursera.org/course/compneuro

Computation can be embedded in many substrates







Alternative physical substrates can be used to make computers

Computation controls physical substrates (output of computation is the physical substrate)

The molecular programming project

The history of computing has taught us two things: first, that the principles of computing can be embodied in a wide variety of physical substrates from gears and springs to transistors, and second that the mastery of a new physical substrate for computing has the potential to transform technology.

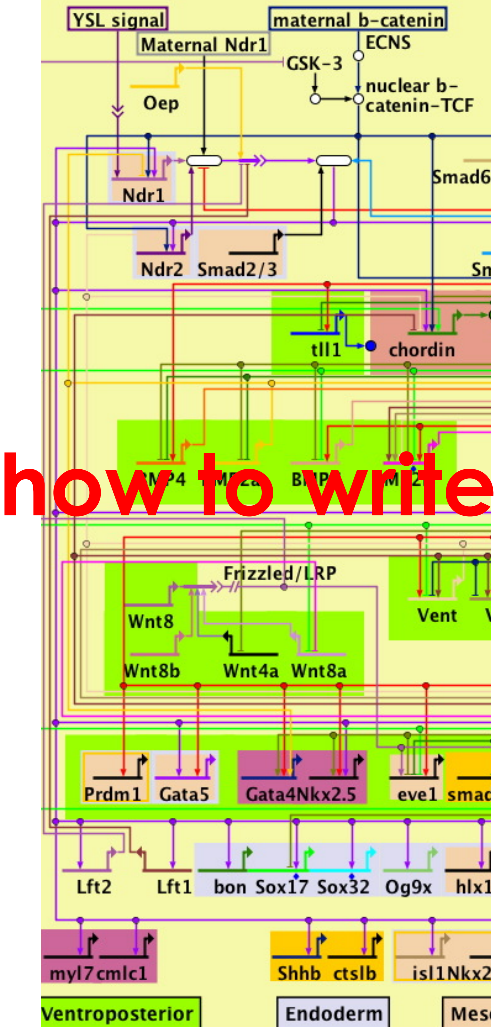
Another revolution is just beginning, one that will result in new types of programmable systems based on molecules. Like the previous revolutions, this “molecular programming revolution” will take much of its theory from computer science, but will require reformulation of familiar concepts such as programming languages and compilers, data structures and algorithms, resources and complexity, concurrency and stochasticity, correctness and robustness. With molecular programming, chemistry will become the new information technology of the 21st century.

Biological inspiration

DNA Genome

```
...GTGGTACAGGTG
AATTTGGGTAGGCTA
AATTGTCCATAGTTT
ATGTGTGTGAATGAG
GGTGTATGGATGTTT
CTCAGAGATGGGTTG
CAGCTGGAAGGGCGT
CCATTGTGGAAAACA
TATGCTGGAGAAGTT
GCCGGTTCATTCTGC
TGTGGCGACCCCAGA
TTAATAAAAGGACTA
AGCCGAAAAGAAAAT
GAAACATATATATAT
ATATATATATATATA
TATATATATA...
```

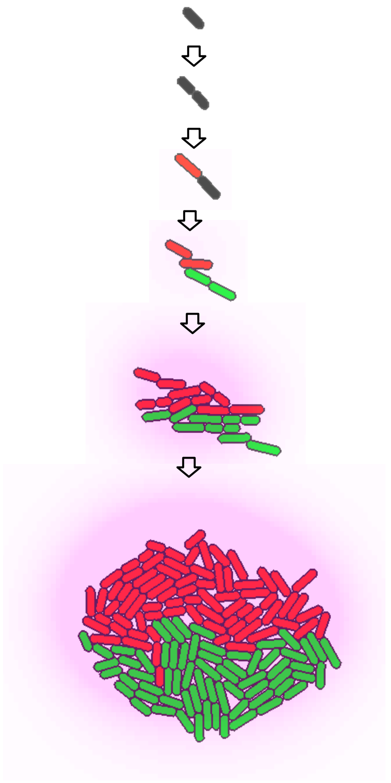
Regulatory Circuitry



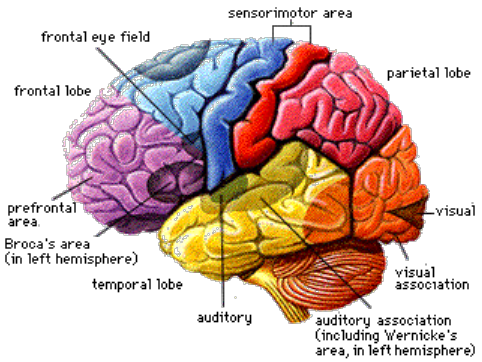
Zebrafish Development

Can we learn how to write such a program?

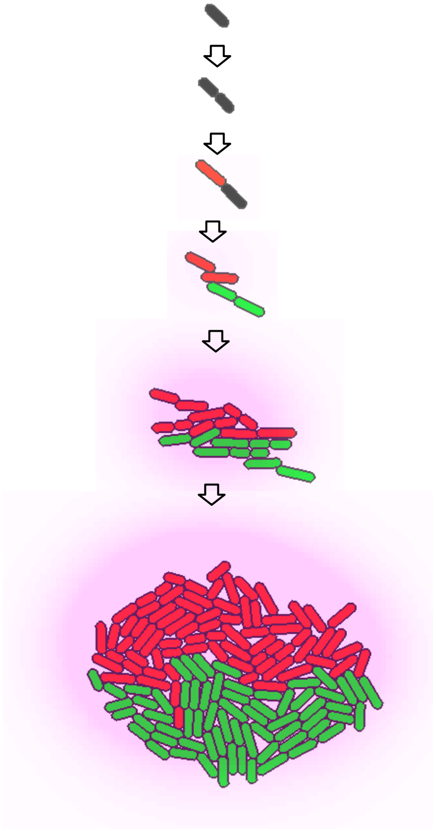
Motivating question for this class



What abstractions, architectures and programming languages are appropriate for specifying spatial organization and differentiation of cells/matter such that we can synthesize a system as complex as the brain?



