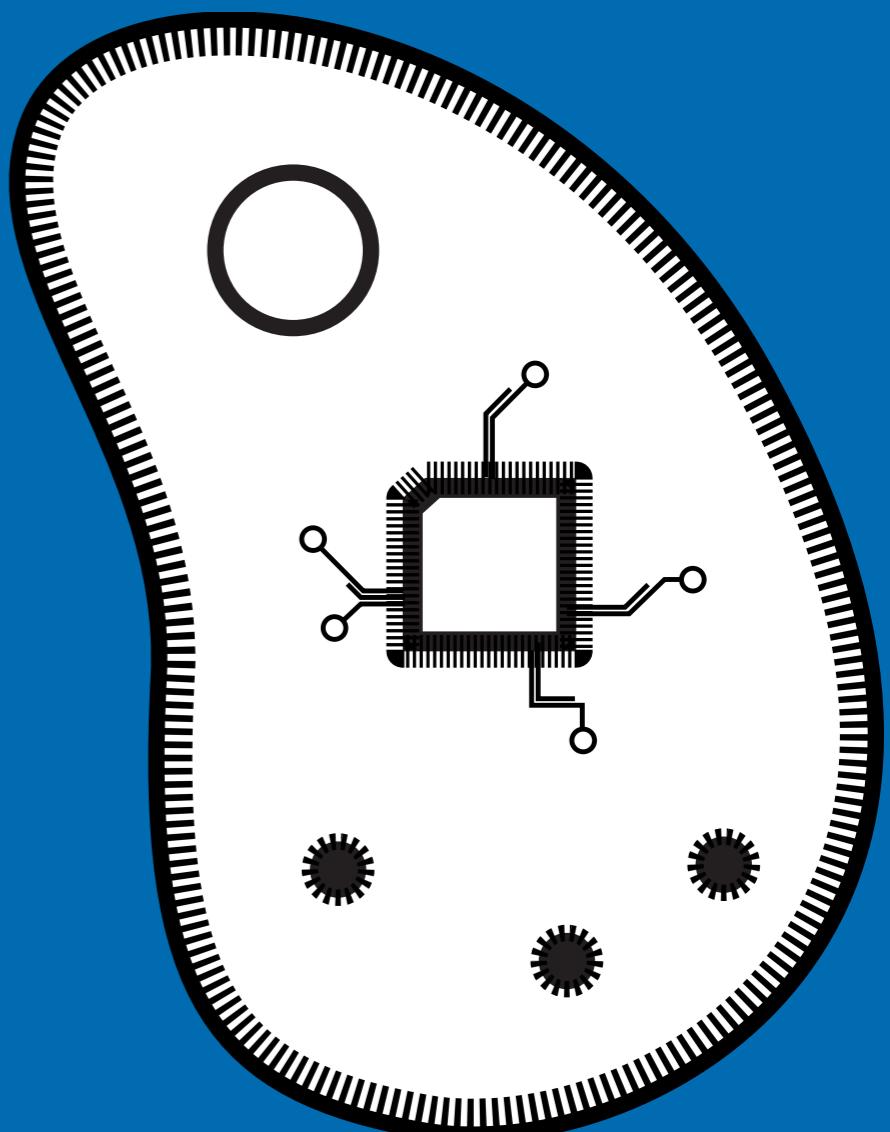


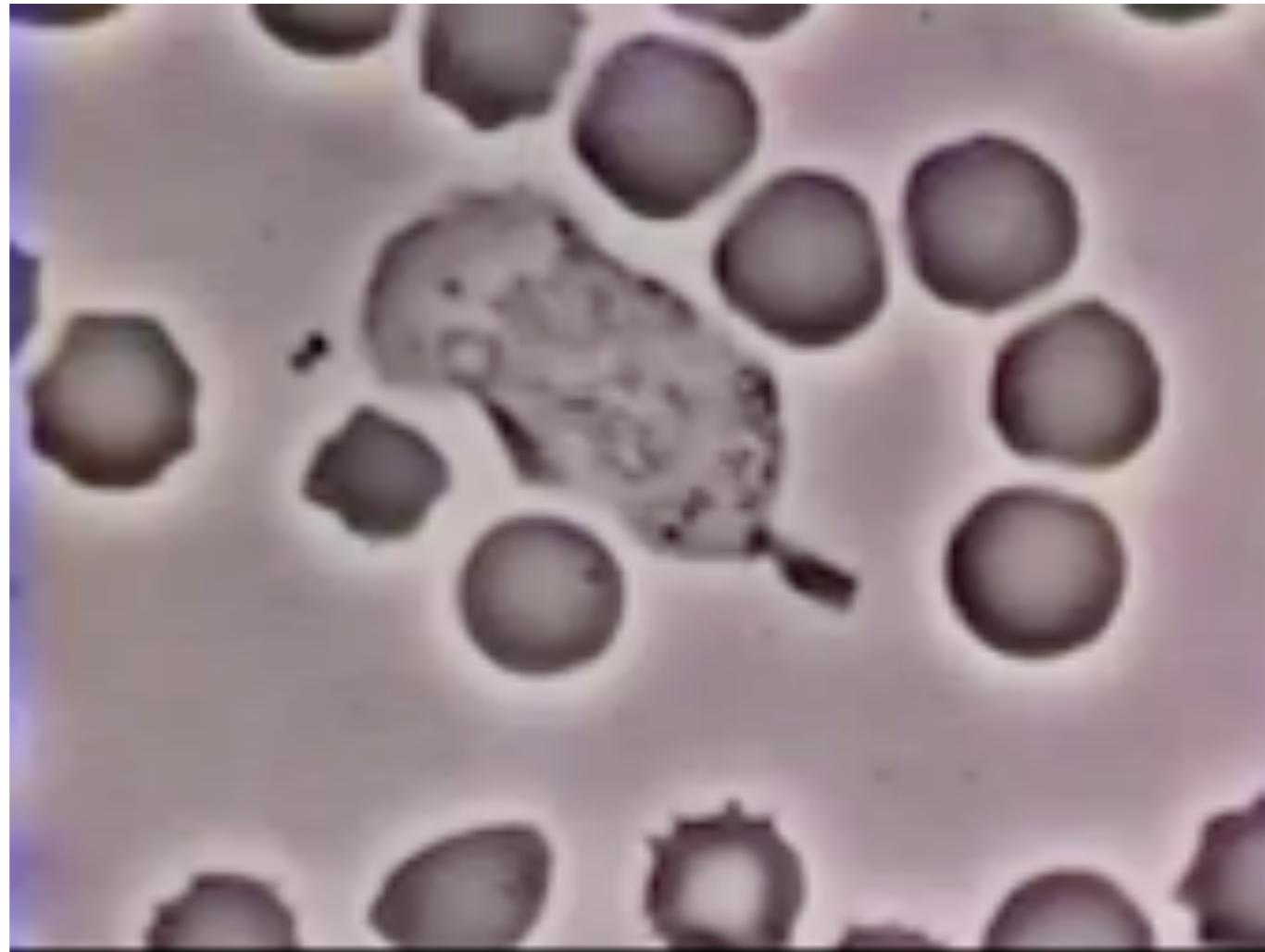
Cellular computing: a primer

Jerome Bonnet
Centre de Biologie Structurale, Montpellier
jerome.bonnet@inserm.fr

QBio Master 1- synthetic biology, 2021



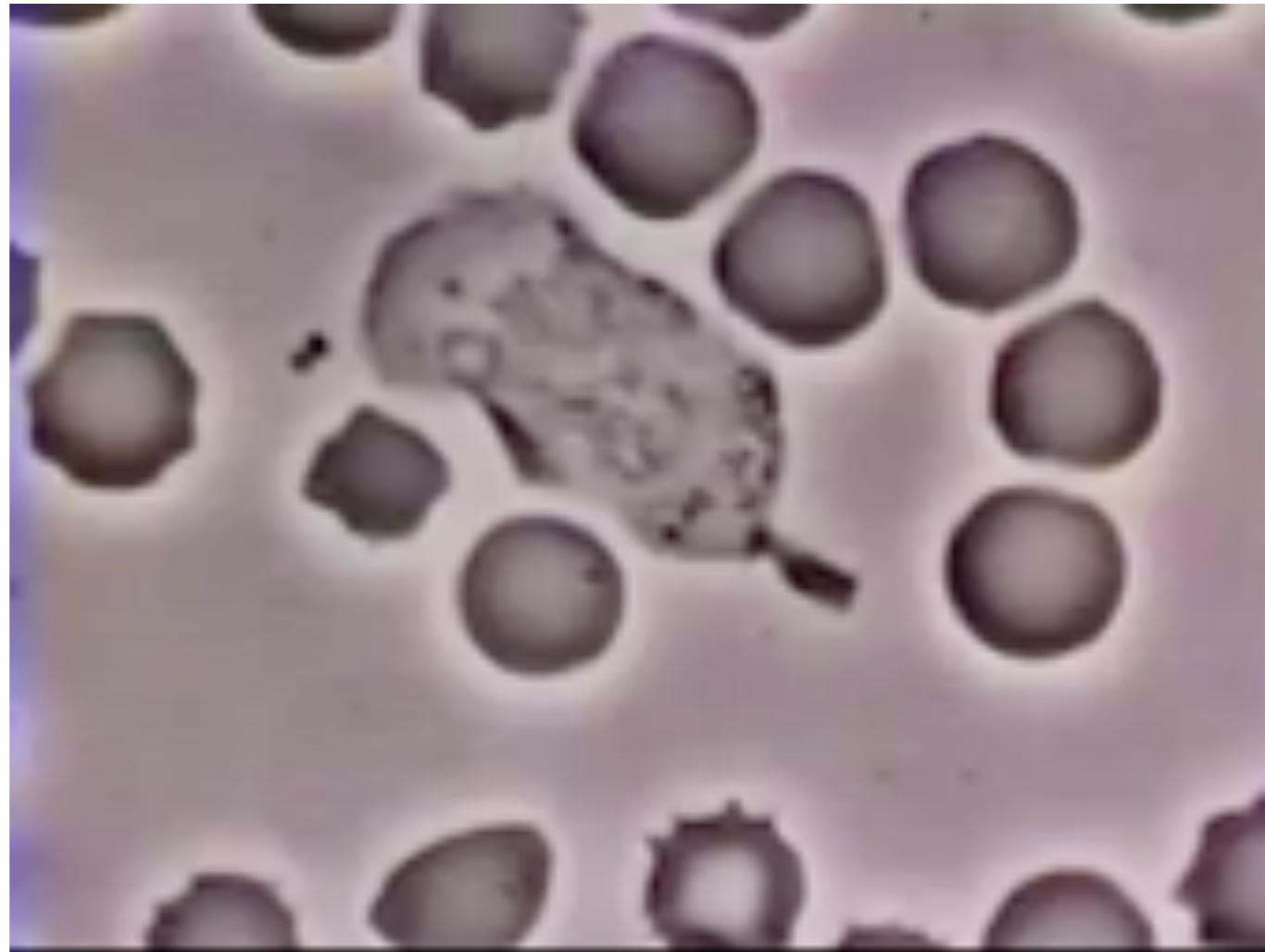
Cells perform computation



Neutrophil chasing bacterium

David Rogers, Vanderbilt University, circa 1956.

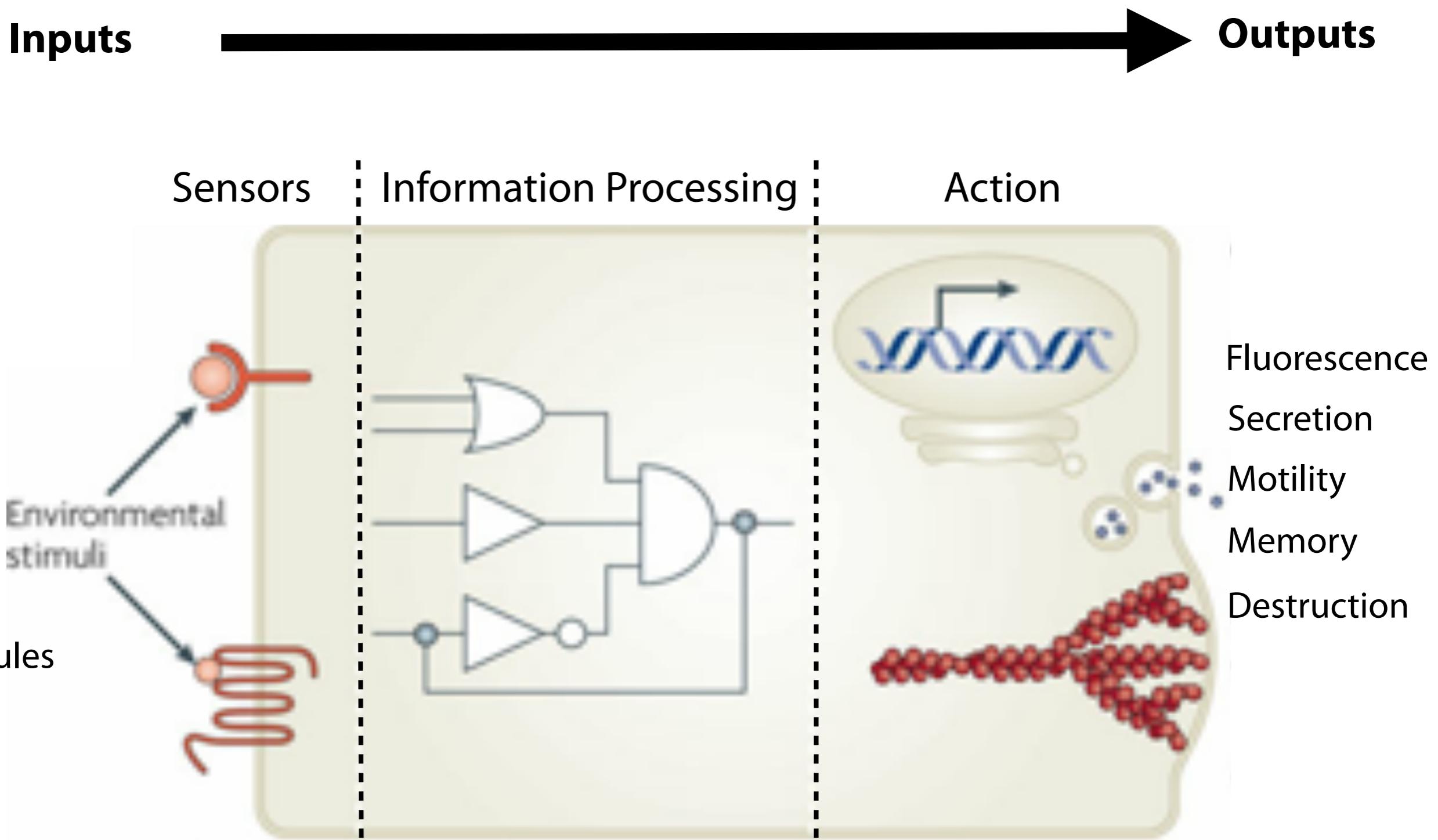
Cells perform computation



Neutrophil chasing bacterium

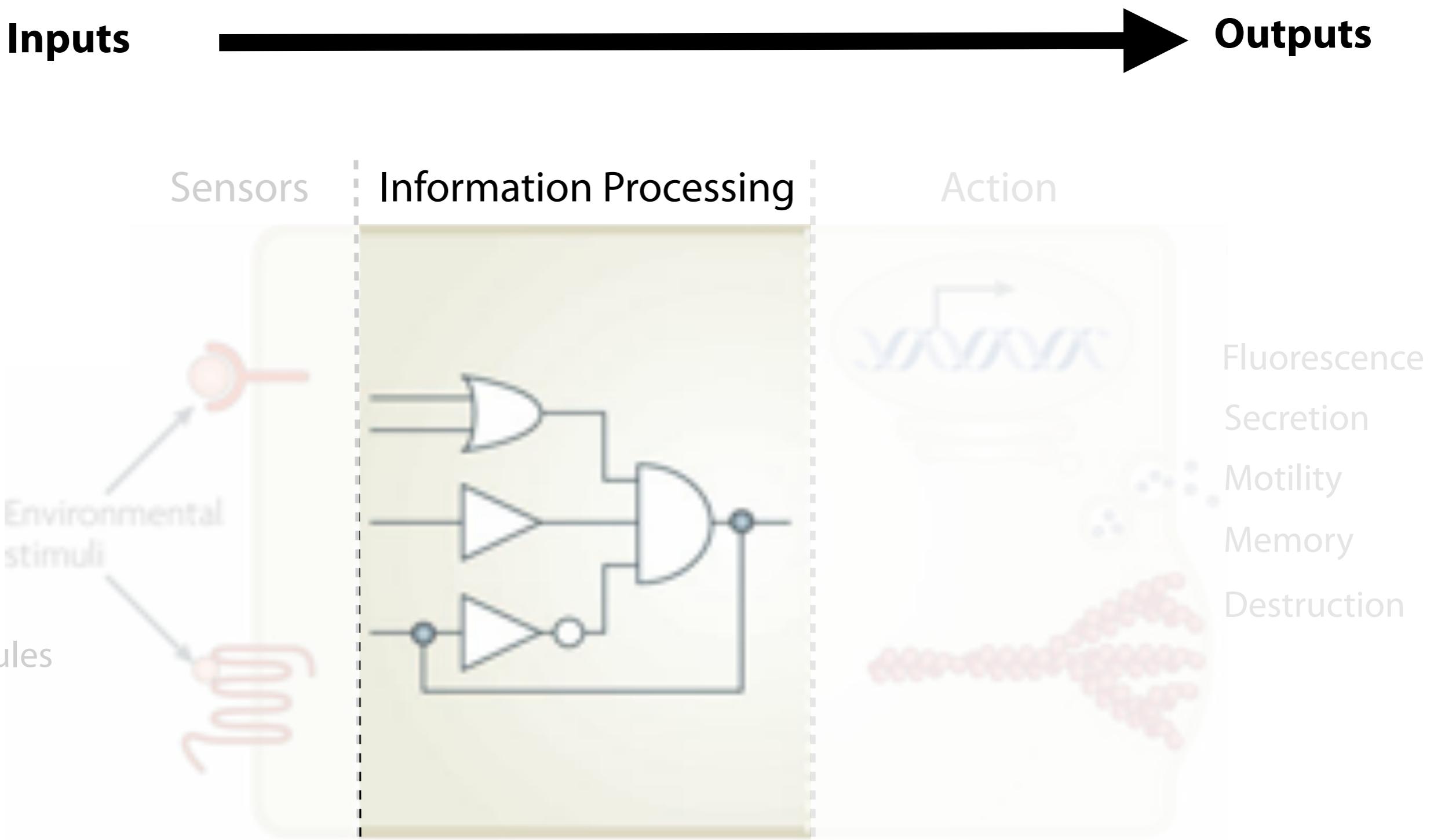
David Rogers, Vanderbilt University, circa 1956.

Cells perform computation



Adapted from Lim, 2010

Cells perform computation



Adapted from Lim, 2010

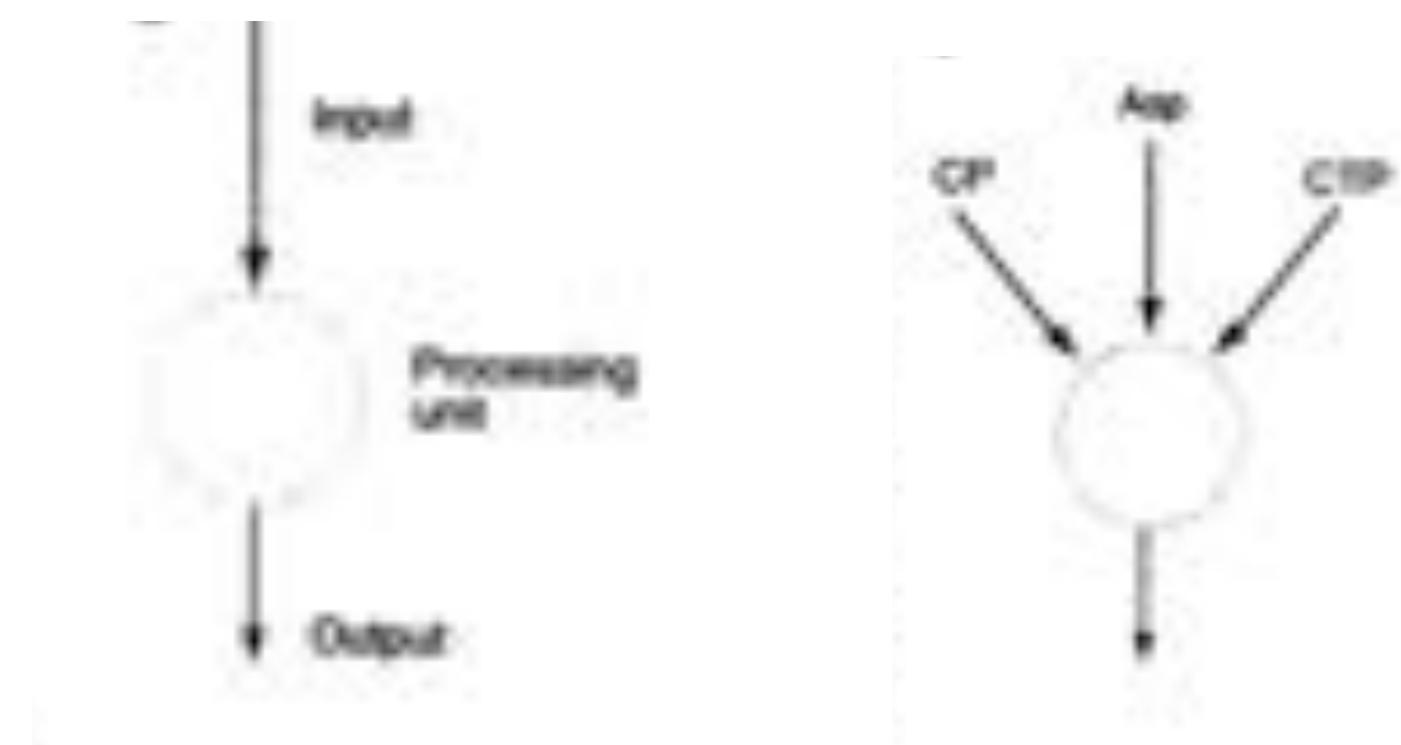
Protein molecules as computational elements in living cells

Dennis Bray



Many proteins in living cells appear to have as their primary function the transfer and processing of information, rather than the chemical transformation of metabolic intermediates or the building of cellular structures. Such proteins are functionally linked through allosteric or other mechanisms into biochemical 'circuits' that perform a variety of simple computational tasks including amplification, integration and information storage.

Nature, 1995



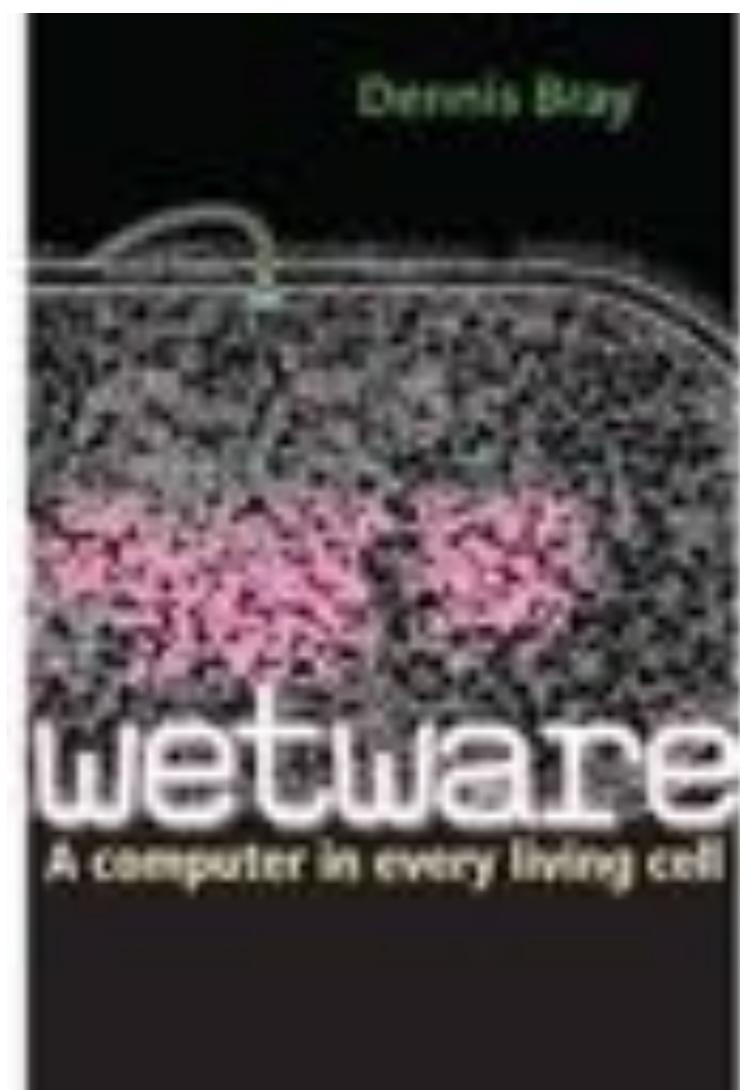
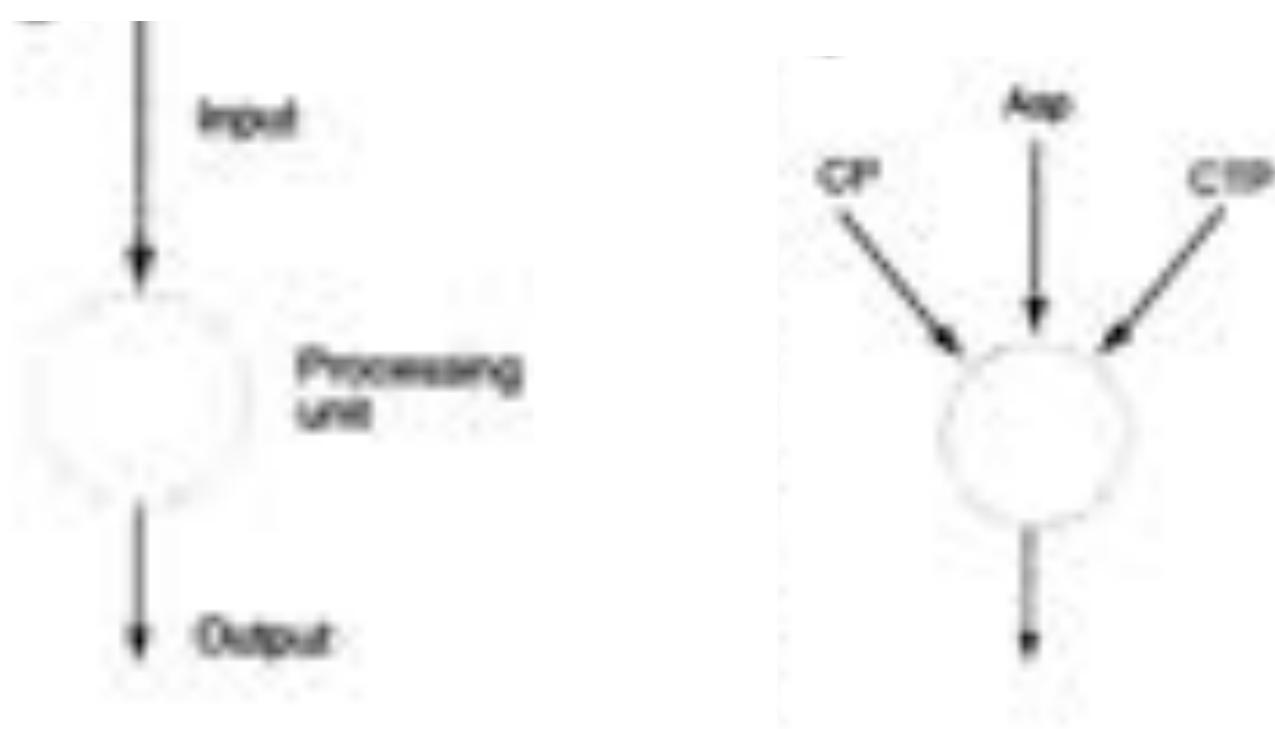
Protein molecules as computational elements in living cells

Dennis Bray



Many proteins in living cells appear to have as their primary function the transfer and processing of information, rather than the chemical transformation of metabolic intermediates or the building of cellular structures. Such proteins are functionally linked through allosteric or other mechanisms into biochemical 'circuits' that perform a variety of simple computational tasks including amplification, integration and information storage.

Nature, 1995



Boolean Formalization of Genetic Control Circuits

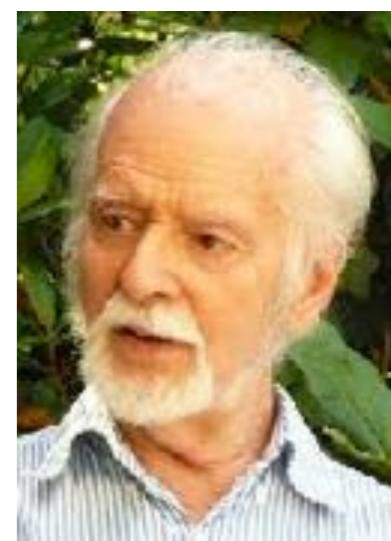
RENÉ THOMAS

*Laboratoire de Génétique, Faculté des Sciences,
et Laboratoire des systèmes logiques et numériques,
Faculté des Sciences appliquées, Université libre de Bruxelles*

(Received 25 January 1973, and in revised form 21 June 1973)

This paper is an attempt to formalize in Boolean terms genetic situations, from simple concepts like recessitivity and *cis*-dominance, to models describing complex control circuits.

A primary objective was to provide a language describing in a compact and unambiguous way, systems which become more and more difficult to describe as their complexity is being unravelled. Expression of a gene is given as a binary function of binary variables of three types:



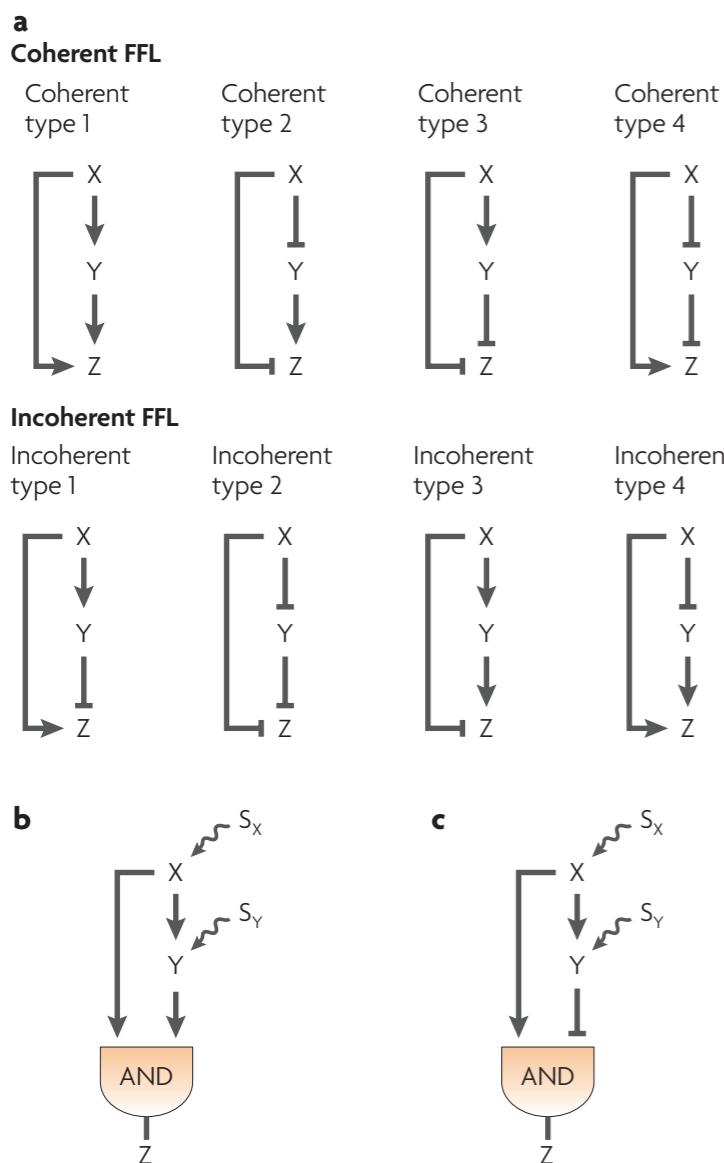
Network motifs: theory and experimental approaches

Uri Alon



Abstract | Transcription regulation networks control the expression of genes. The transcription networks of well-studied microorganisms appear to be made up of a small set of recurring regulation patterns, called network motifs. The same network motifs have recently been found in diverse organisms from bacteria to humans, suggesting that they serve as basic building blocks of transcription networks. Here I review network motifs and their functions, with an emphasis on experimental studies. Network motifs in other biological networks are also mentioned, including signalling and neuronal networks.

Nature review genetics, 2007

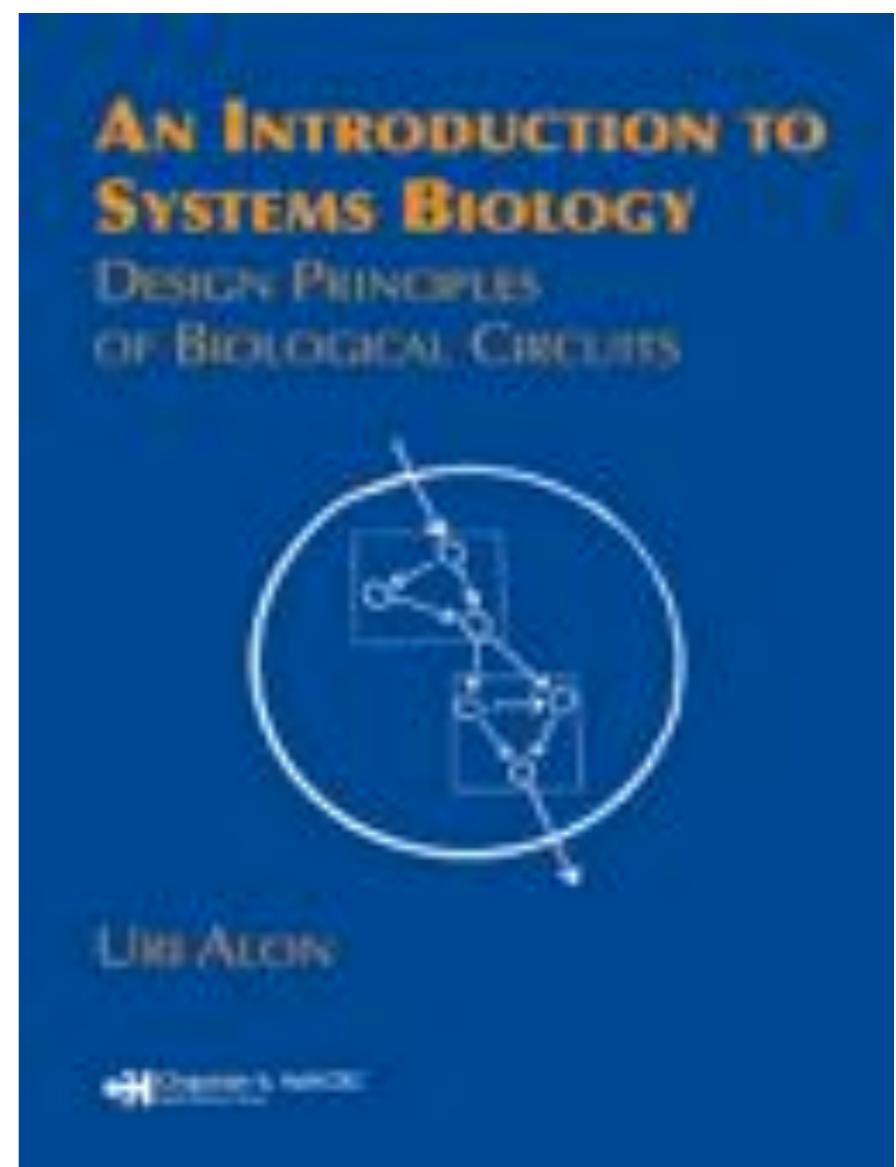


Network motifs: theory and experimental approaches

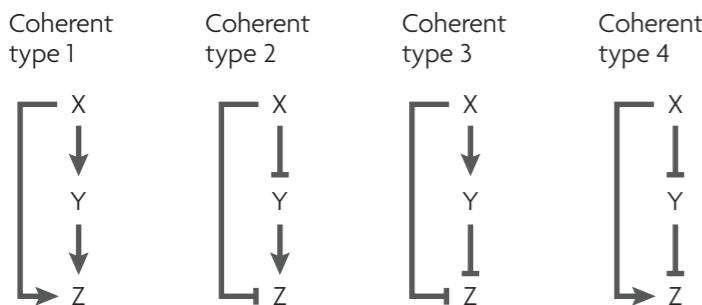
Uri Alon

Abstract | Transcription regulation networks control the expression of genes. The transcription networks of well-studied microorganisms appear to be made up of a small set of recurring regulation patterns, called network motifs. The same network motifs have recently been found in diverse organisms from bacteria to humans, suggesting that they serve as basic building blocks of transcription networks. Here I review network motifs and their functions, with an emphasis on experimental studies. Network motifs in other biological networks are also mentioned, including signalling and neuronal networks.

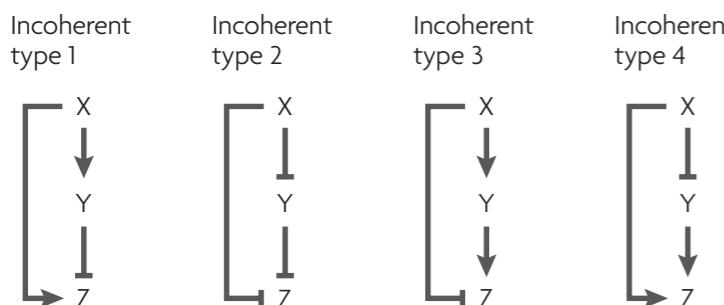
Nature review genetics, 2007



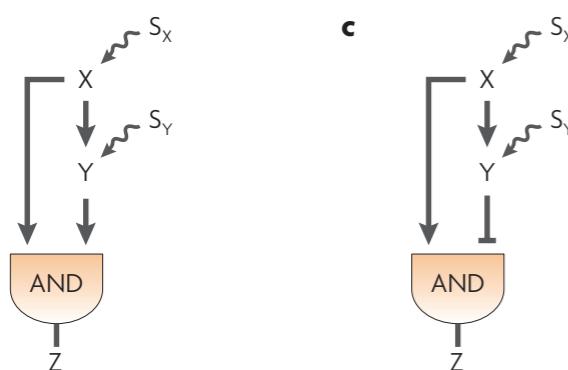
a
Coherent FFL



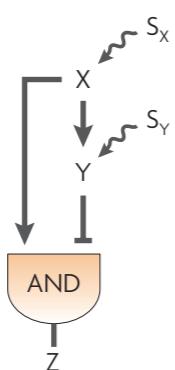
Incoherent FFL



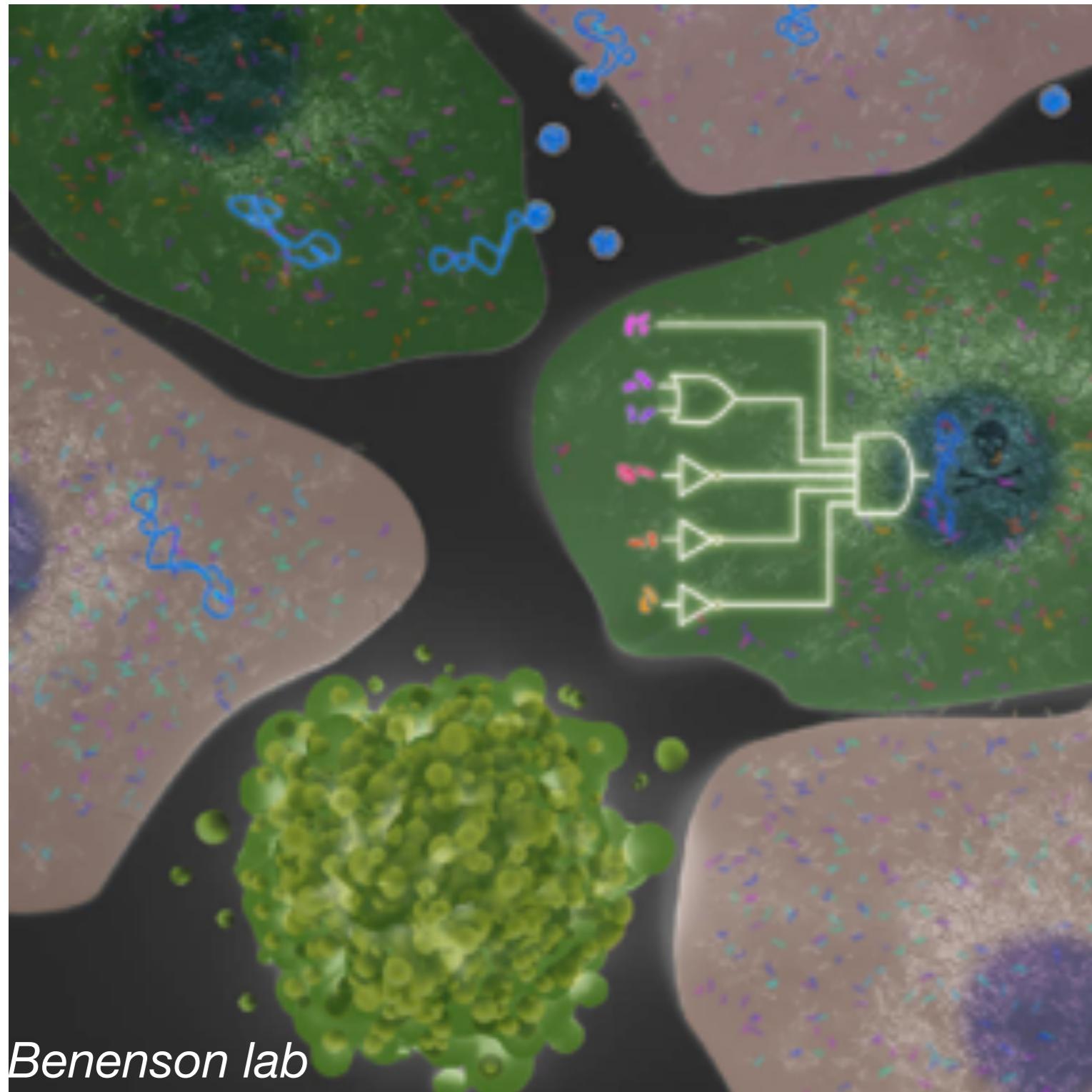
b



c

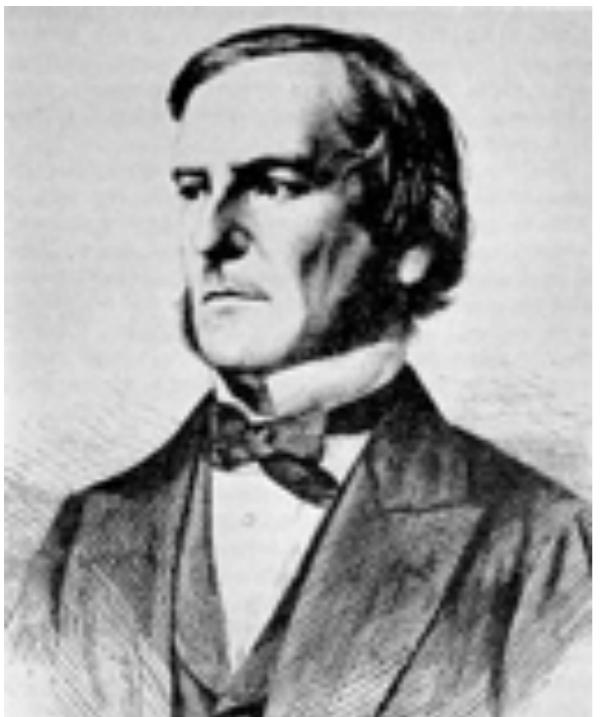


Programming living cells?