What I cannot create I do not understand (R. Feynman)

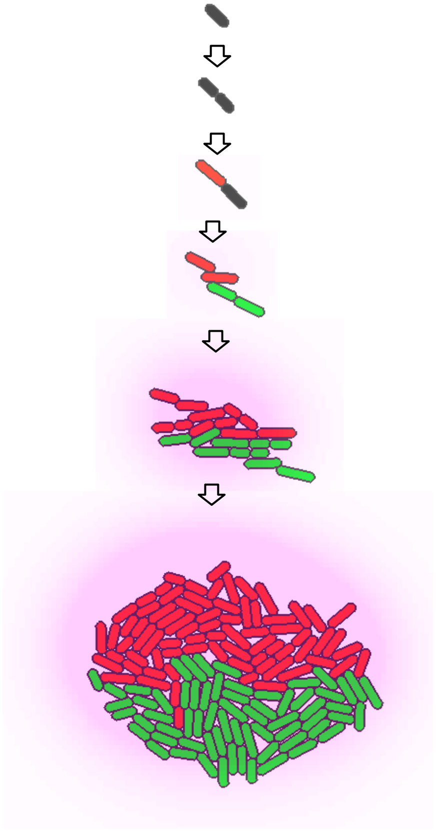


We're taking a forward engineering/rational design approach to understanding biological computation.

Biological systems are evolved and may not provide an ideal blueprint for engineering. We still have no idea how to read/write biological "code."

This is not a developmental biology class and also not a computational biology class

Models of computation



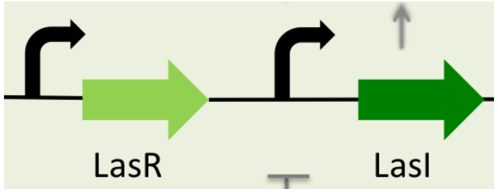
- Cellular automata
- Register machines
- Chemical reaction networks (analog circuits)
- Digital logic circuits
- Finite state machines
- Neural networks
- …

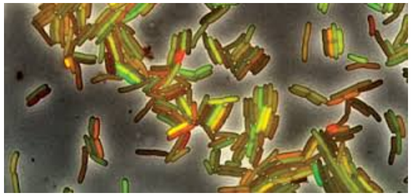
From molecules to the brain

Topic 1: DNA nanotechnology and molecular programming: The basic modules are programmable biomolecules

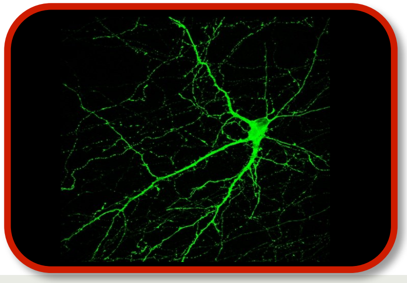
```
DNA:  GTGGTACAGGTG
RNA:  GGGCUGUUUUUC
Prot:MTYRLELNGKTL
```

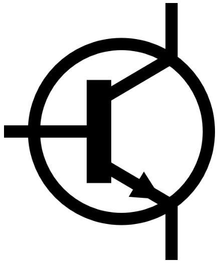
Topic 2: Synthetic biology (gene circuit engineering): The basic building blocks are genes and their products