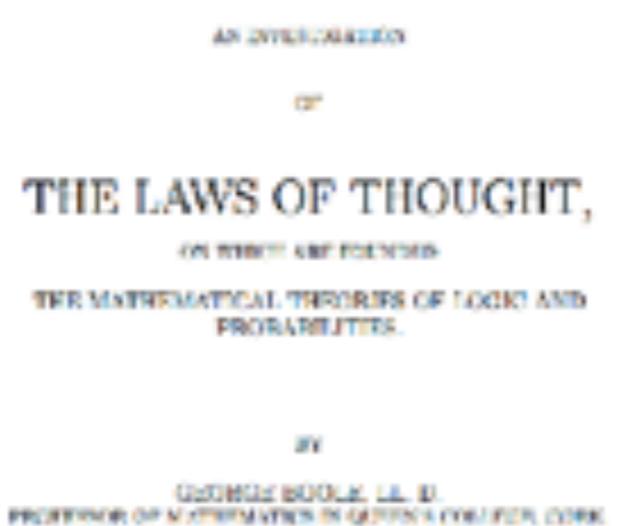


Benenson lab

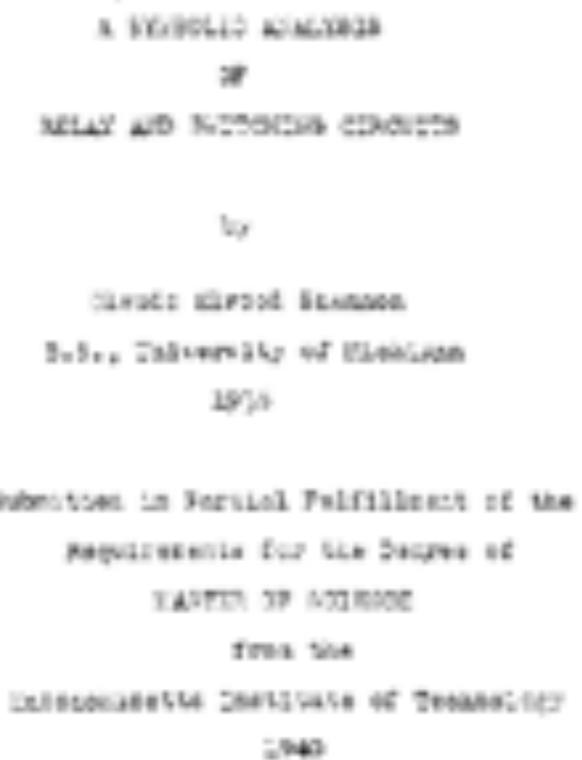
Boole, Shannon, and logic



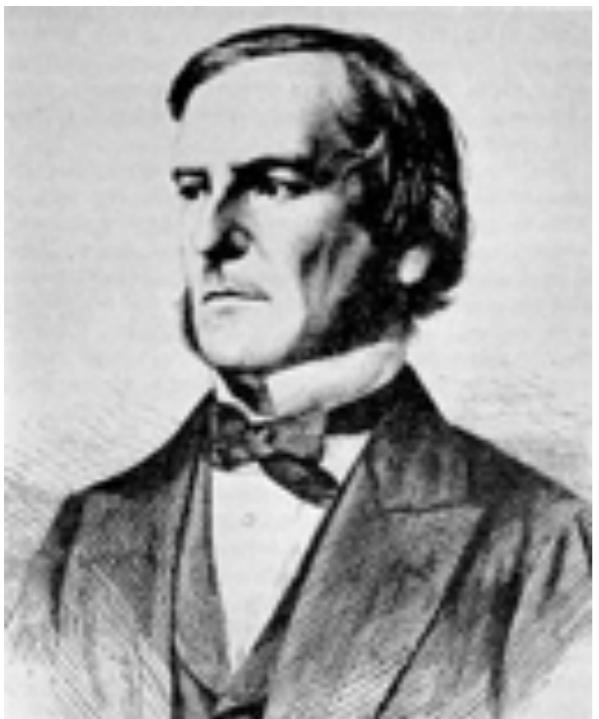
George Boole, c.1854



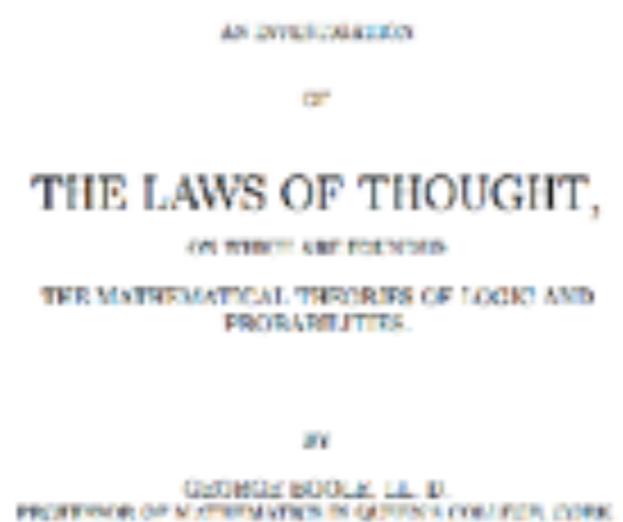
Claude Shannon, c.1937



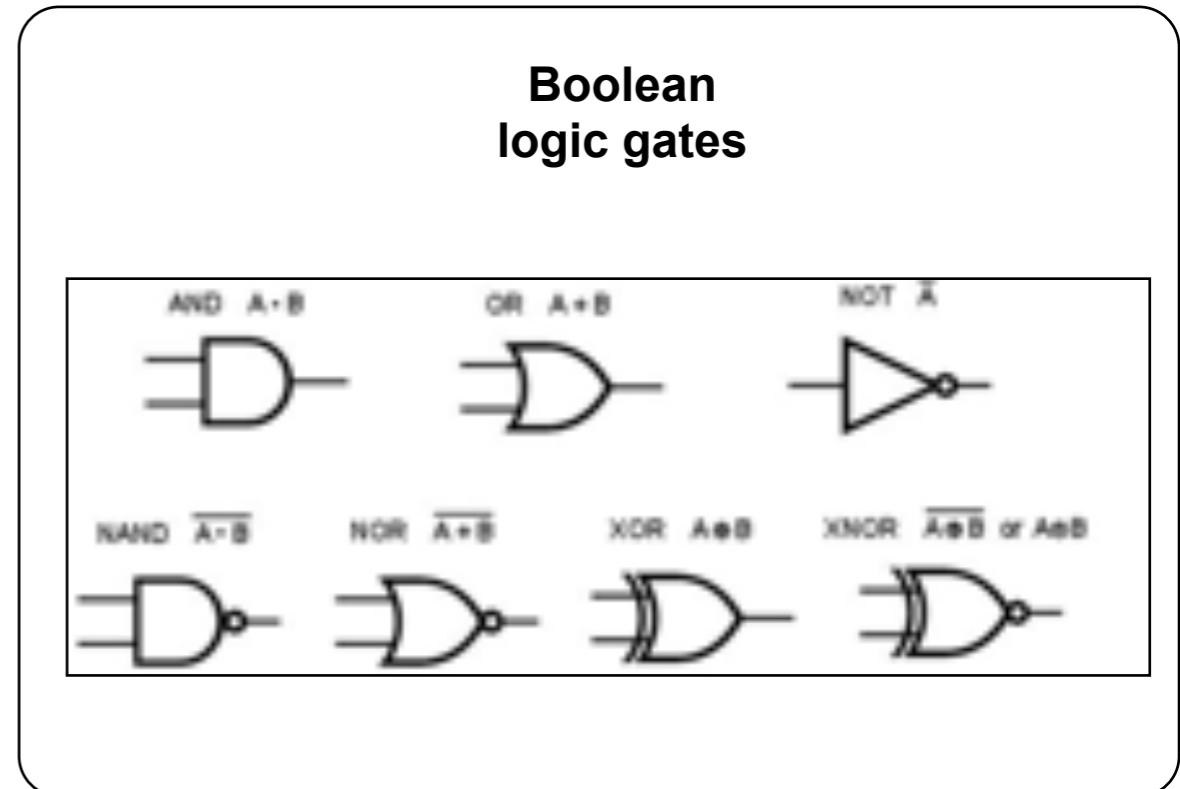
Boole, Shannon, and logic

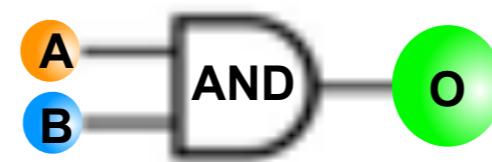
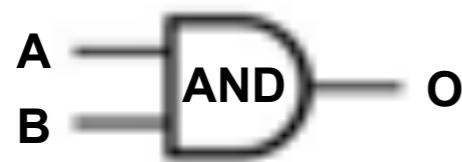


George Boole, c.1854

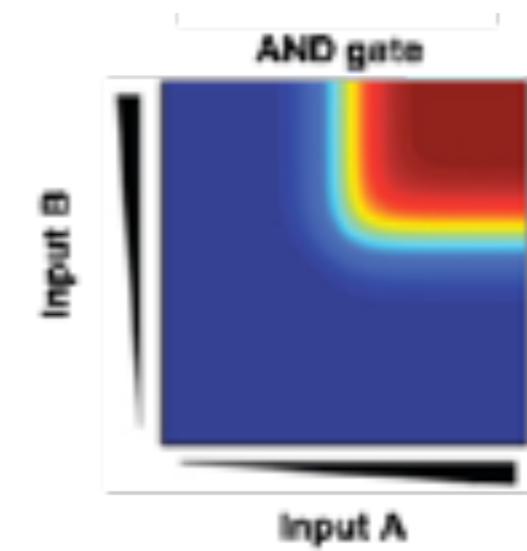


Claude Shannon, c.1937



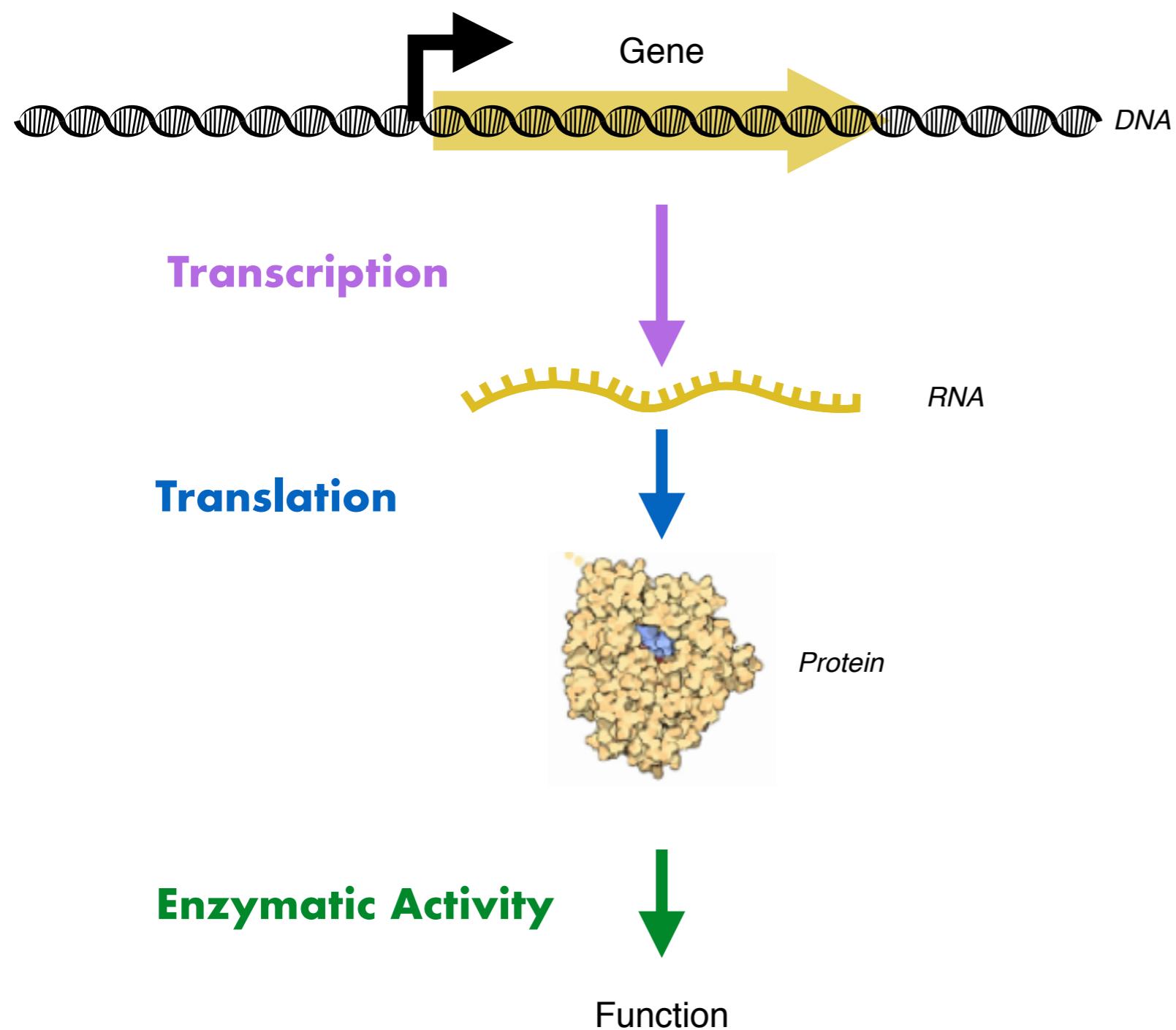


A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

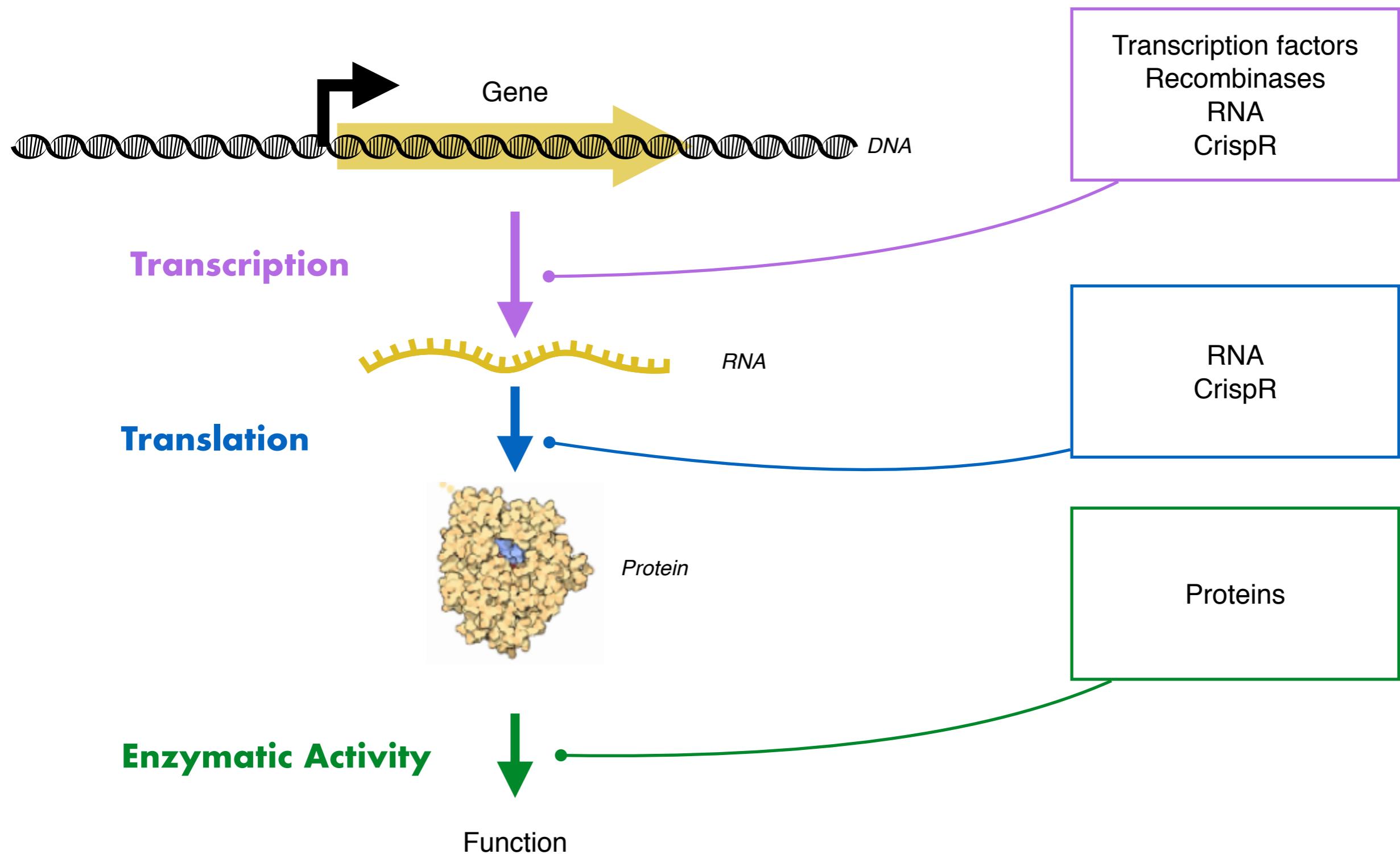


A	B	sortie
0	0	0
0	1	0
1	0	0
1	1	1

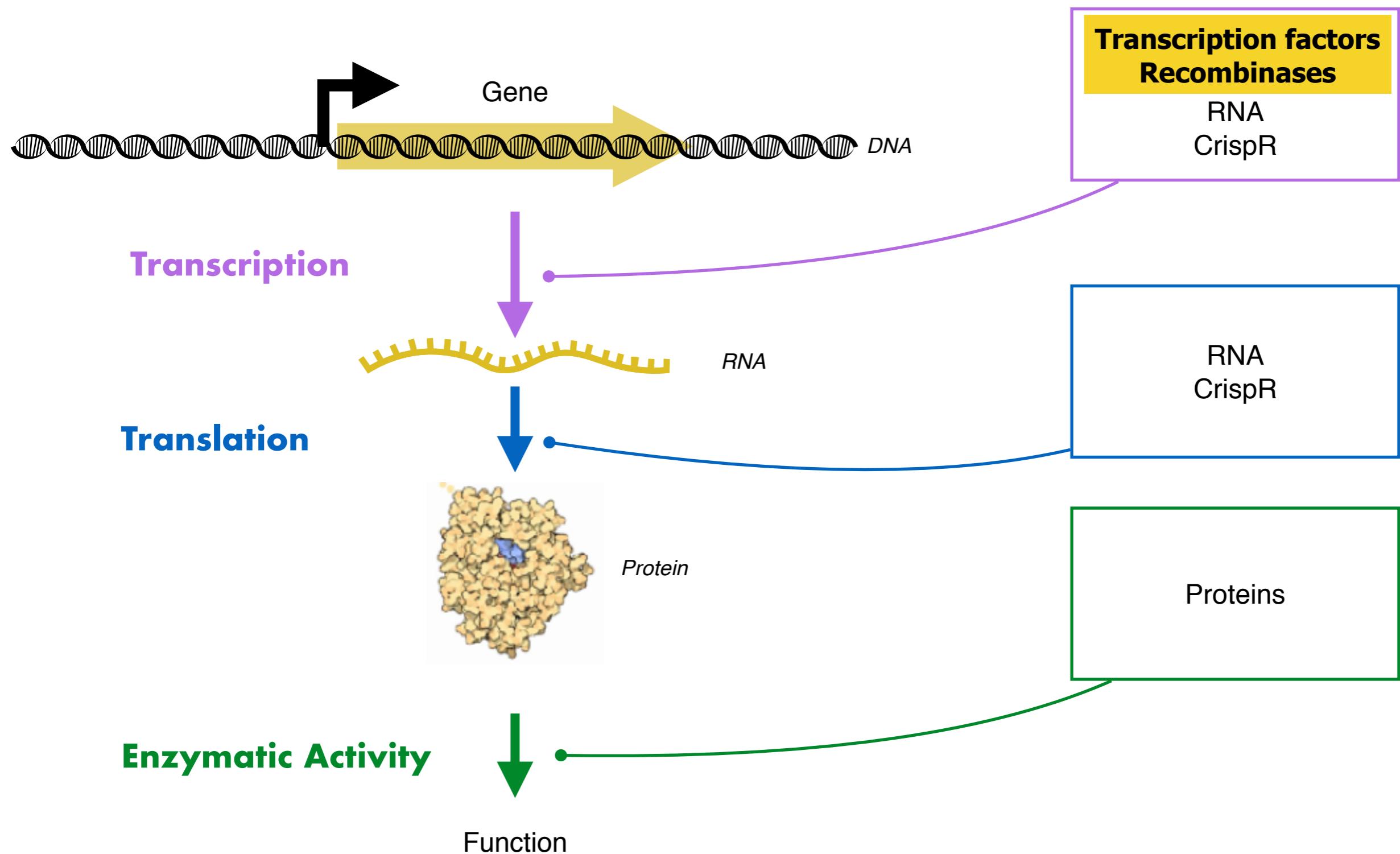
Controlling the central dogma to program cellular behavior



Controlling the central dogma to program cellular behavior

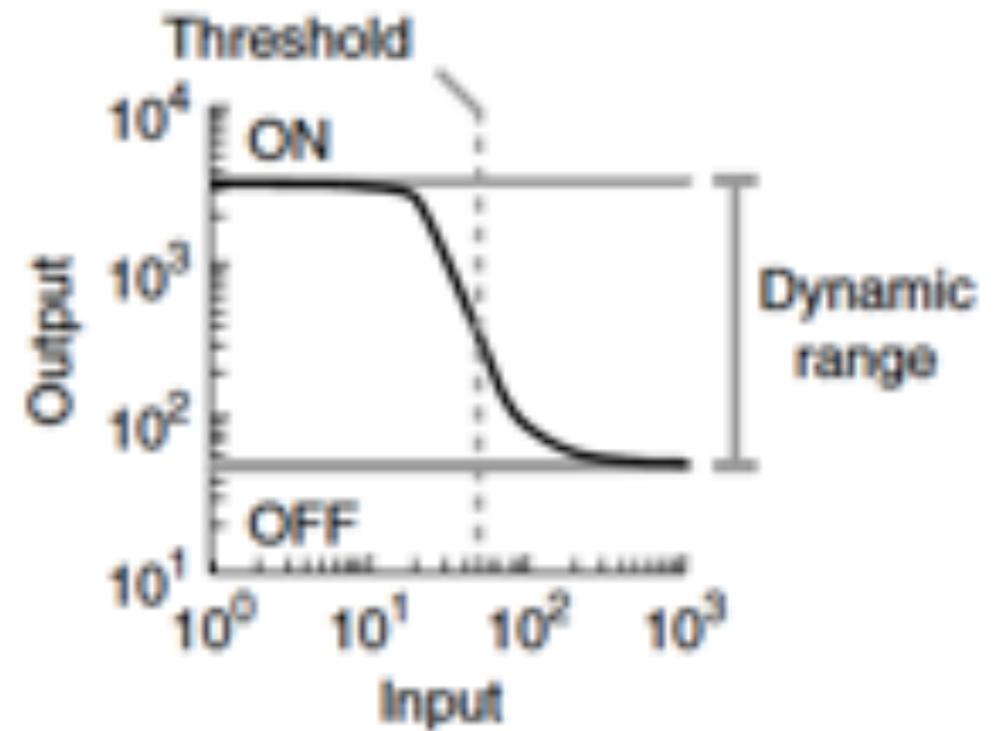
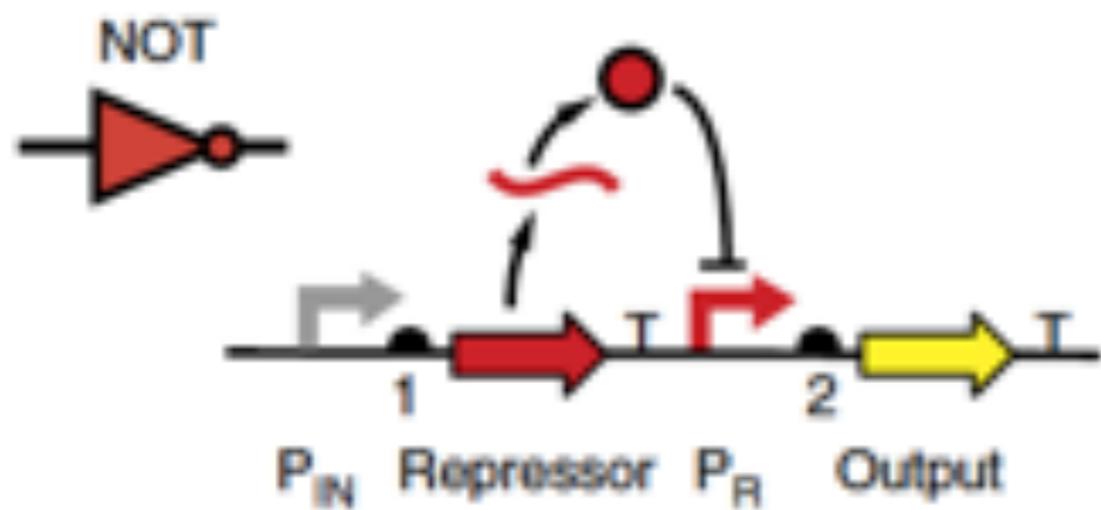