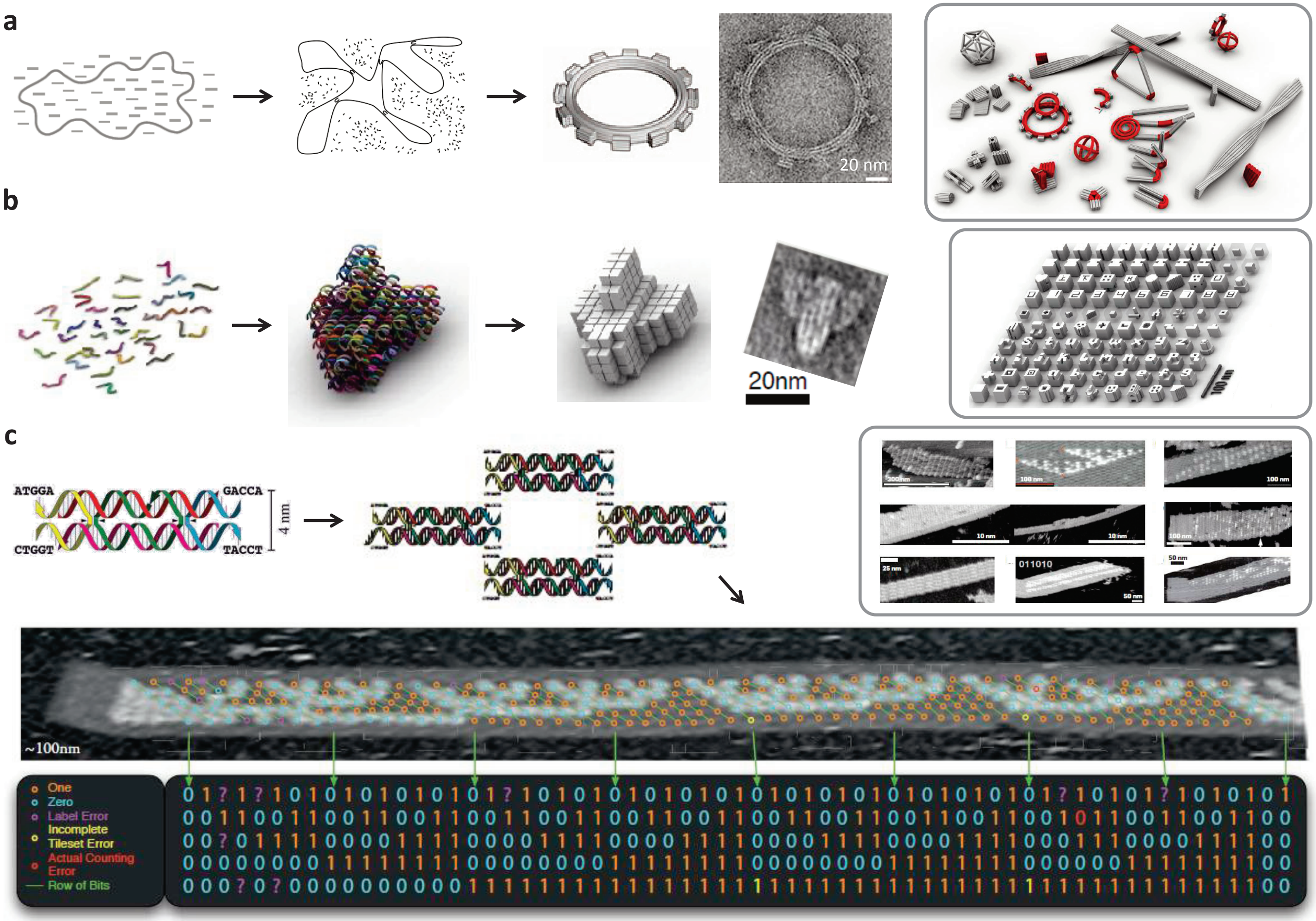


Topic 3: Computation in the brain: The basic building blocks are cells





Topic 1: Molecular programming

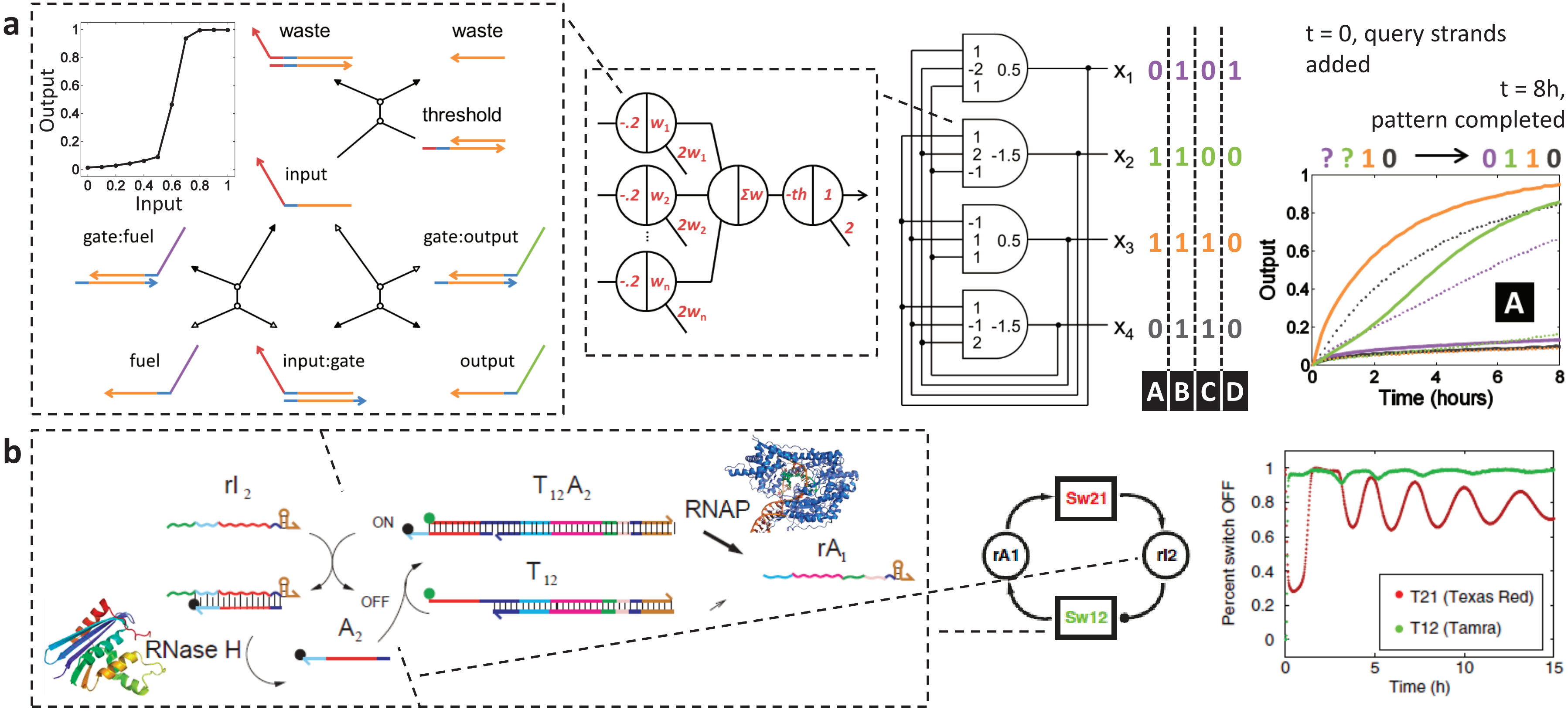


○ One
○ Zero
○ Label Error
○ Incomplete
○ Tileset Error
○ Actual Counting Error
— Row of Bits

```
0 1 ? 1 ? 1 0 1 0 1 0 1 0 1 0 1 0 1 ? 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 ? 1 0 1 0 1 ? 1 0 1 0 1 0 1
0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0
0 0 ? 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 ? 0 ? 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
```