Controlling the central dogma to program cellular behavior



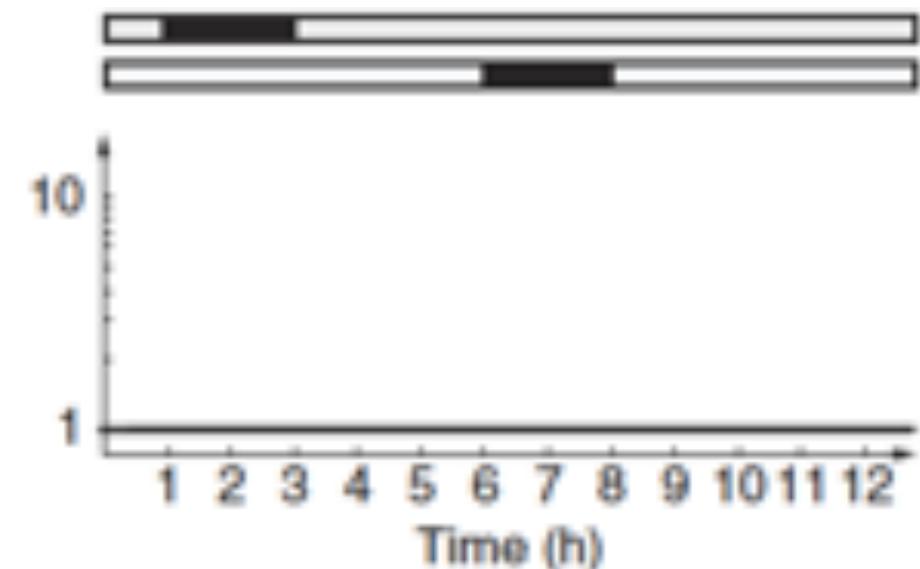
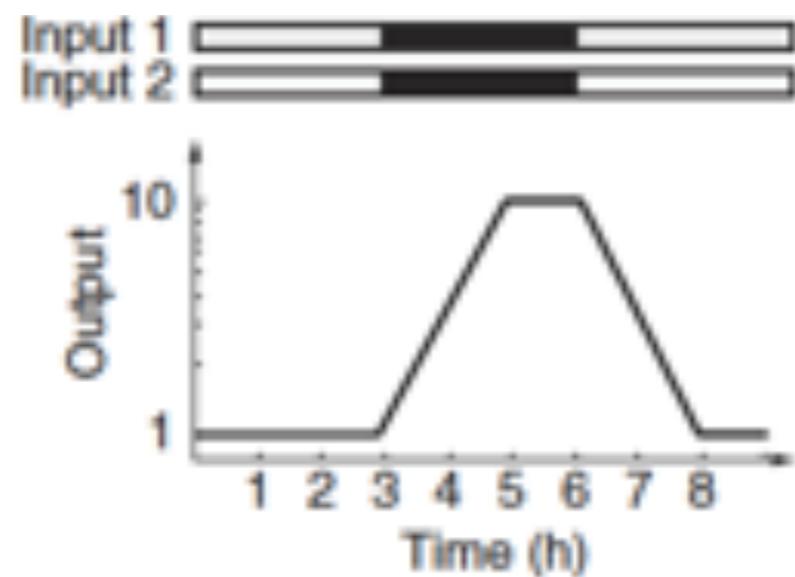
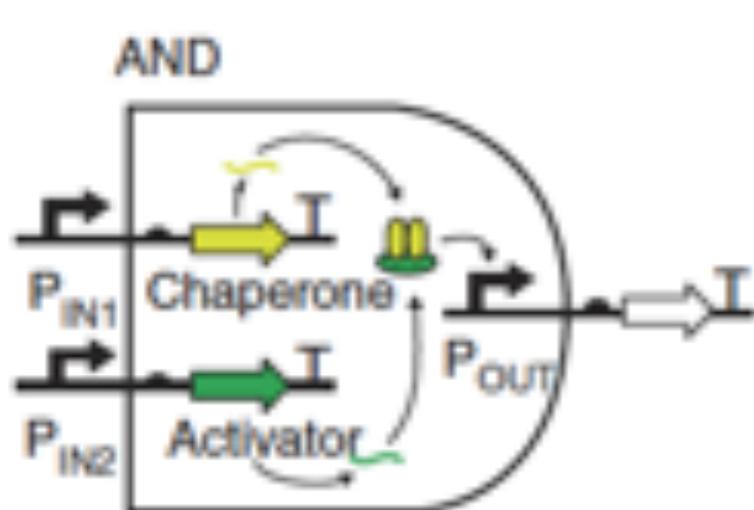
NOT gate (inverter) using a repressor

A	OUT
0	1
1	0



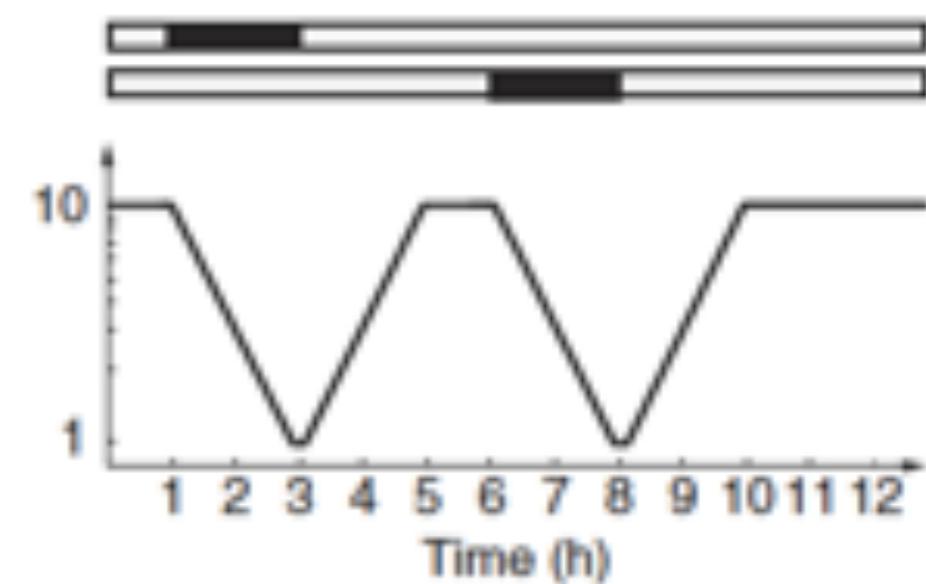
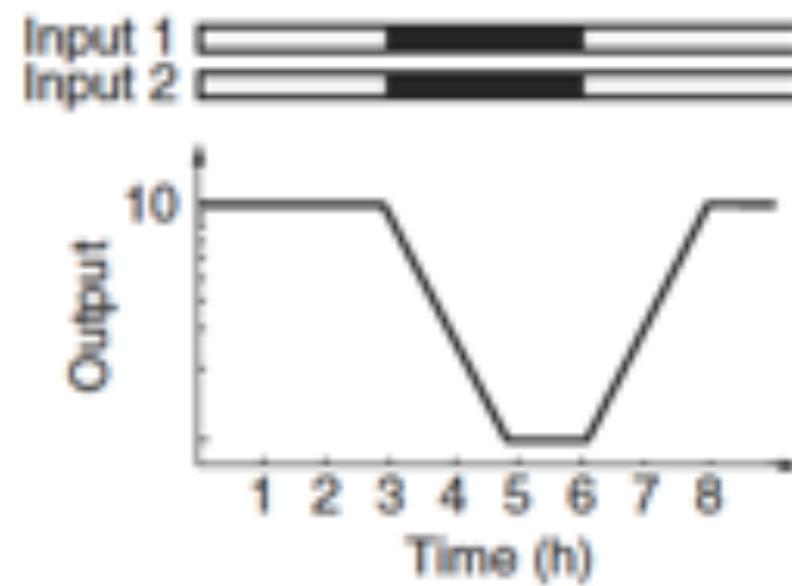
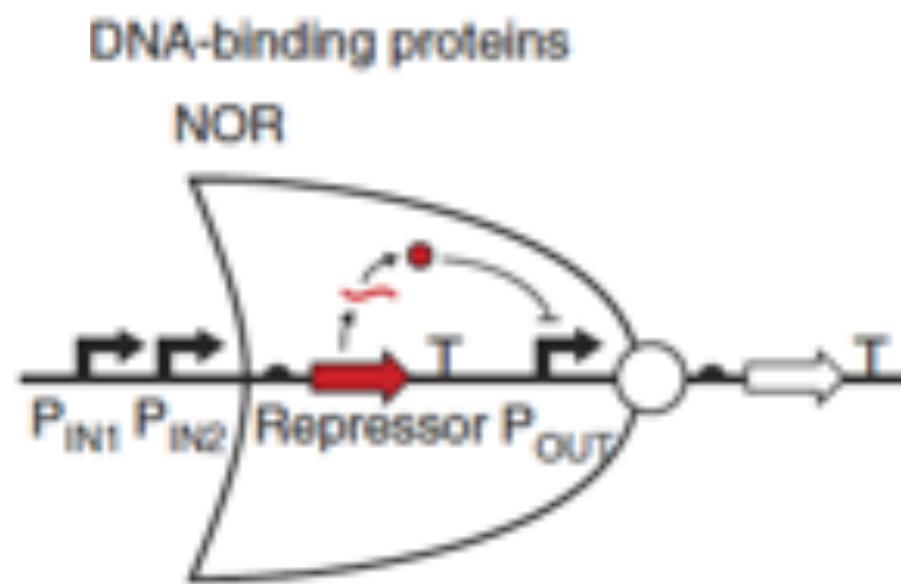
AND gate using transcriptional control

Gate	Truth Table		
	IN1	IN2	OUT
AND	0	0	0
	1	0	0
	0	1	0
	1	1	1



NOR gates using tandem promoters and a repressor

Gate	Truth Table		
	P _{IN1}	P _{IN2}	OUT
NOR	0	0	1
	1	0	0
	0	1	0
	1	1	0

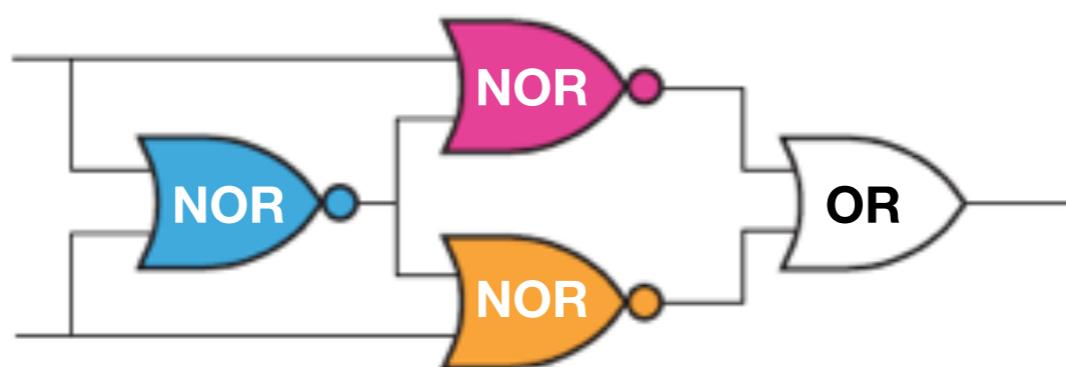


XOR by multilayering NOR/OR gates

Gate	Truth Table		
	Ctrl1	Ctrl2	DUT
XOR	0	0	0
	1	0	1
	0	1	1
	1	1	0

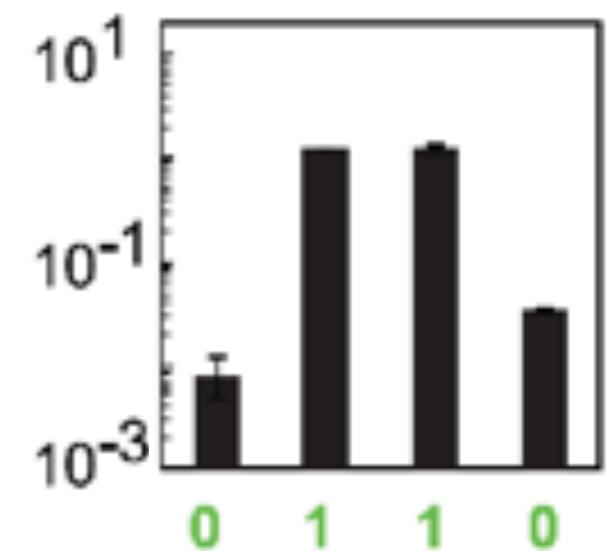
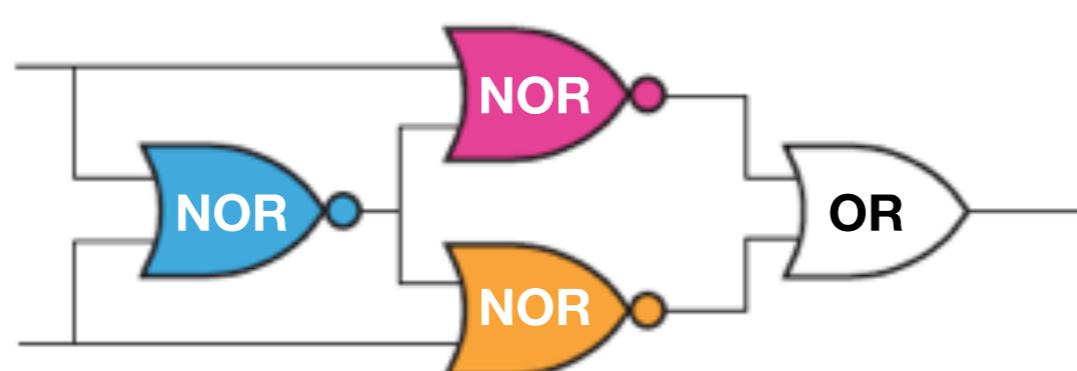
XOR by multilayering NOR/OR gates

Gate	Truth Table		
	Ctrl1	Ctrl2	DUT
XOR	0	0	0
	1	0	1
	0	1	1
	1	1	0



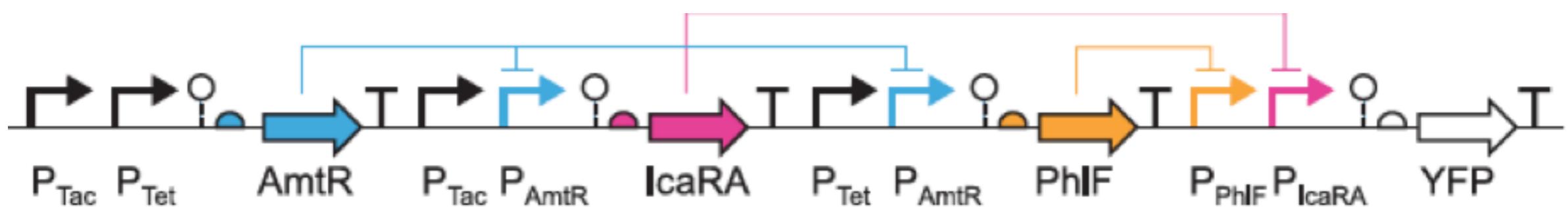
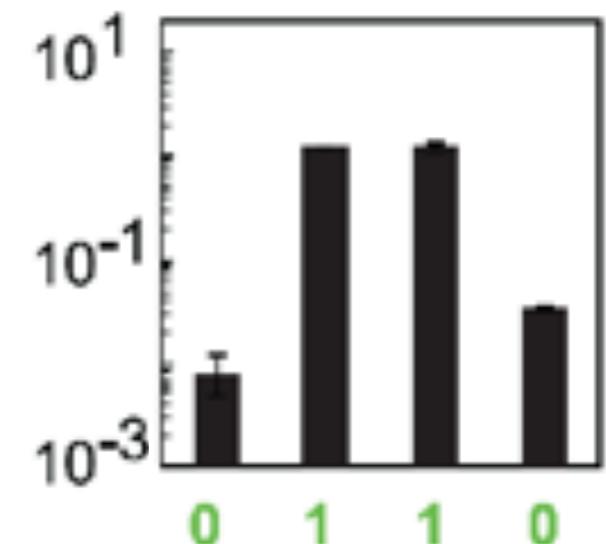
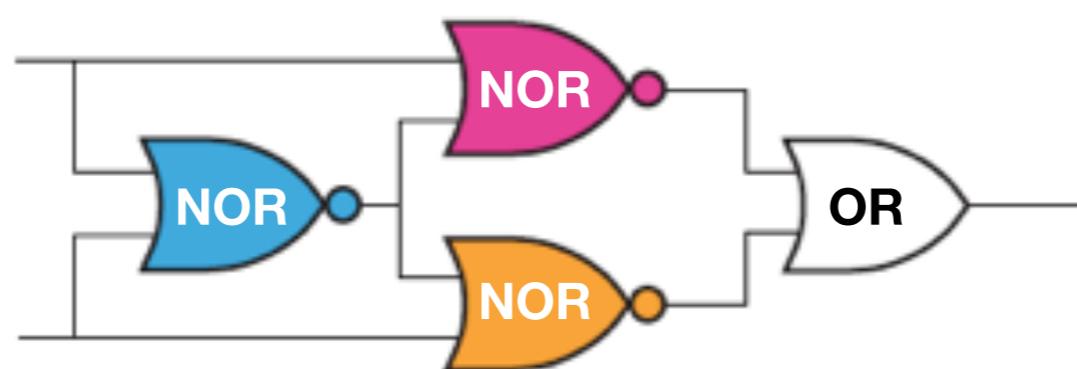
XOR by multilayering NOR/OR gates

Gate	Truth Table		
	Ctrl1	Ctrl2	DUT
XOR	0	0	0
	1	0	1
	0	1	1
	1	1	0



XOR by multilayering NOR/OR gates

Gate	Truth Table
	Ctrl1 Ctrl2 DUT
	0 0 0
	1 0 1
	0 1 1
	1 1 0



More complex gates design using CELLO



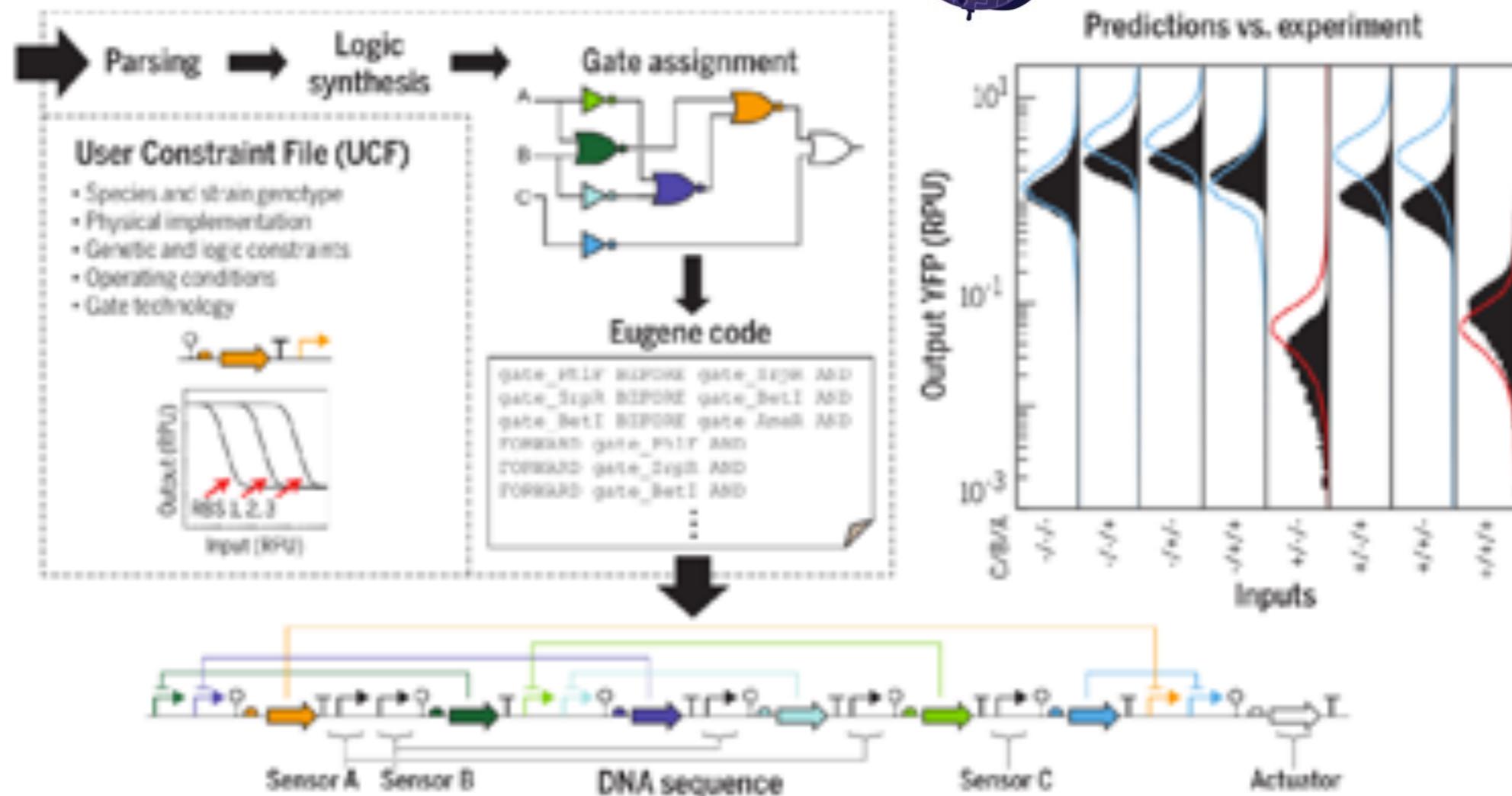
Cello design specification

Sensors	name low high promoter sequence	
A	0.003 2.8	ACCGATCCTTGGCTGTGTTGACAAATT
B	0.001 4.4	TACTCCACCTTGCCTTTTCGCTTA
C	0.008 2.5	ACTTTTCACTCCGCGCCATTCAGAT

Verilog	<pre>module gate1(output out, input A,B,C); begin case({A,B,C}) 3'b000 : out = 1'b1; 3'b001 : out = 1'b0; 3'b010 : out = 1'b1; 3'b011 : out = 1'b0; 3'b100 : out = 1'b0; 3'b101 : out = 1'b1; 3'b110 : out = 1'b0; 3'b111 : out = 1'b1; endcase end endmodule</pre>
----------------	---

Actuators	name sequence
YFP	ATGTTGAGCAAGGCGAGGACTGTGACCGGCT

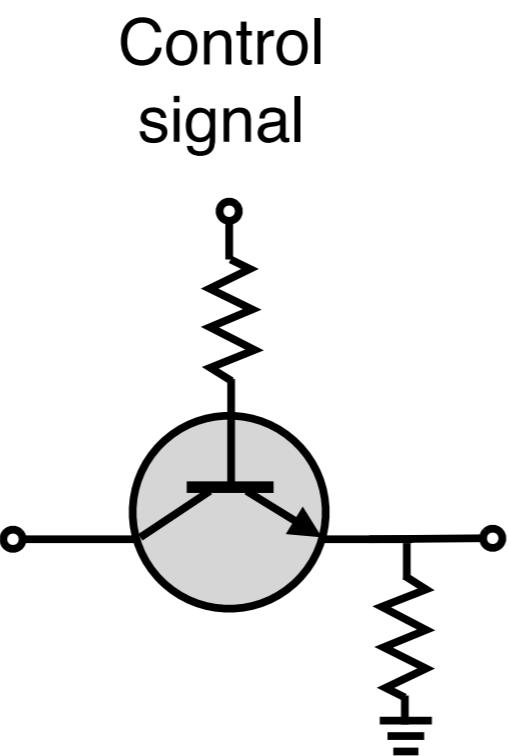
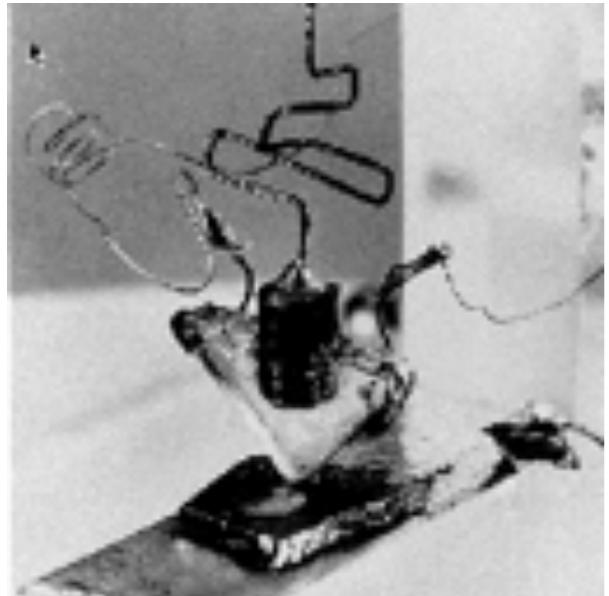
Run



<http://www.cellocad.org>

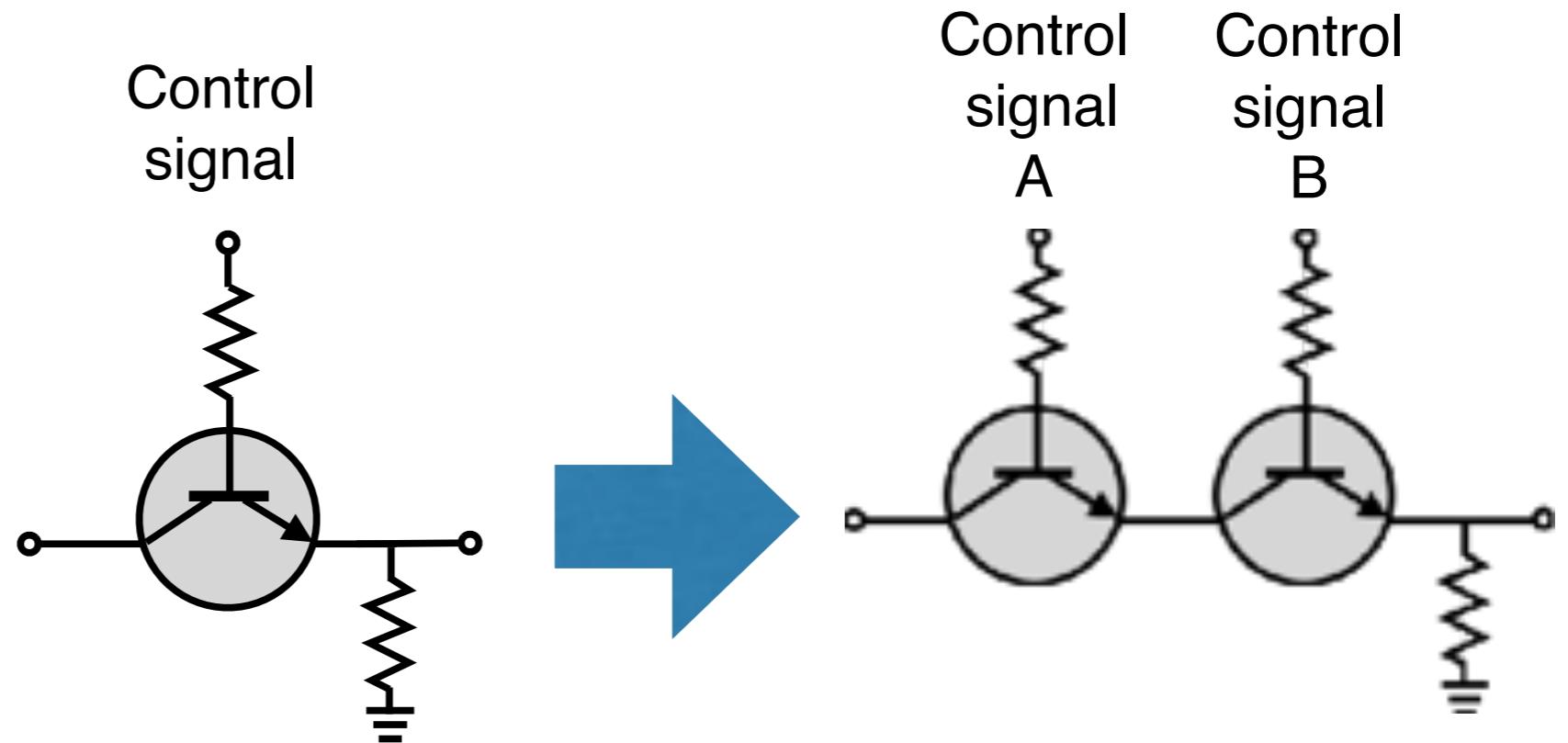
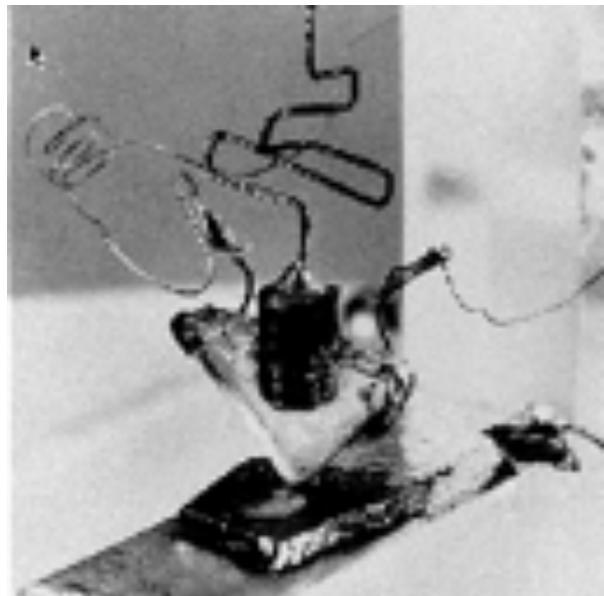
Transistors et portes logiques

Transistor: Un signal contrôle le flux du courant entre 2 points du circuit.



Transistors et portes logiques

Transistor: Un signal contrôle le flux du courant entre 2 points du circuit.



Porte AND

A	B	sortie
0	0	0
0	1	0
1	0	0
1	1	1

Biological current: flow of RNA polymerase along DNA

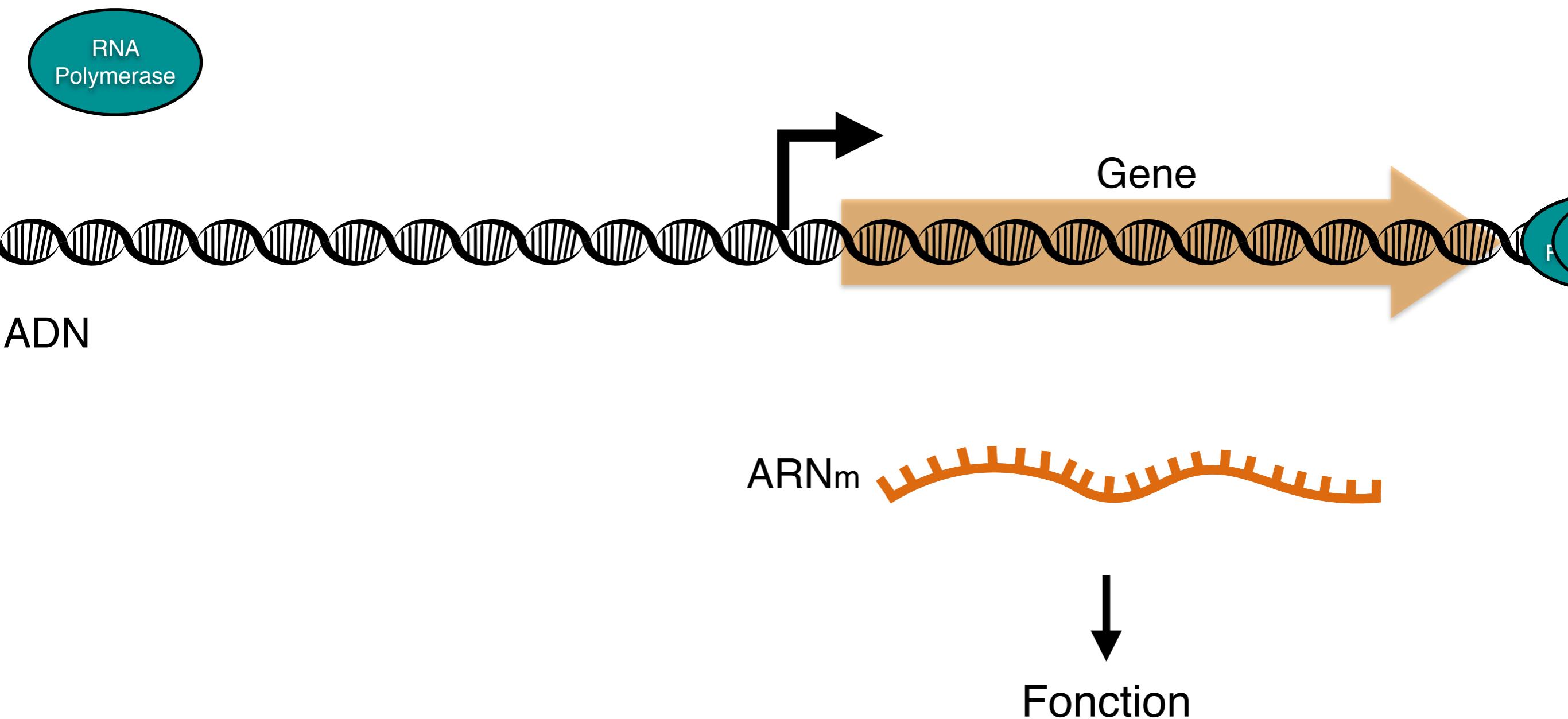
Biological current: flow of RNA polymerase along DNA

RNA
Polymerase

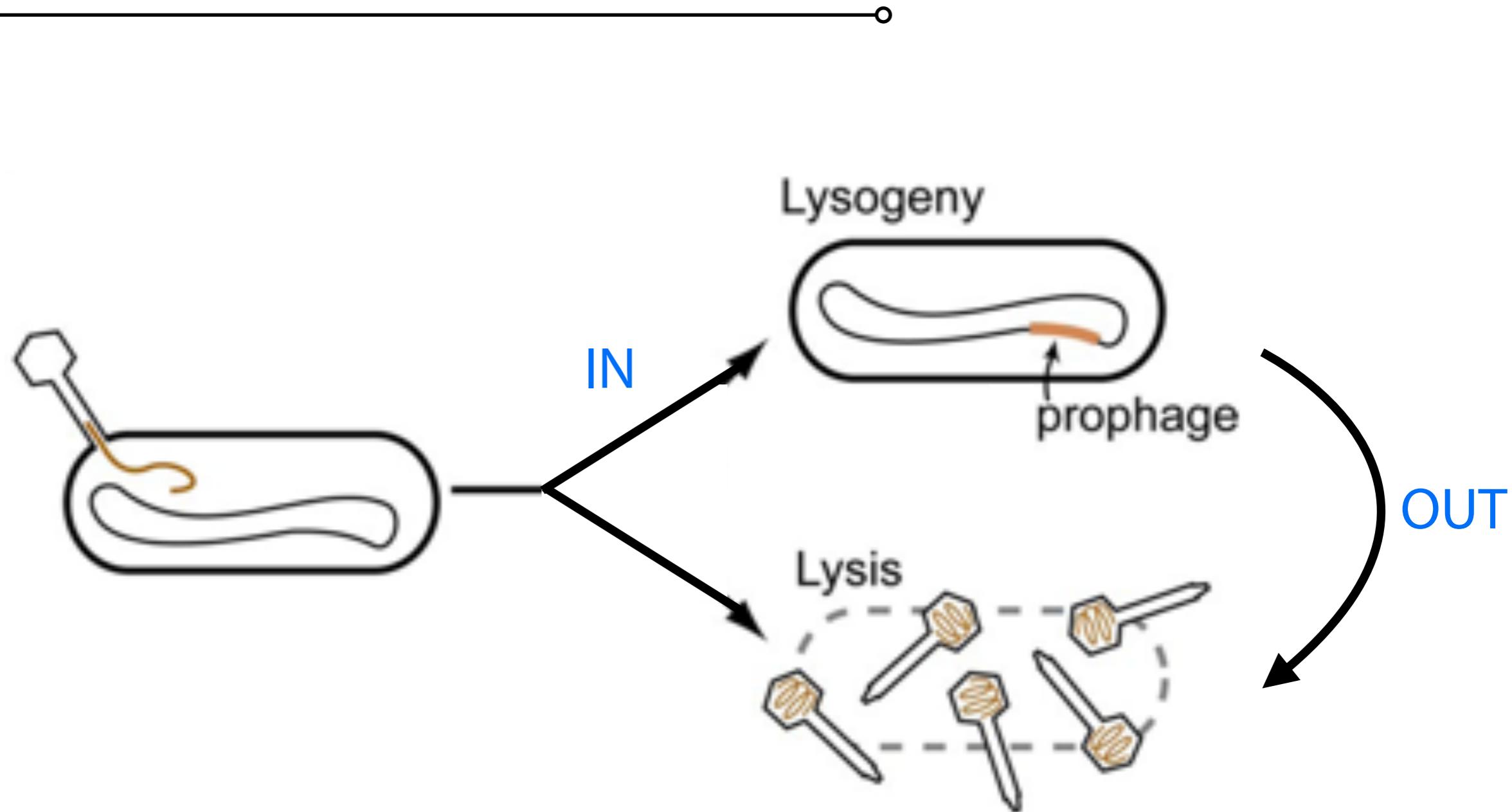


ADN

Biological current: flow of RNA polymerase along DNA



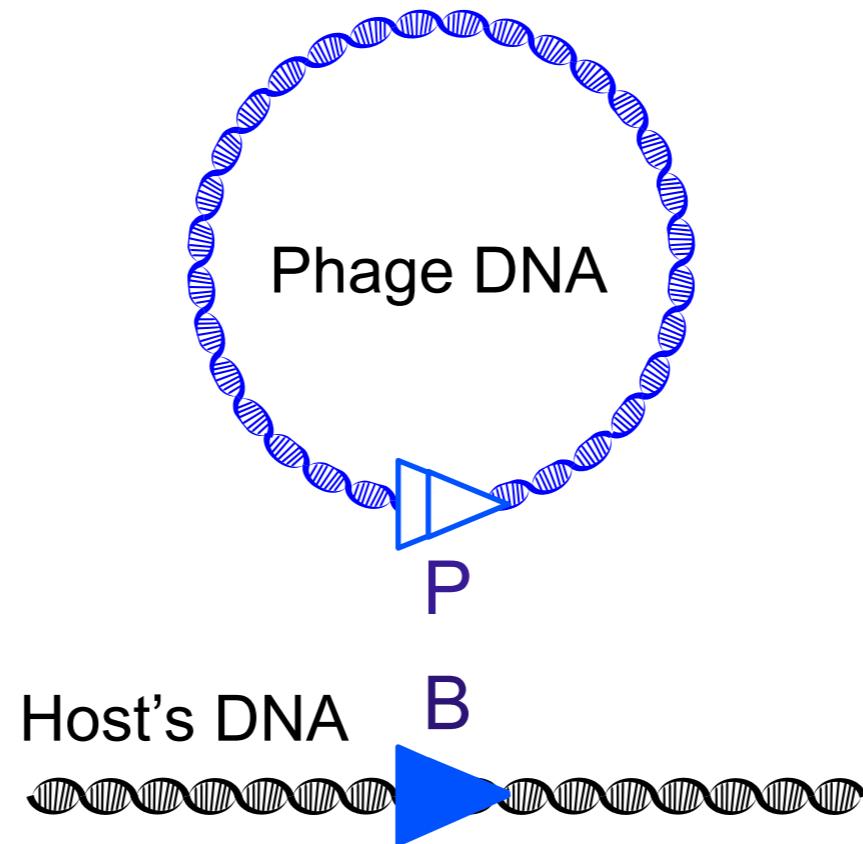
Temperate Phages Can Integrate Their Genome Into Their Host's Chromosome



Adapted from St-Pierre & Endy, PNAS, 2008

Hijacking the phage recombination machinery

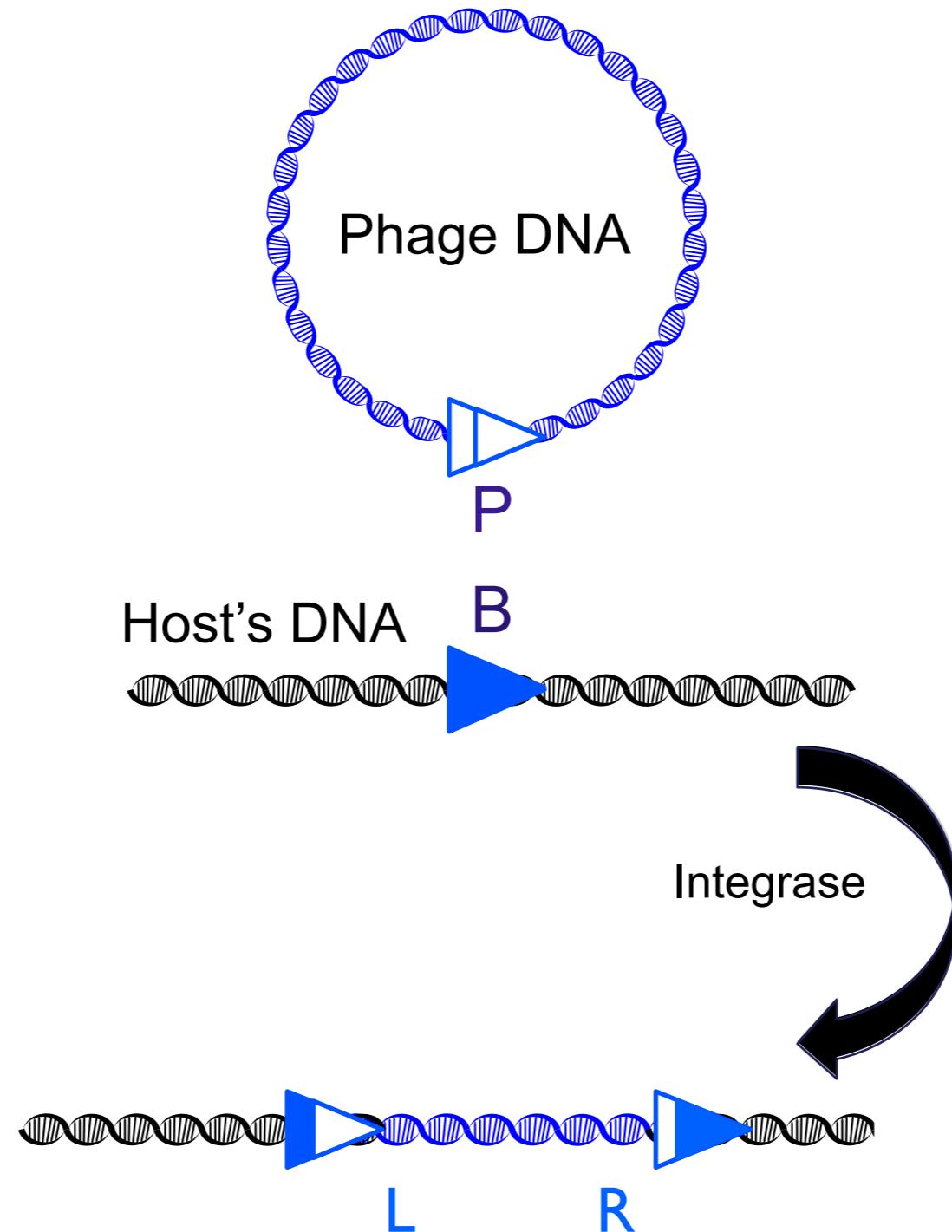
Natural Int-Xis system



See also: Reyes et al., 1979 ; Pollock and Nash, 1983; Podhajska et al., 1985

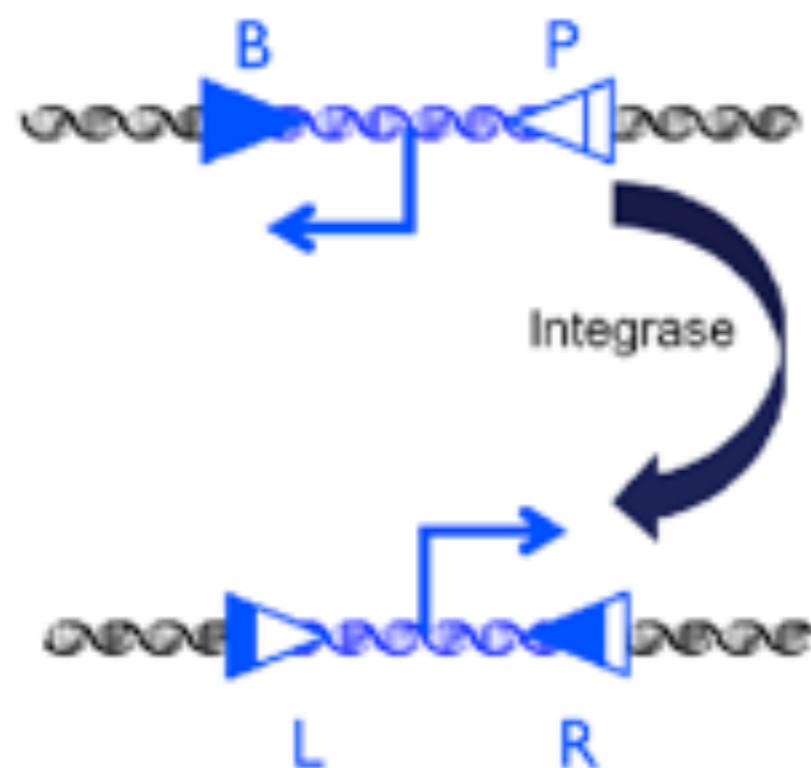
Hijacking the phage recombination machinery