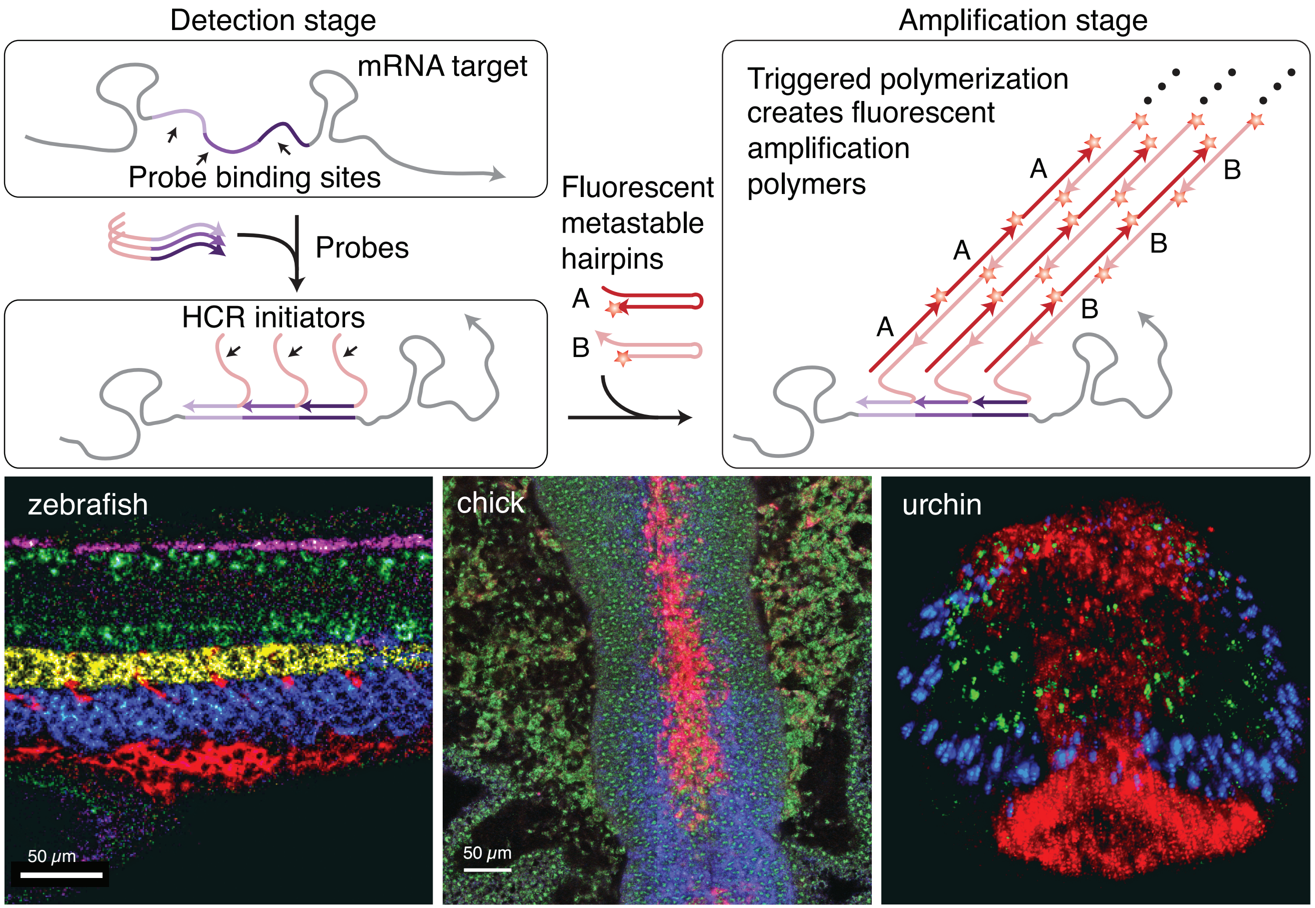
Structural DNA nanotechnology
(images by Rothemund, Shih, Winfree, Yin and others)

Topic 1: Molecular programming



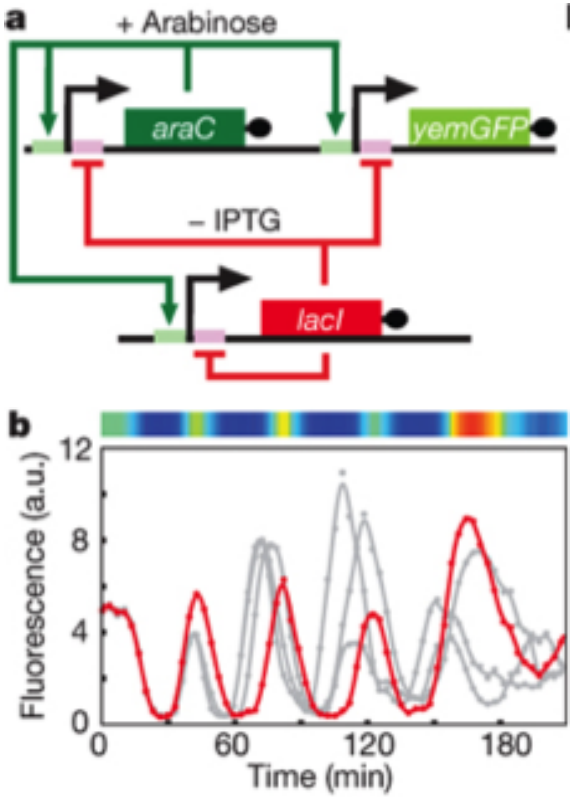
Logic circuits and dynamical systems
(images by Qian, Winfree and others)

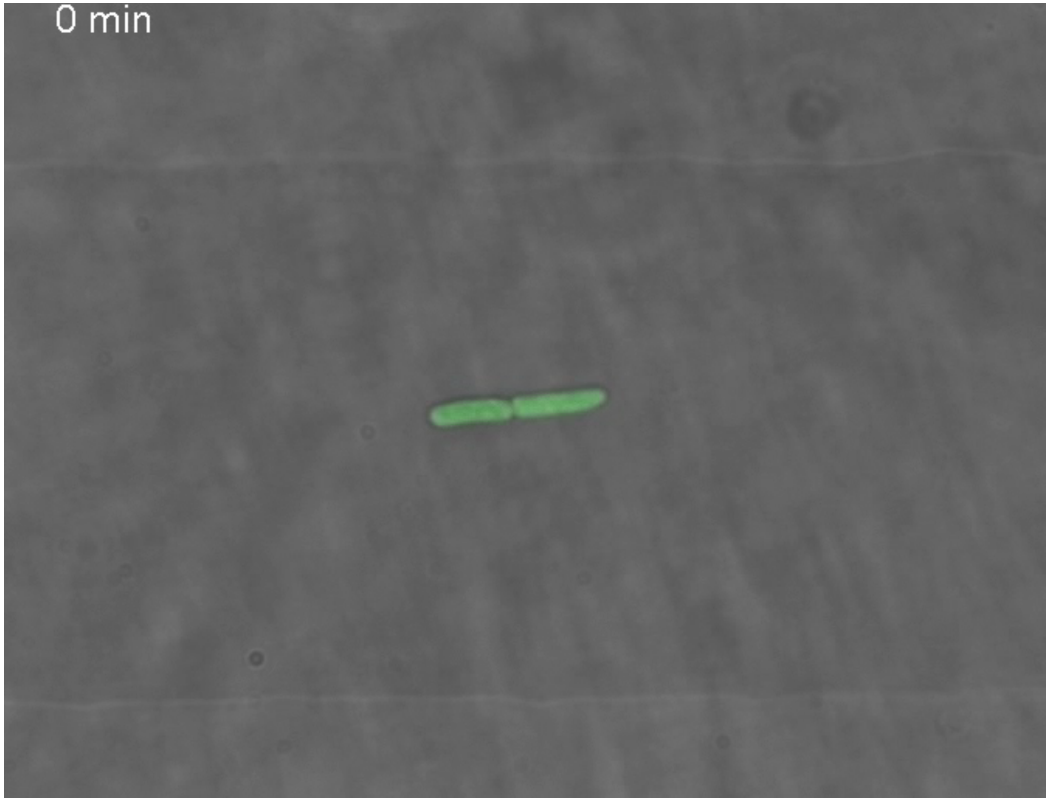
Topic 1: Molecular programming



Imaging application
(images by Pierce and Fraser)

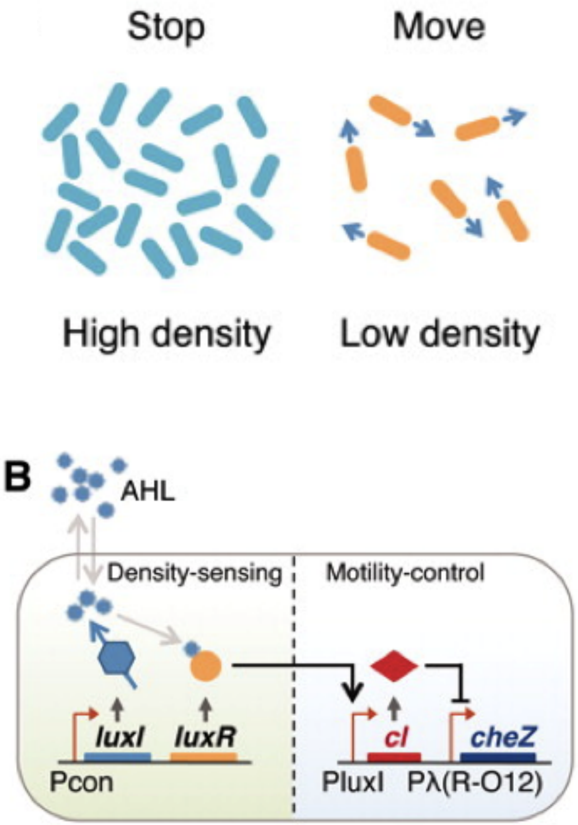
Topic 2: Synthetic biology

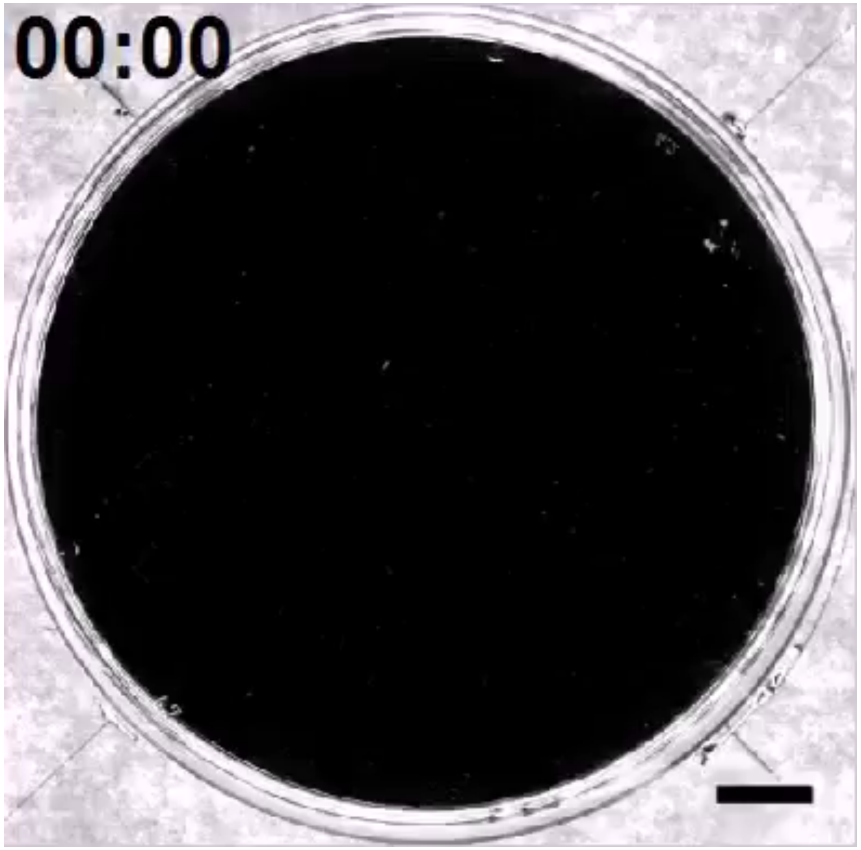




A biological oscillator
(Hasty lab)

Topic 2: Synthetic biology

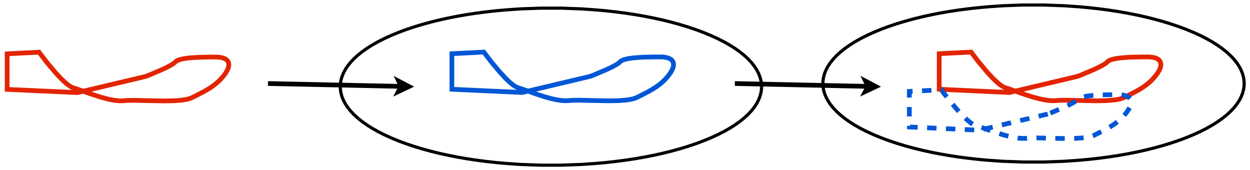




A synthetic pattern forming system
(lab)

Topic 2: Synthetic biology

Installing a synthetic genome



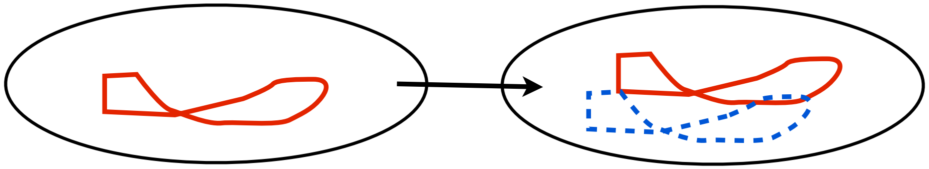
Synthetic genome
“new operating system”

Host cell

Install

Step 1: Complete synthesis of the genome from *Mycoplasma genitalium,* 580K basepairs