Natural Int-Xis system

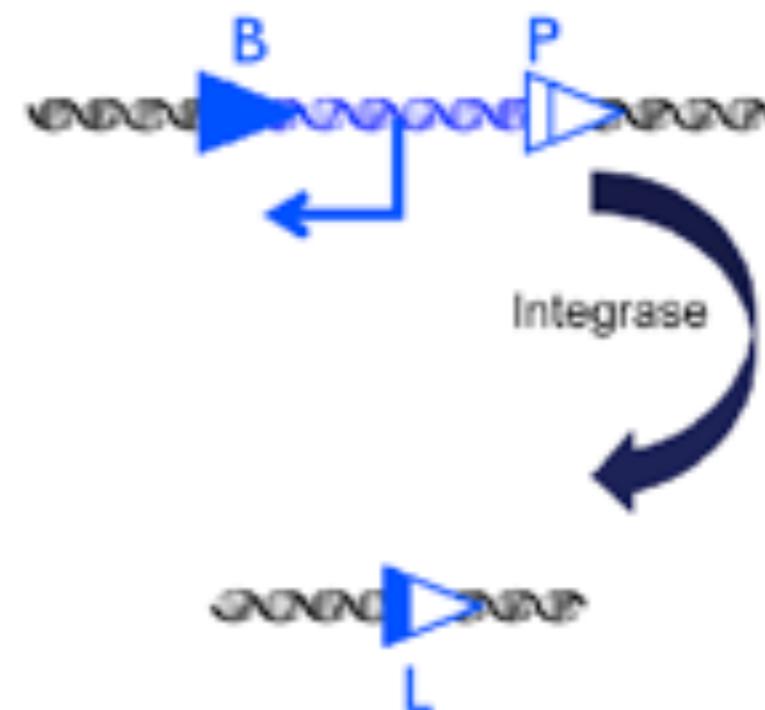


See also: Reyes et al., 1979 ; Pollock and Nash, 1983; Podhajska et al., 1985

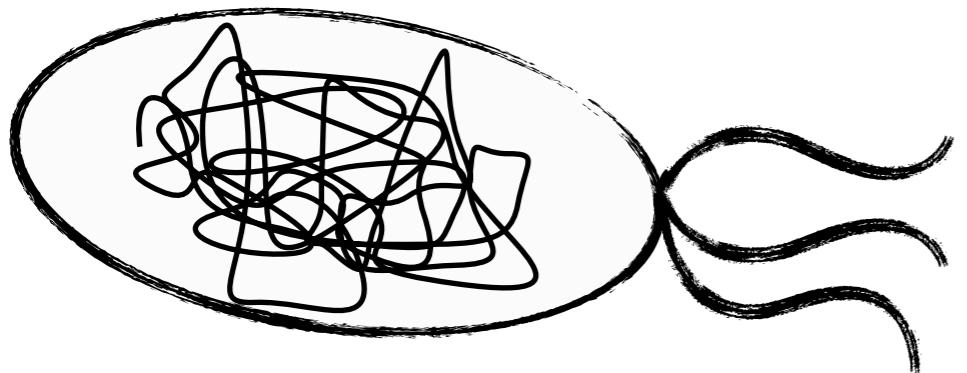
INVERSION



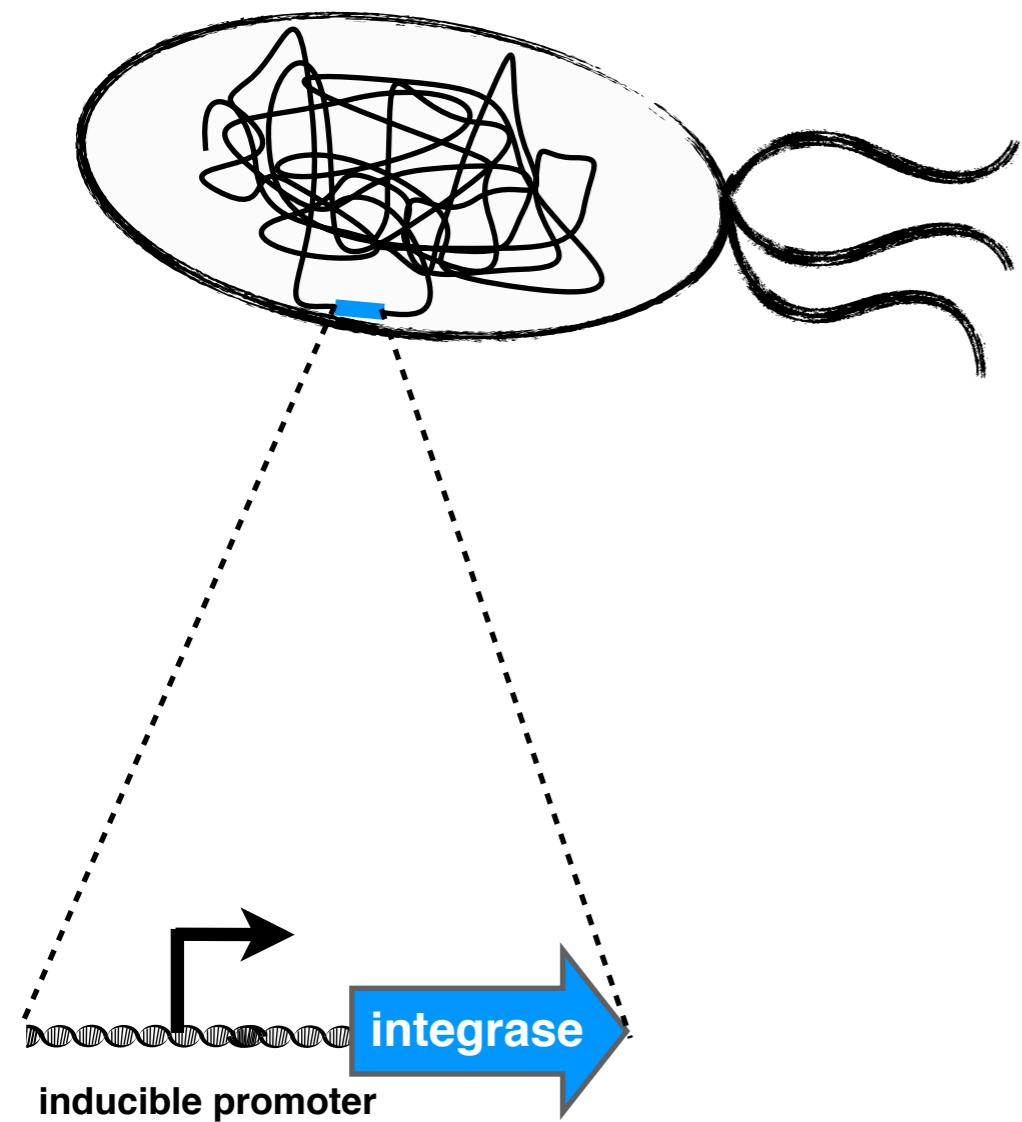
EXCISION



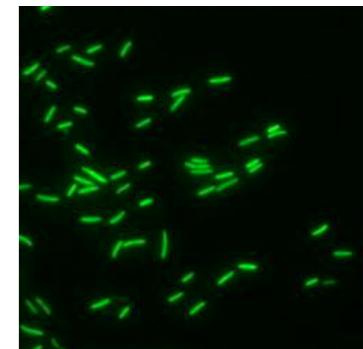
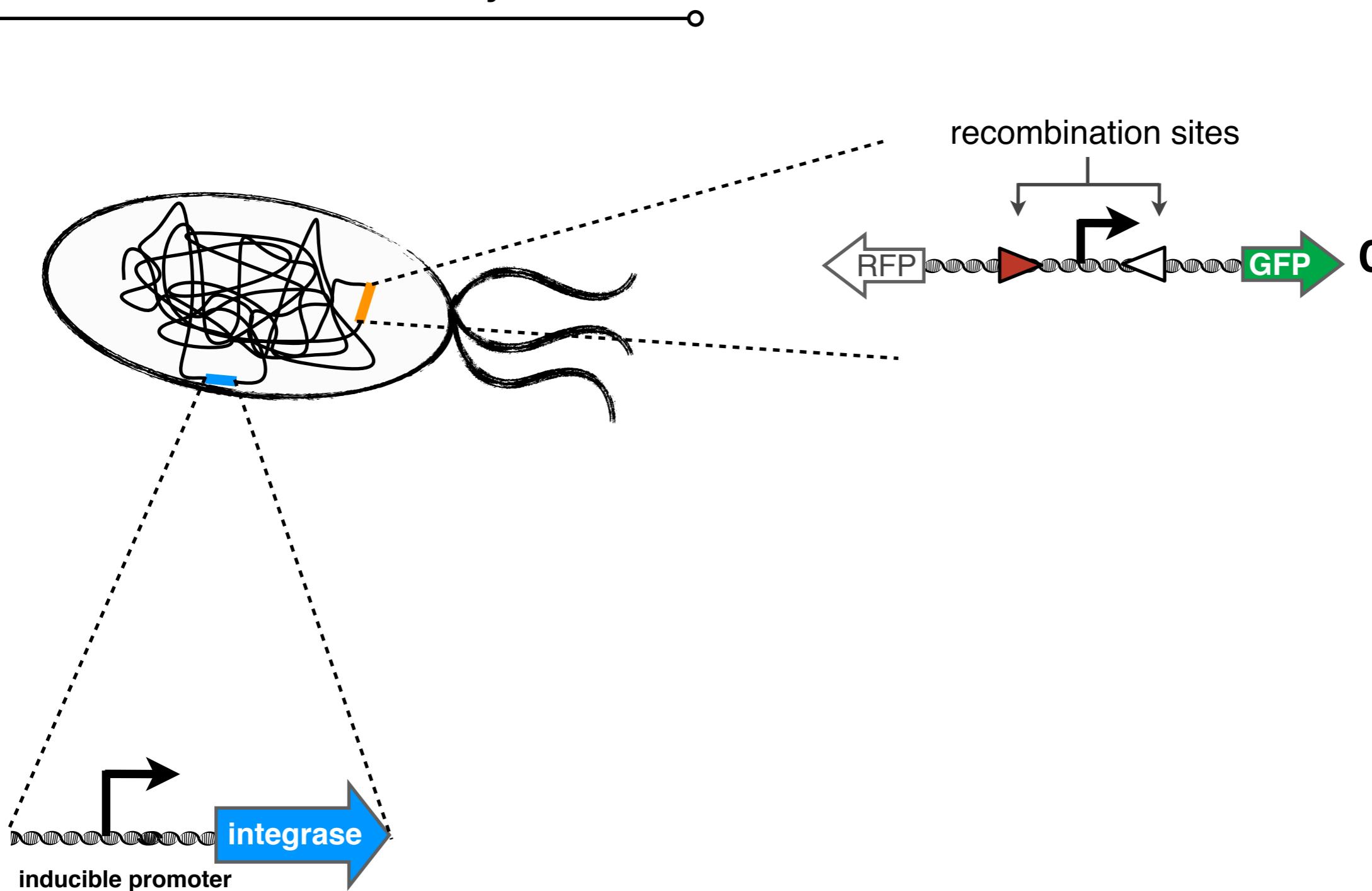
Switches with memory



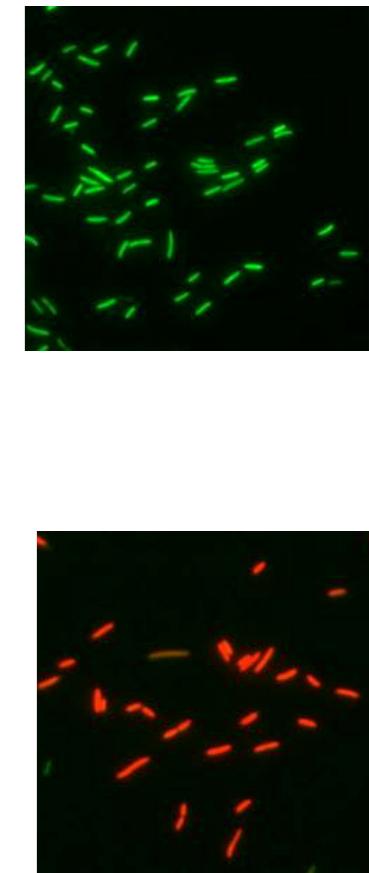
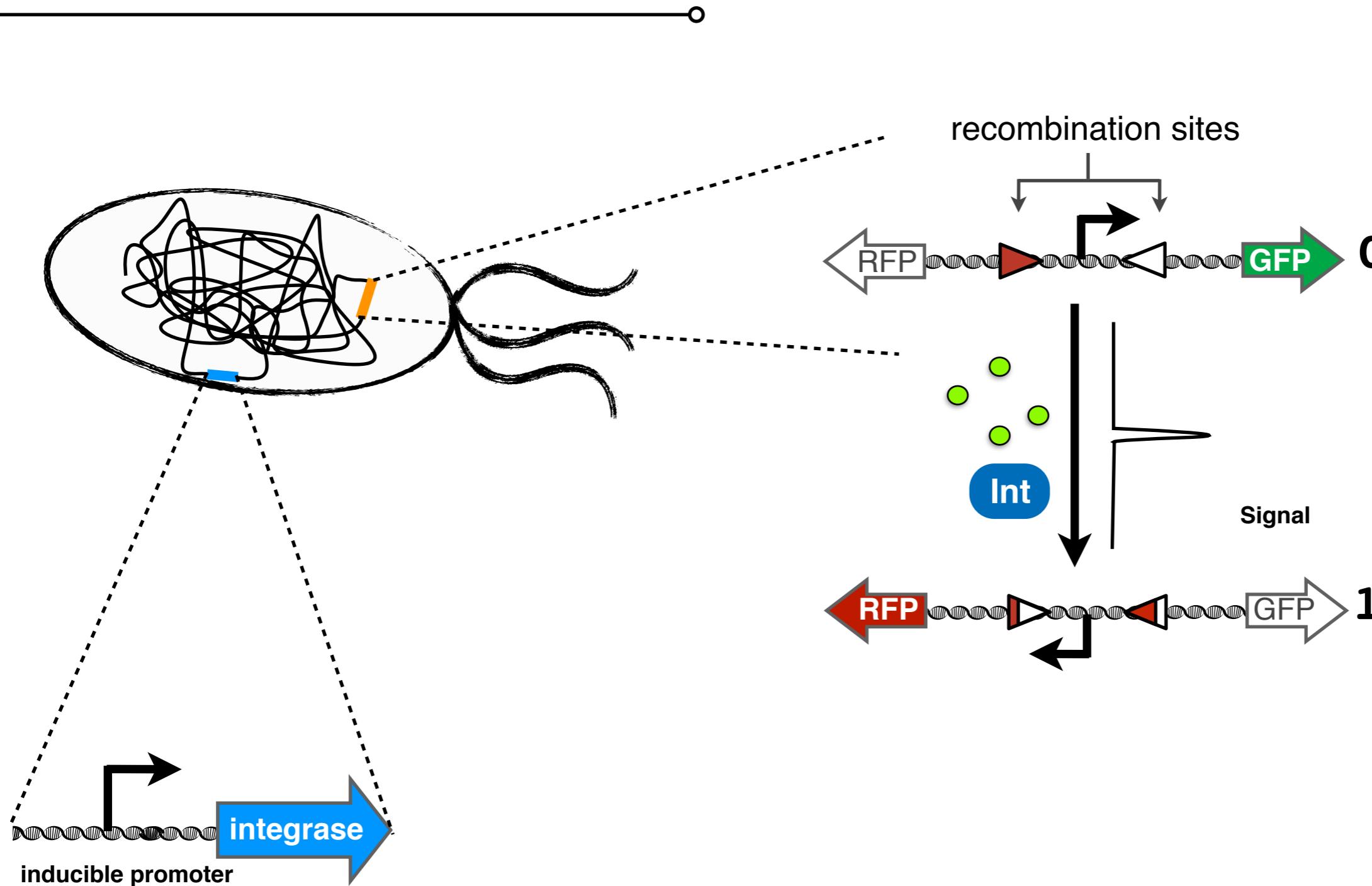
Switches with memory



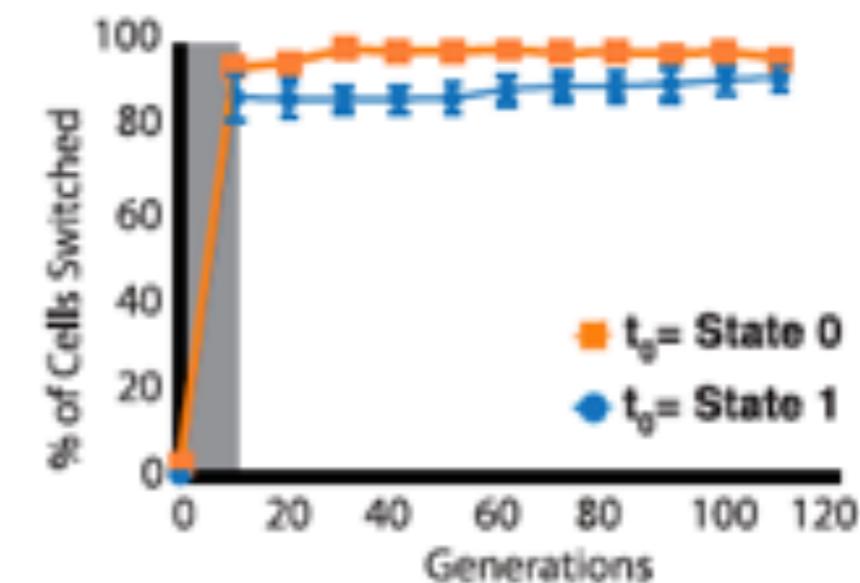
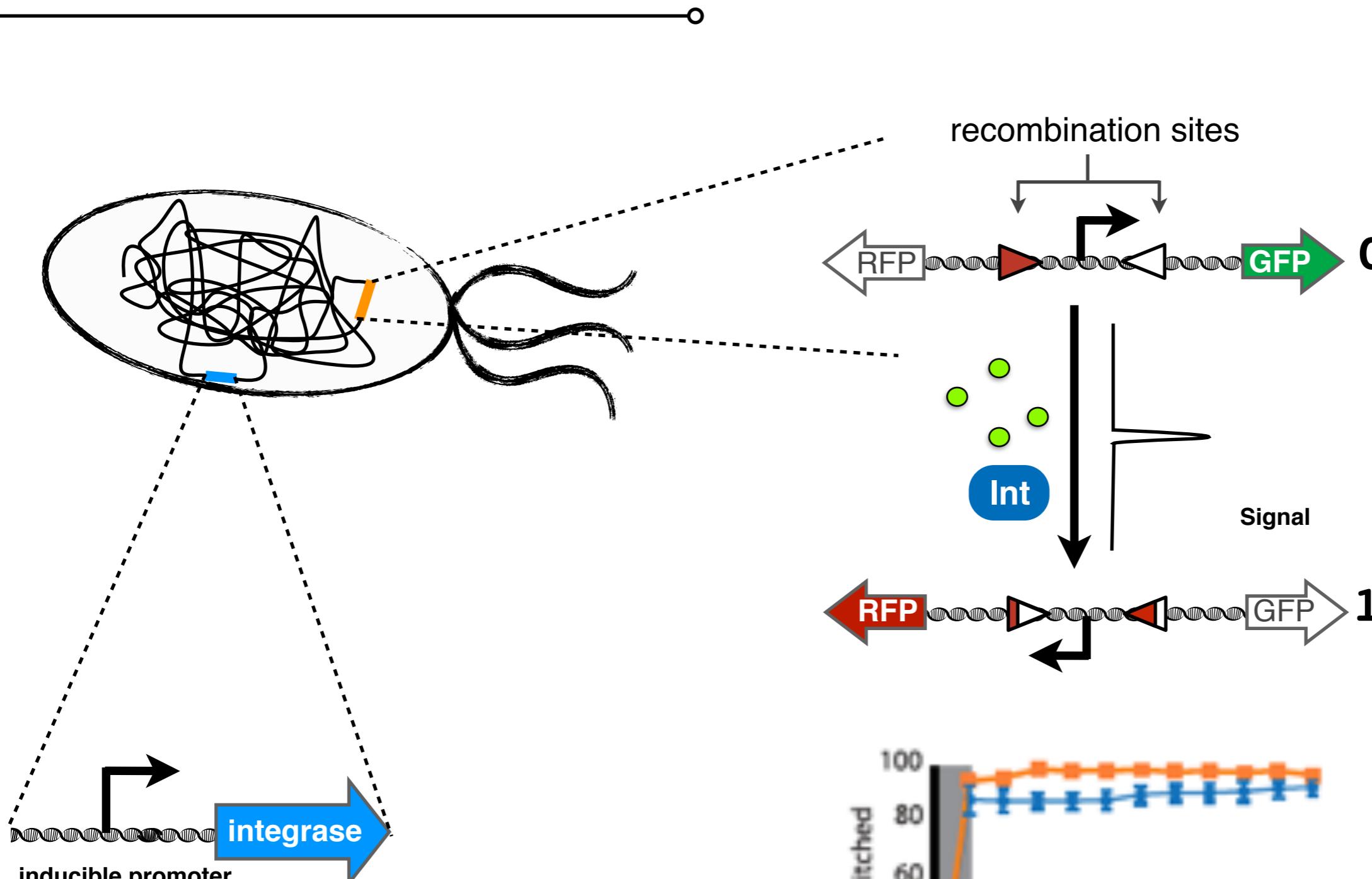
Switches with memory



Switches with memory



Switches with memory

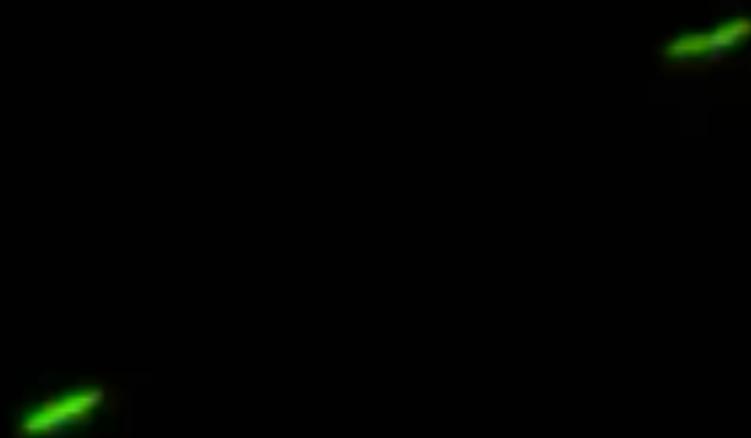


GFP

RFP



Merge



0

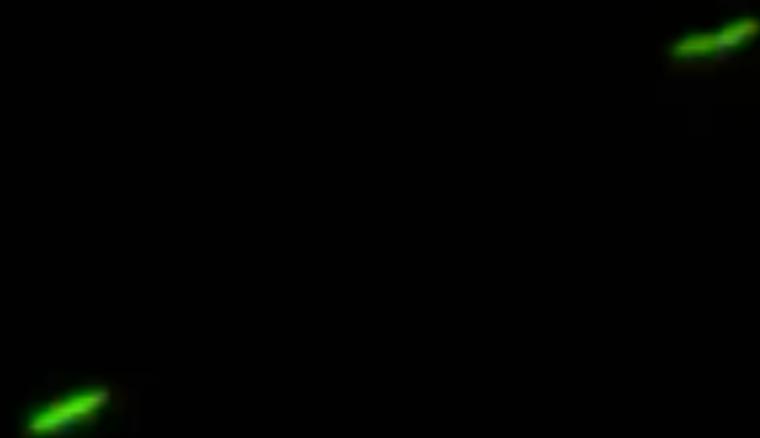
22

GFP

RFP



Merge



0

22

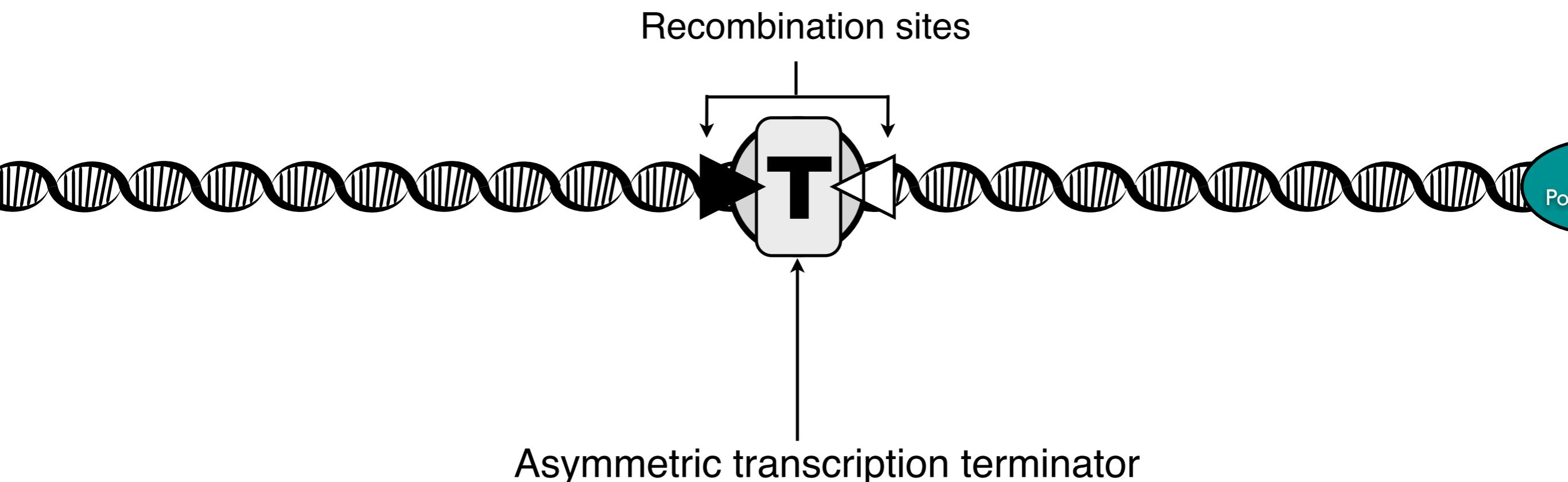
Transcriptor: controlling the flow of RNA polymerase