Step 2: Genome transfer from one cell to another (similar) cell



Mycoplasma mycoides capri

Mycoplasma capricolum

Craig Venter
Hamilton Smith
and others

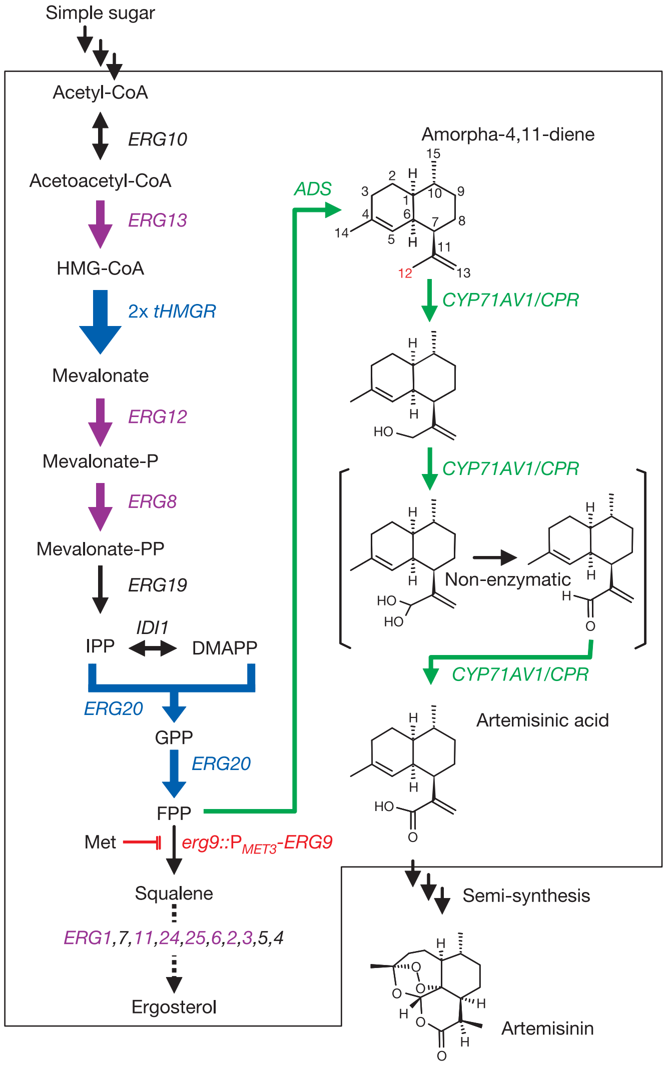
Topic 2: Synthetic biology

Application: Artemisinin synthesis (Keasling lab,)

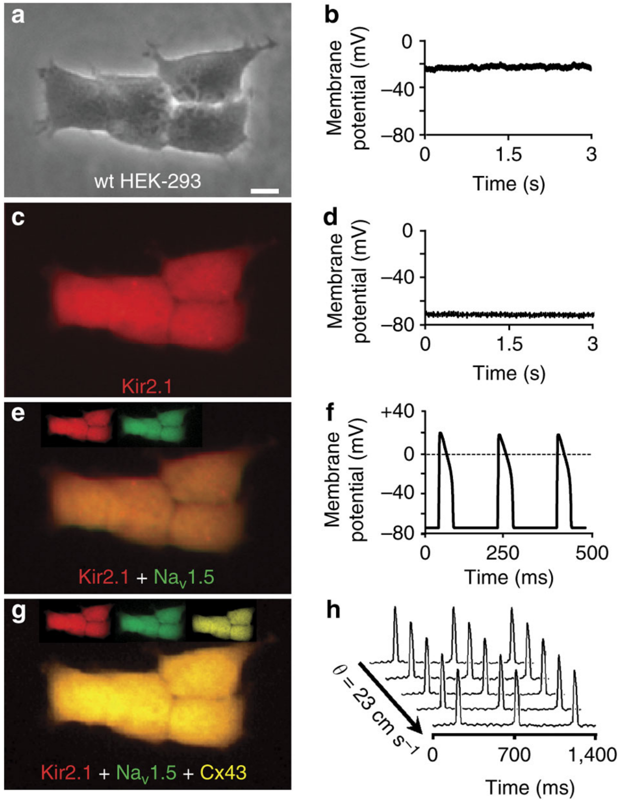
Artemisinin is an anti-malarial drug that is difficult to synthesize chemically. Can we make it through biotechnology?

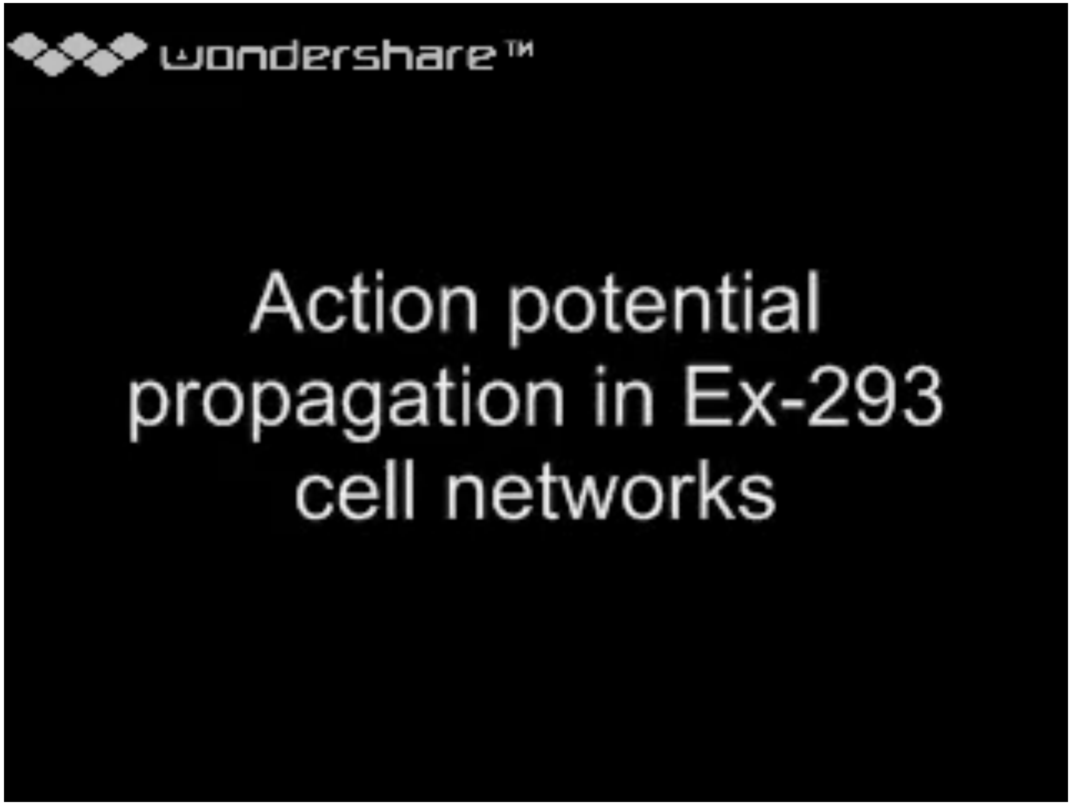
Challenges:

- Efficiently use resources
- Potentially toxic intermediates



Topic 2: Synthetic biology



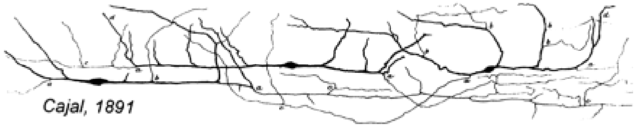


Action potential propagation in Ex-293 cell networks

Topic 3: Neural computation

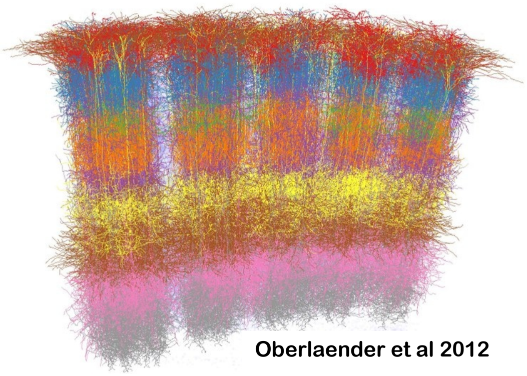
The brain is organized at multiple scales

Neurons



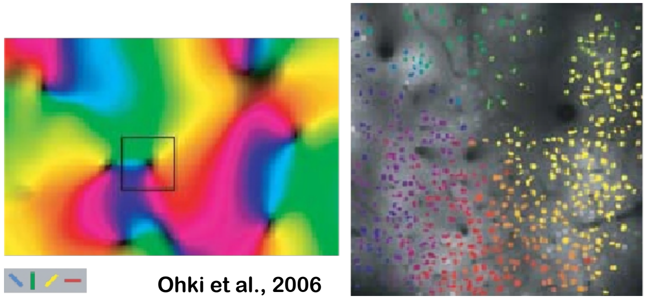
Cajal, 1891

Columns



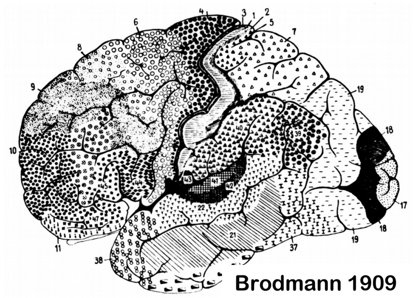
Oberlaender et al 2012

Maps



Ohki et al., 2006

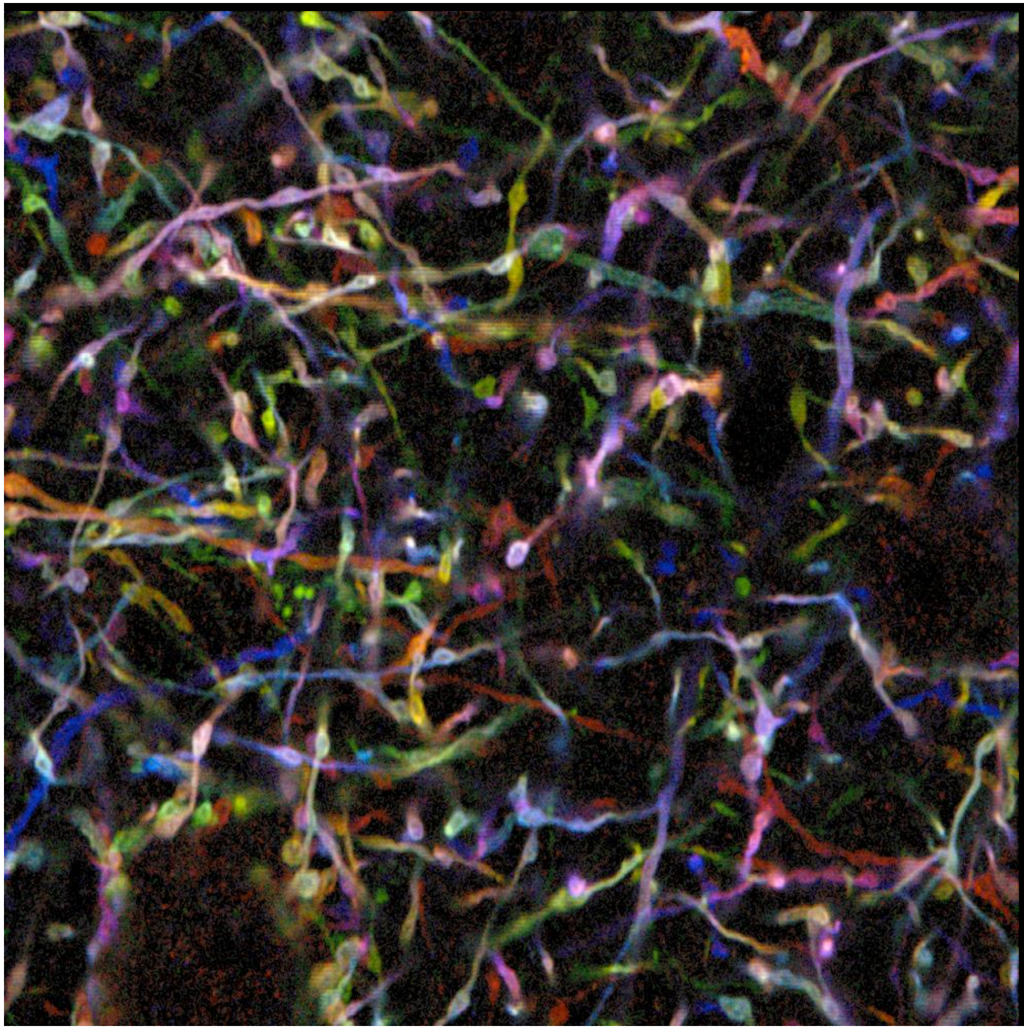
Areas



Brodmann 1909

Jack Gallant, Berkeley

Topic 3: Neural computation



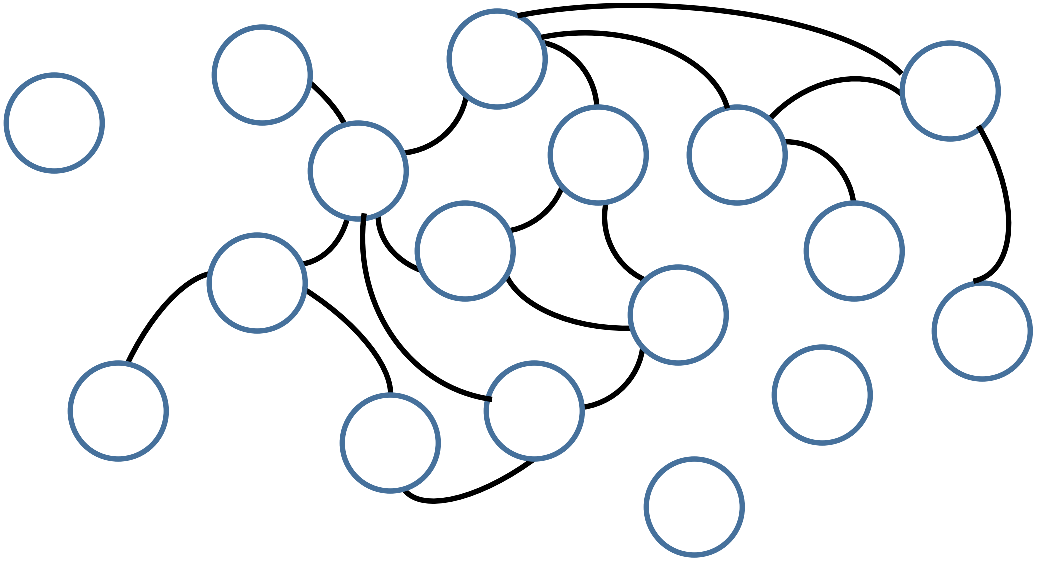
Cerebral cortex inhibitory axons only via infection with a brainbow virus

But what if we want to see *all* the connections?

Dawen Cai et al., 2013 &
Luke Bogart, Takao Hensch

Jeff Lichtman, Harvard

Topic 3: Neural computation