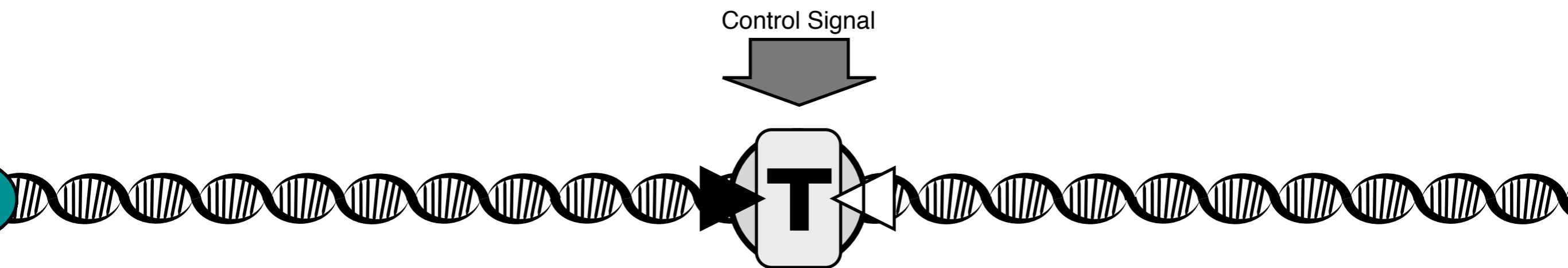
Transcriptor: controlling the flow of RNA polymerase



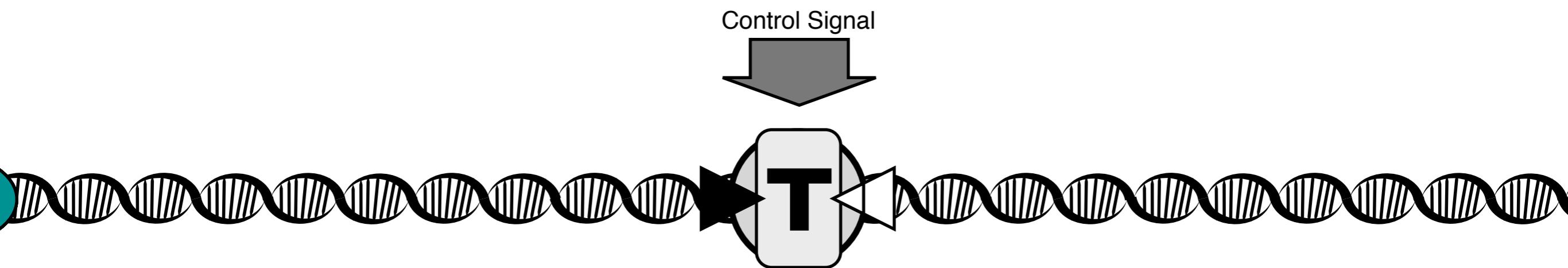
Transcriptor: controlling the flow of RNA polymerase



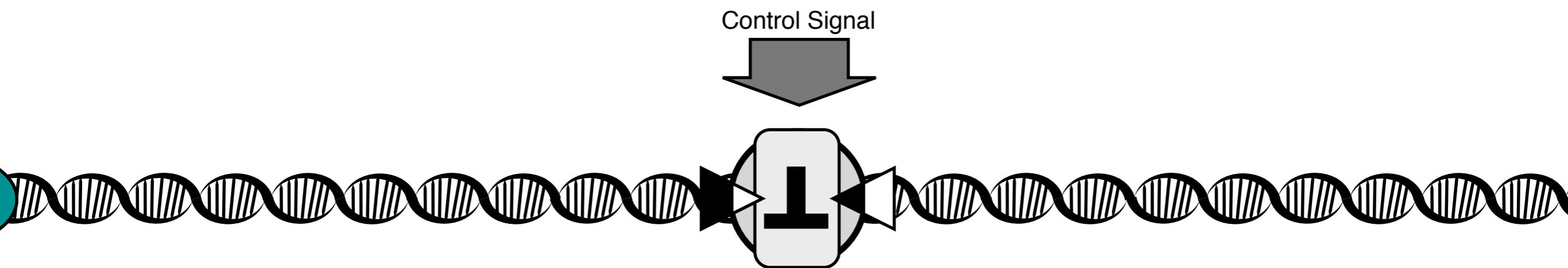
Transcriptor: controlling the flow of RNA polymerase



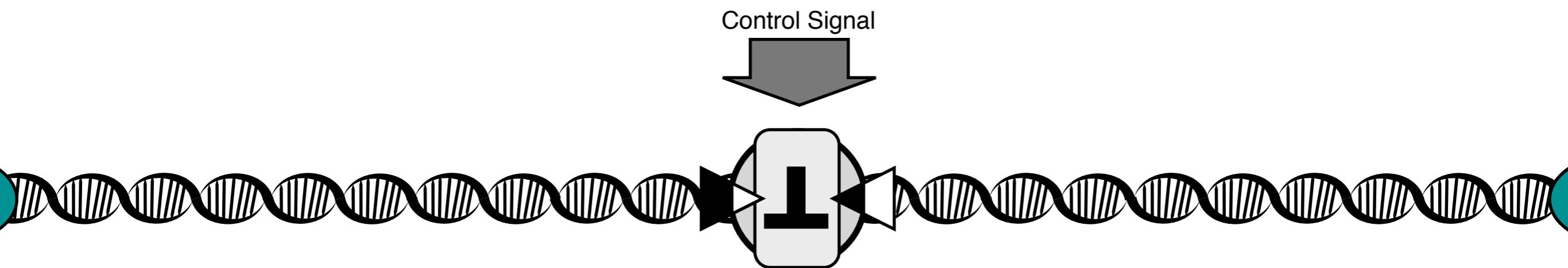
Transcriptor: controlling the flow of RNA polymerase



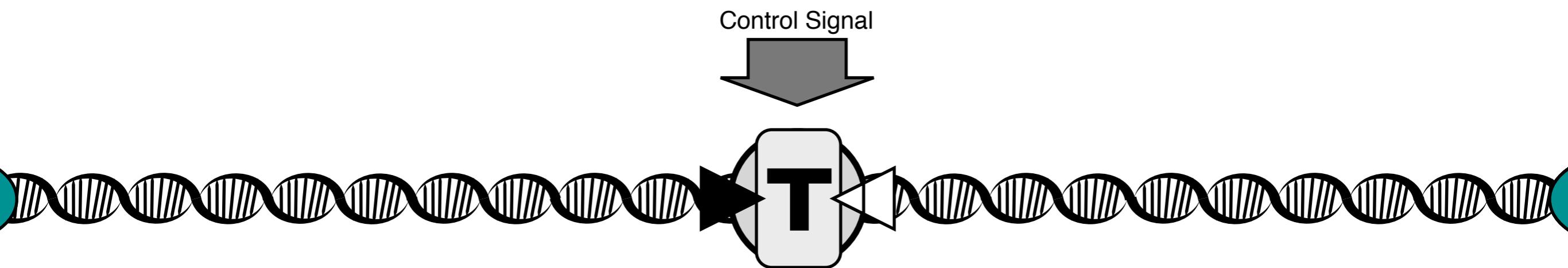
Transcriptor: controlling the flow of RNA polymerase



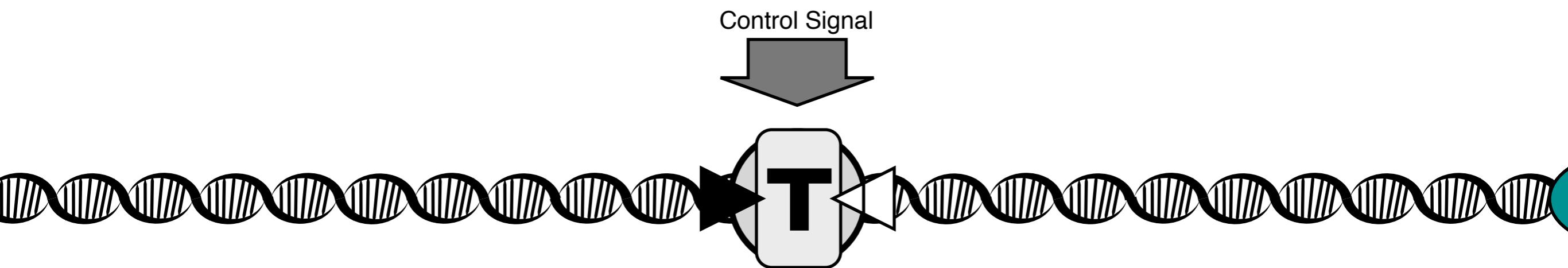
Transcriptor: controlling the flow of RNA polymerase



Transcriptor: controlling the flow of RNA polymerase

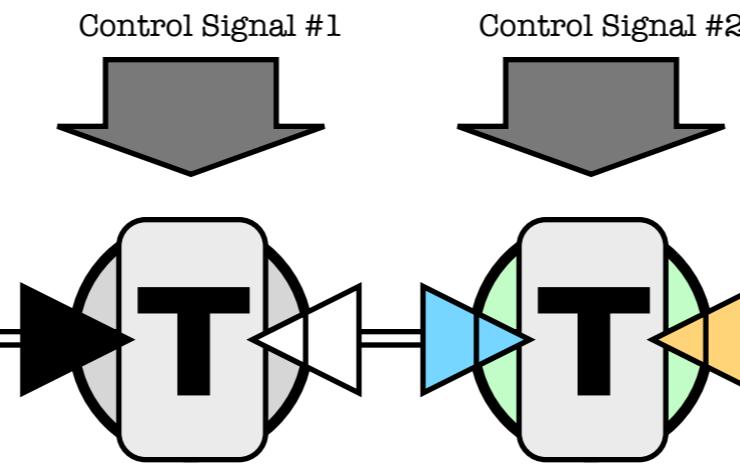


Transcriptor: controlling the flow of RNA polymerase

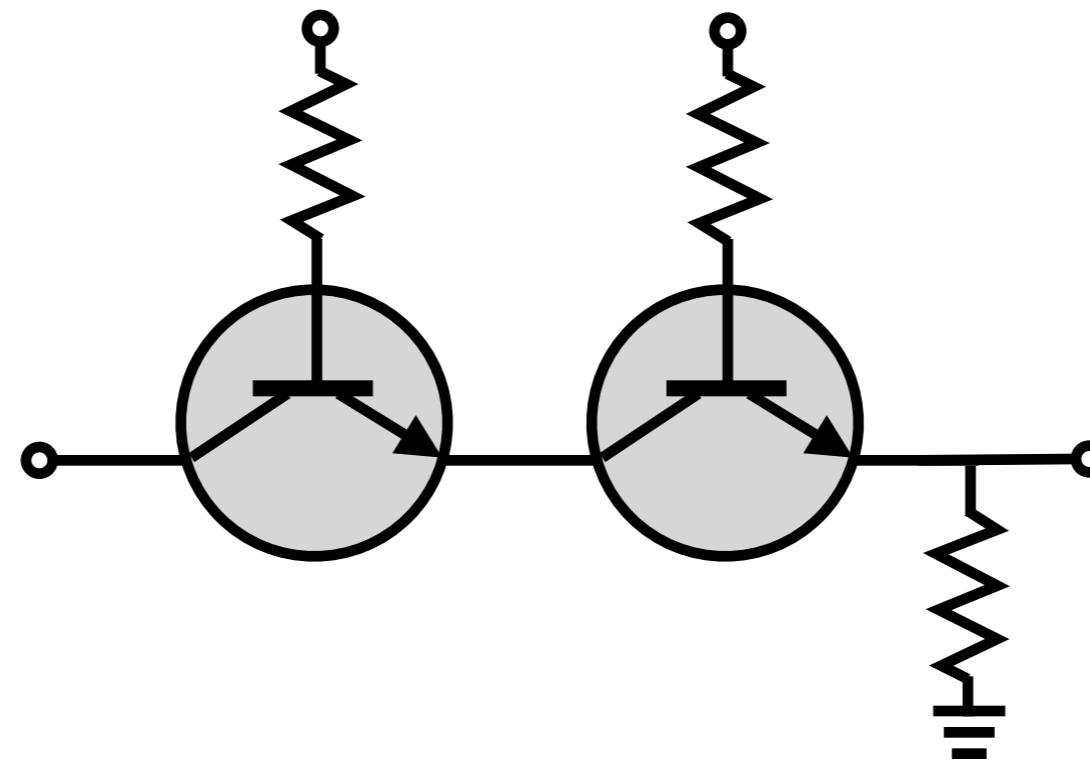
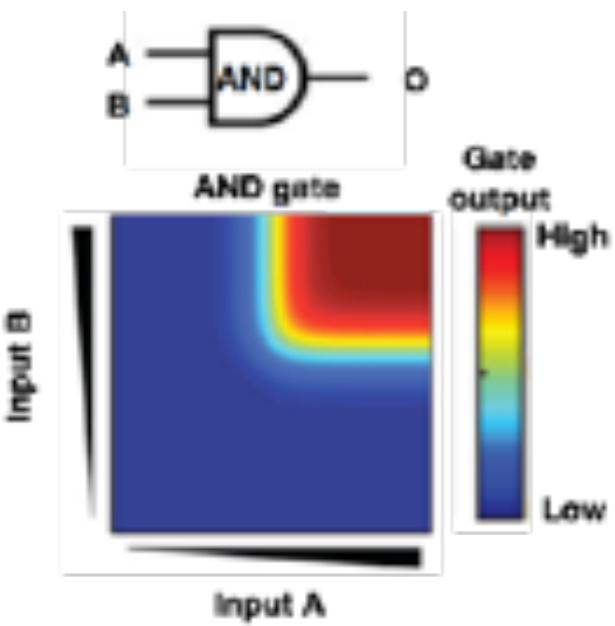


Transcribers can be composed to produce logic functions

AND Gate



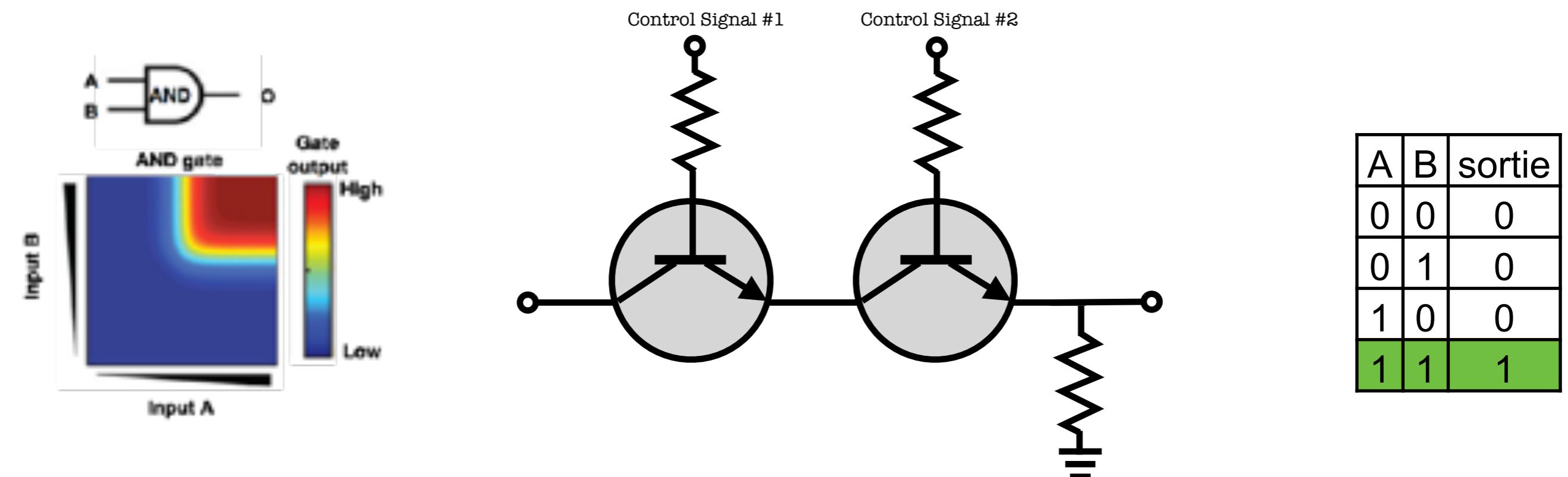
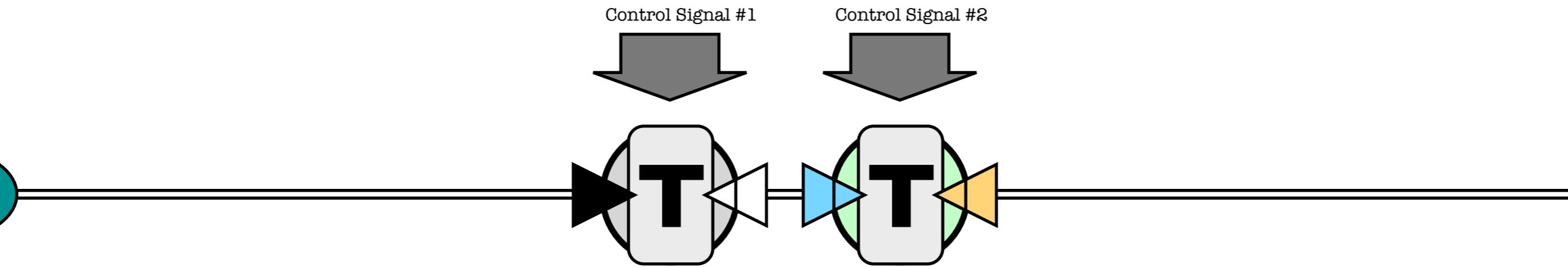
Control Signal #1 Control Signal #2



A	B	sortie
0	0	0
0	1	0
1	0	0
1	1	1

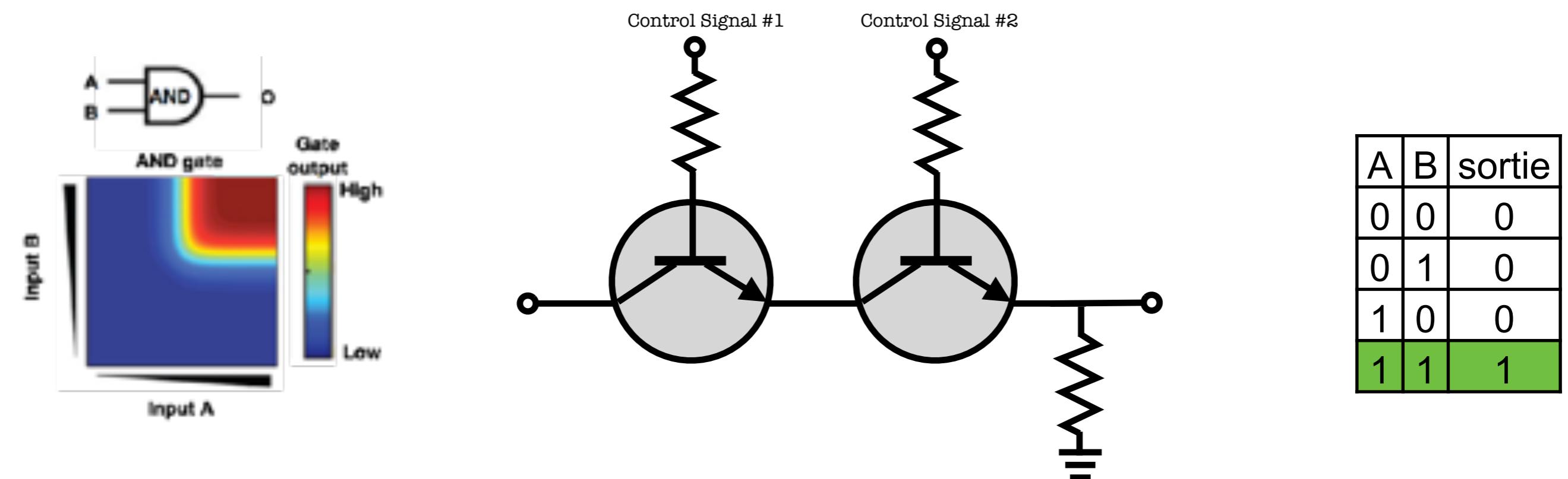
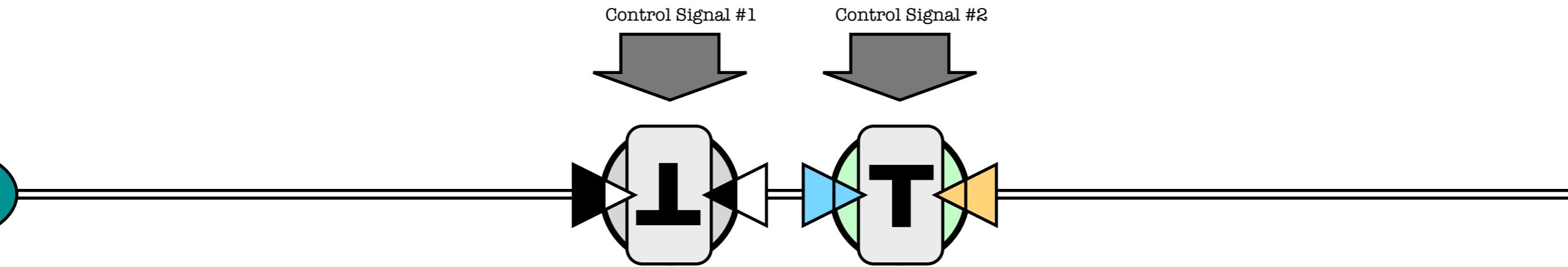
Transcribers can be composed to produce logic functions

AND Gate



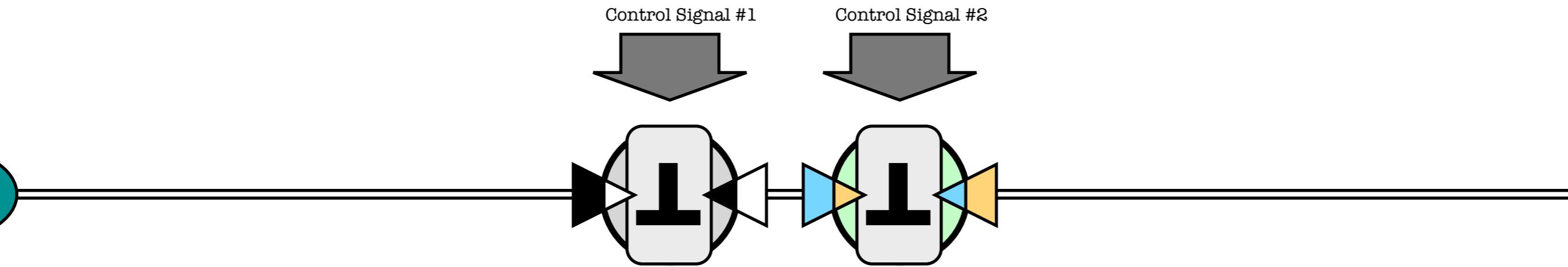
Transcribers can be composed to produce logic functions