Understanding computation in neuronal networks. Toward a mechanistic understanding of the biology computation

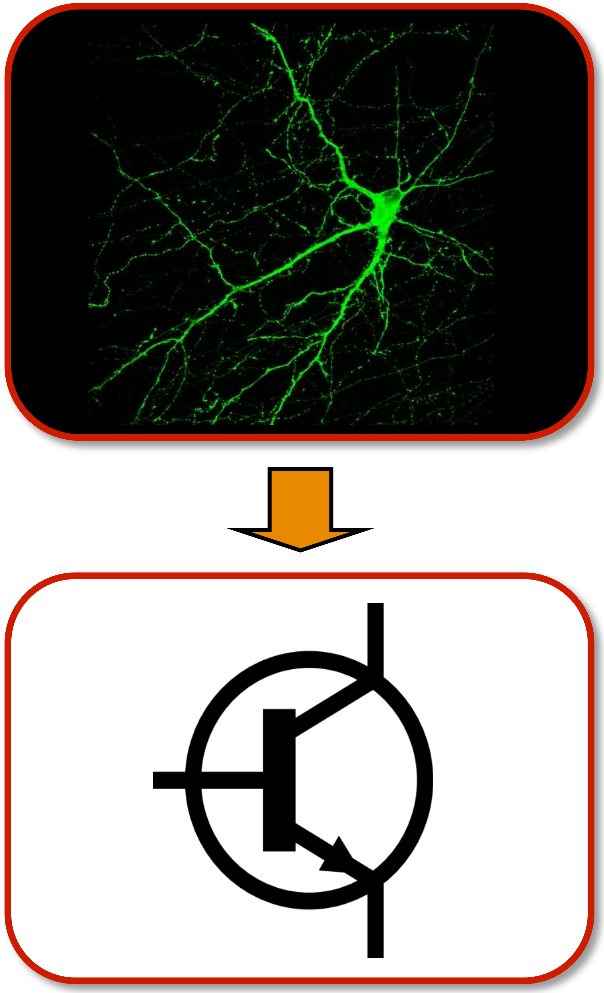
- What do different neuronal types contribute to computation?

- How do we go from neuronal details to global information processing and network function?

Adrienne Fairhall, UW

Topic 3: Neural computation

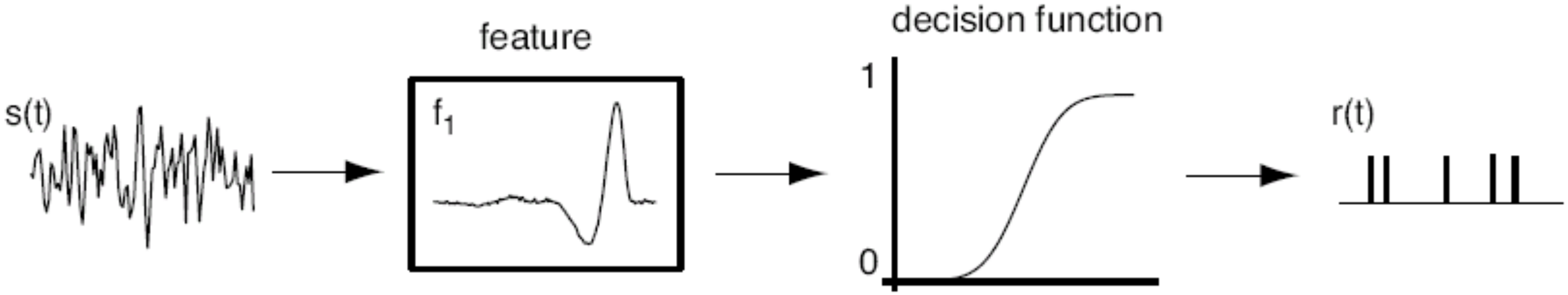
Characterizing neuronal computation



Adrienne Fairhall, UW

Topic 3: Neural computation

What can neurons compute? The dynamical systems properties of neurons lead to a range of different computational properties.



- Integrators or differentiators or fractional differentiators!
- Resonant at certain frequency band
- Distinct representations of information at different timescales
- Modulatable between different modes of operation
- Long timescale changes in operation

Computational character depends on input statistics   Adrienne Fairhall, UW