AND Gate

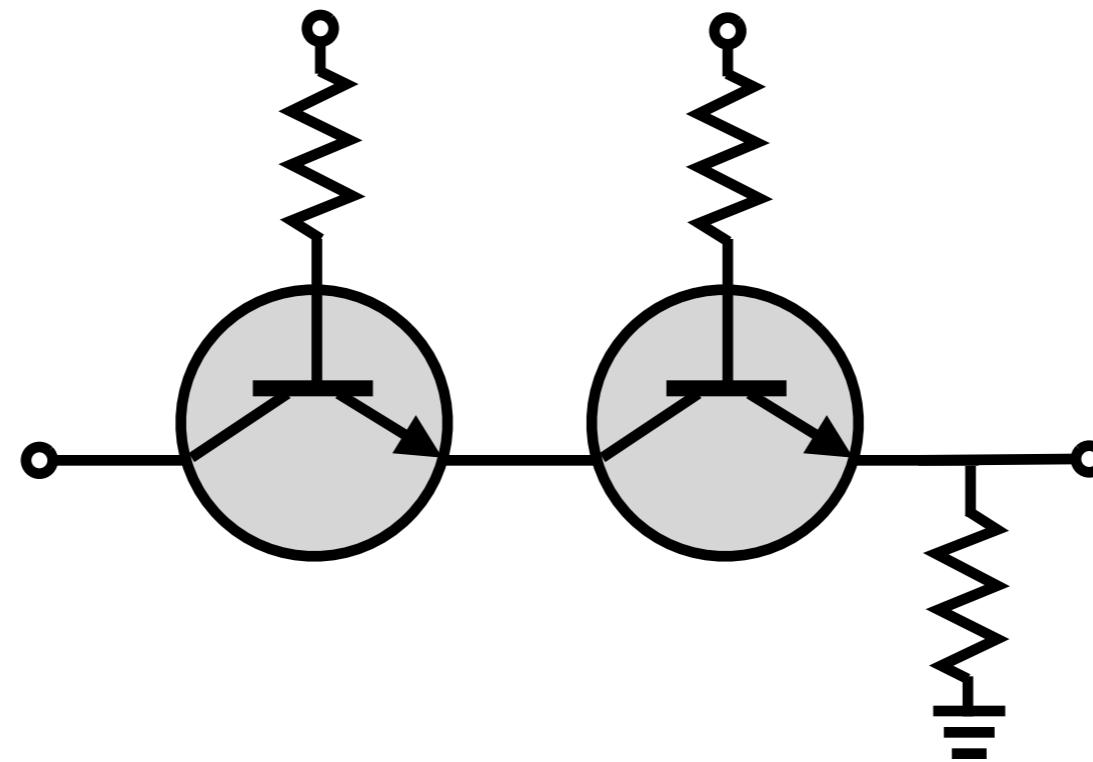
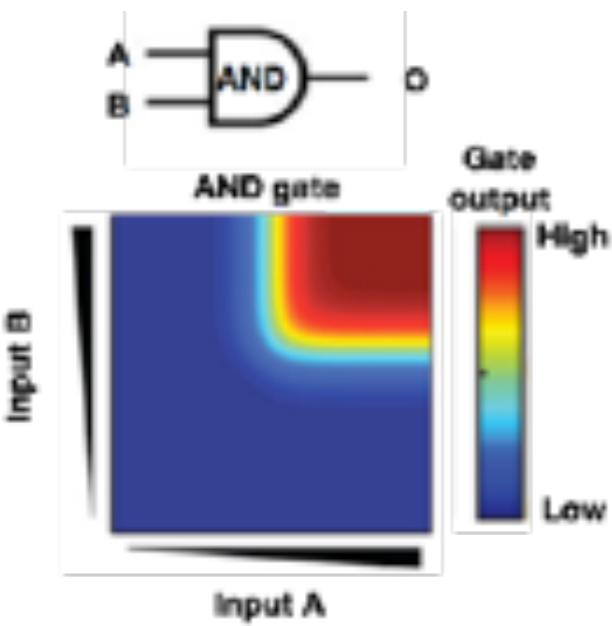


Transcribers can be composed to produce logic functions

AND Gate



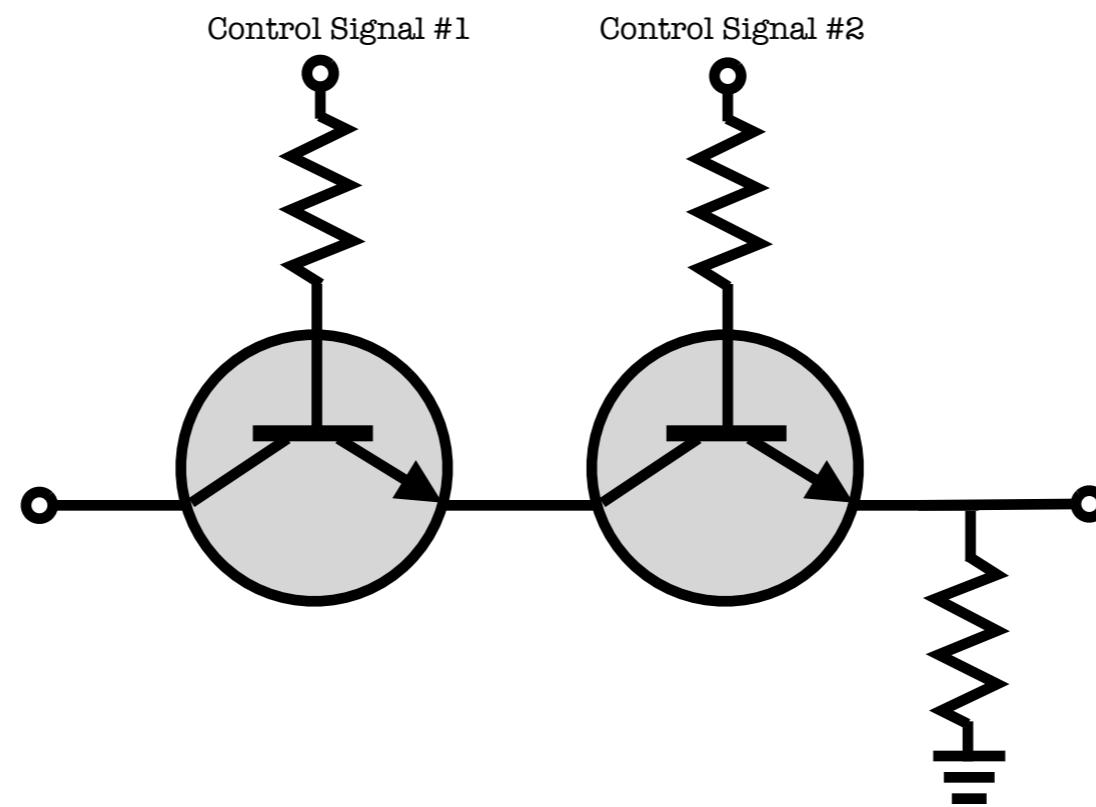
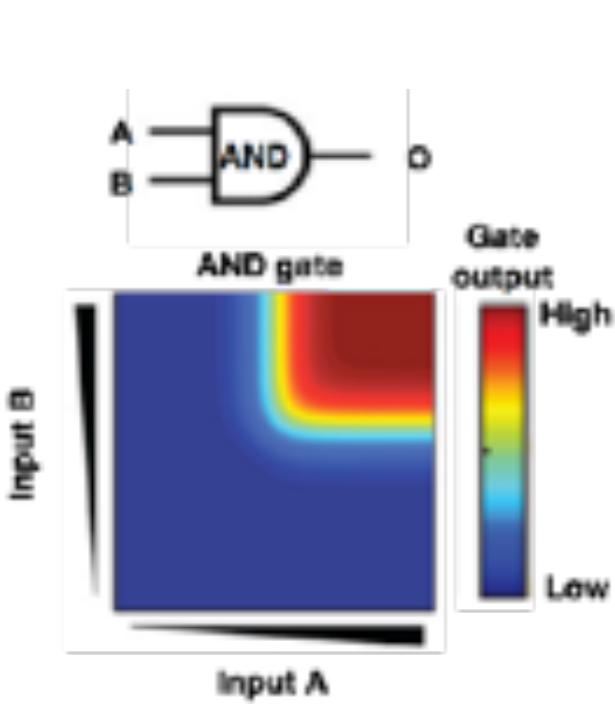
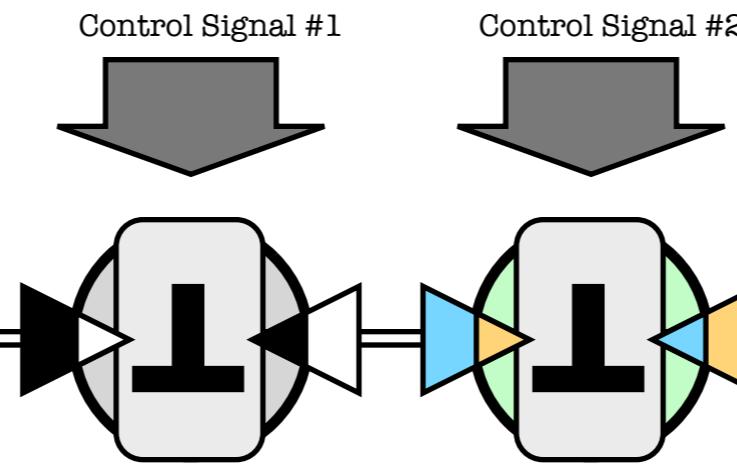
Control Signal #1 Control Signal #2



A	B	sortie
0	0	0
0	1	0
1	0	0
1	1	1

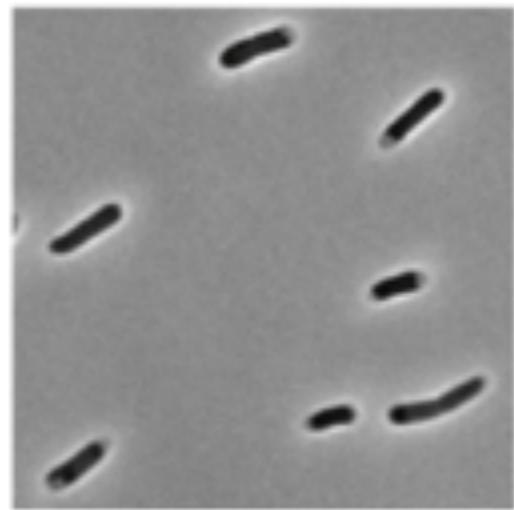
Transcribers can be composed to produce logic functions

AND Gate

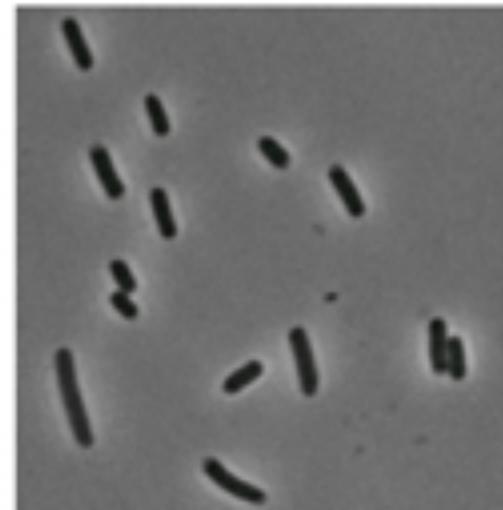


A	B	sortie
0	0	0
0	1	0
1	0	0
1	1	1

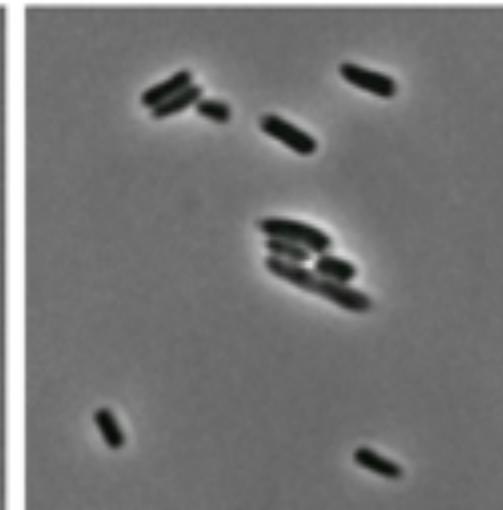
no inducer



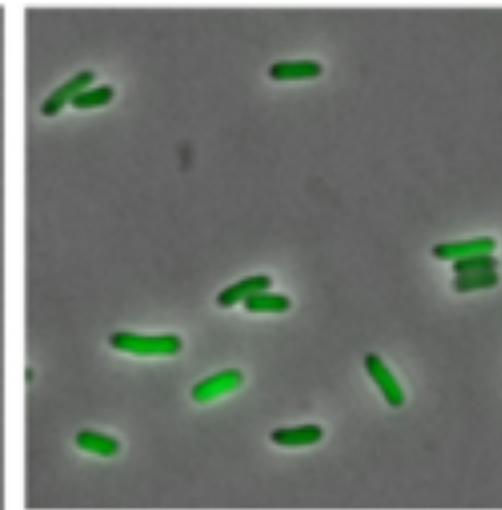
Molecule A



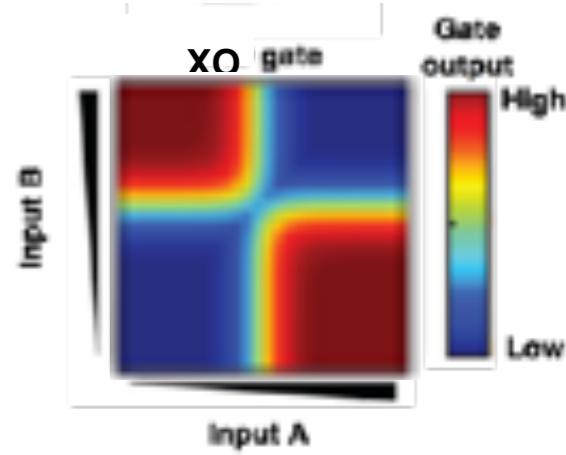
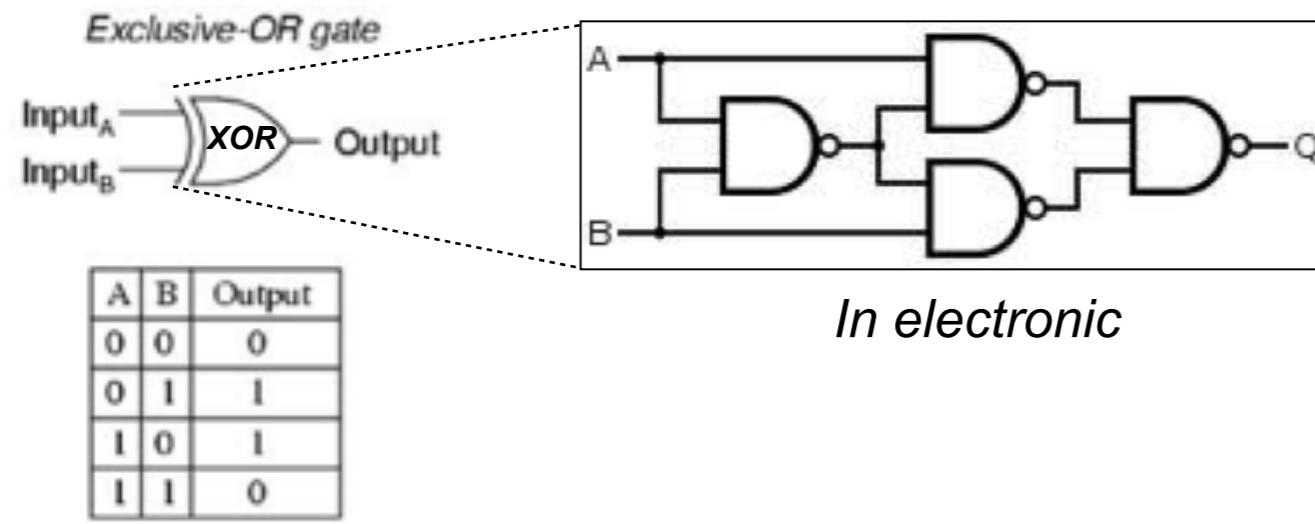
Molecule B



A+ B

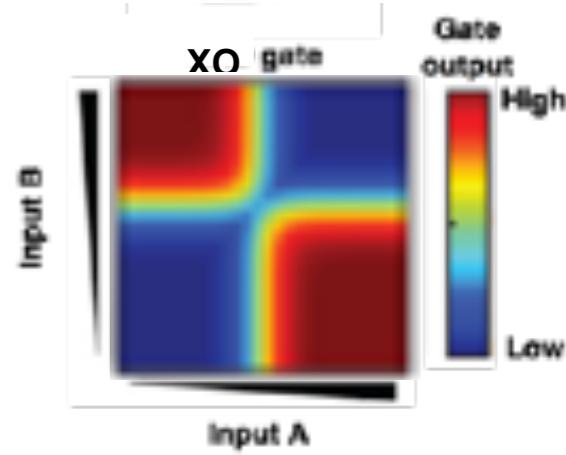
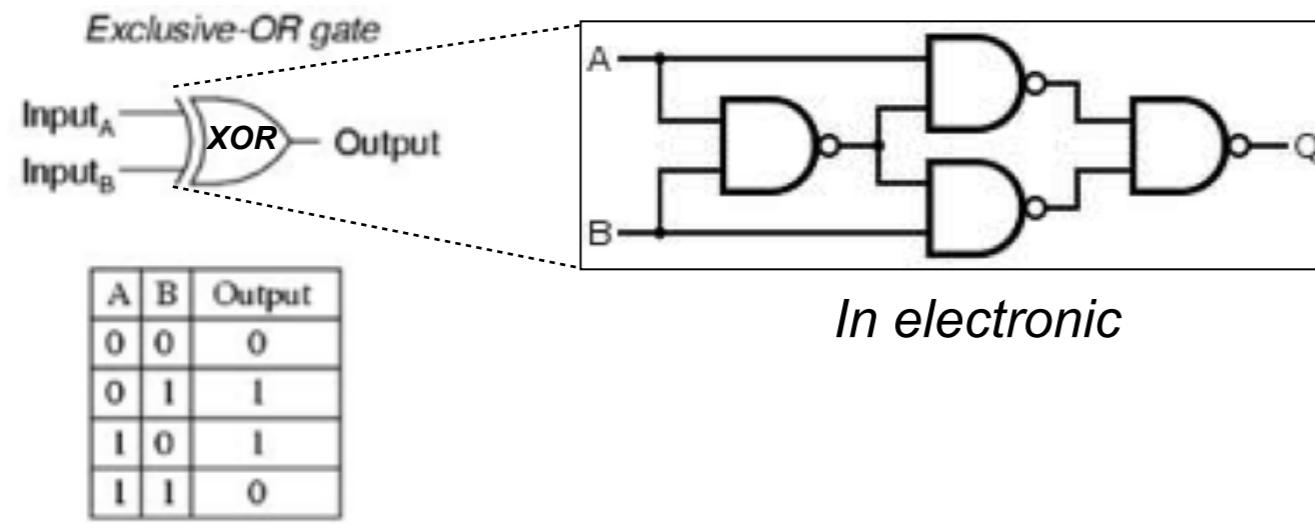


Biology can do better than electronics: example with XOR



A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0

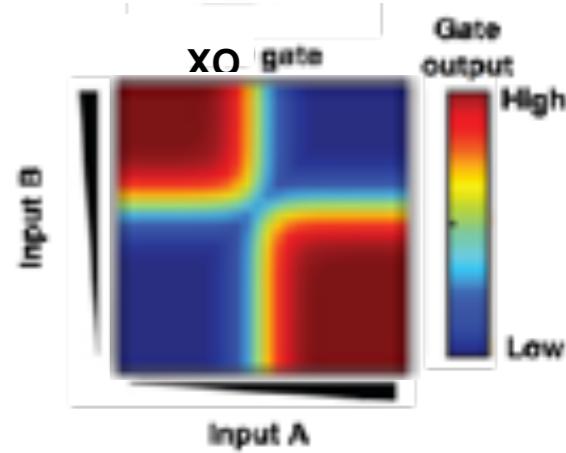
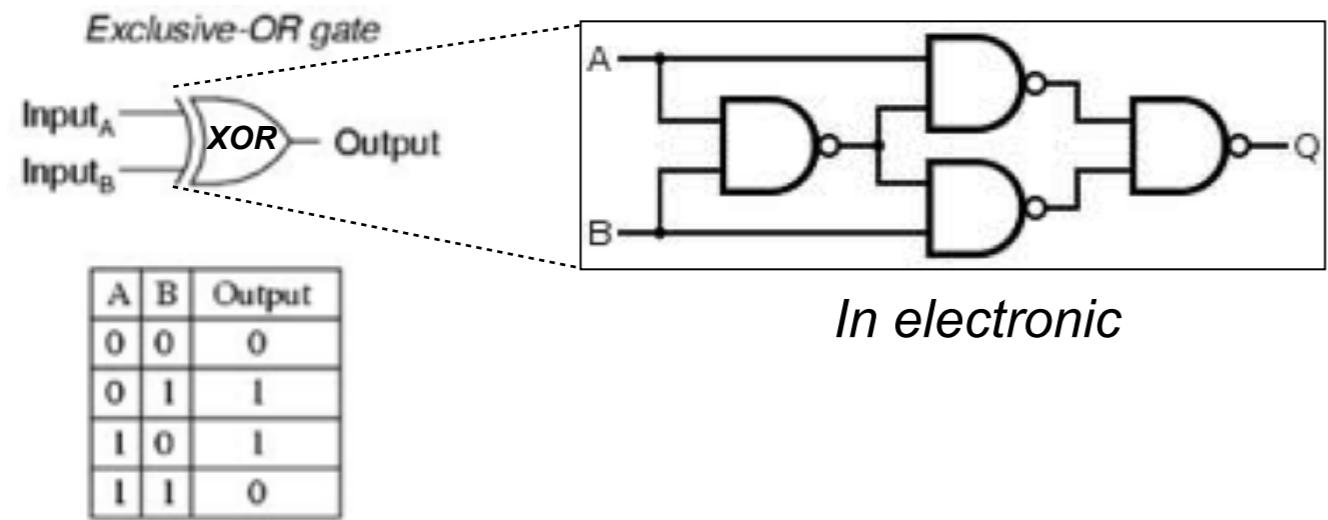
Biology can do better than electronics: example with XOR



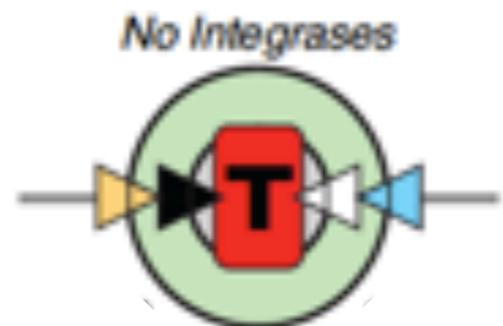
A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0

XOR

Biology can do better than electronics: example with XOR

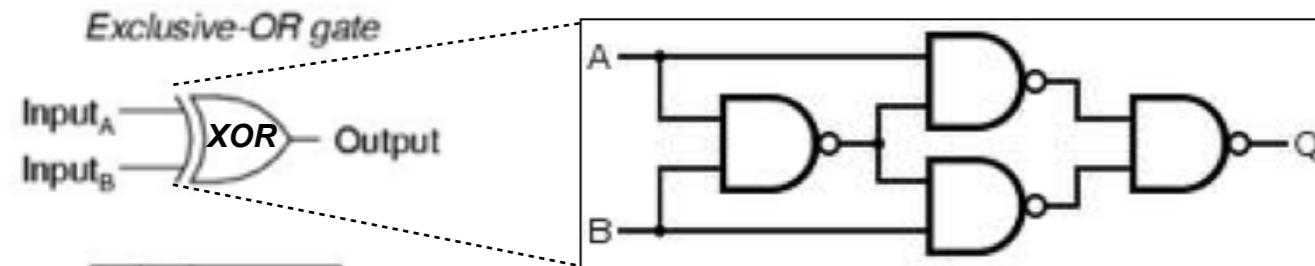


A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0



XOR

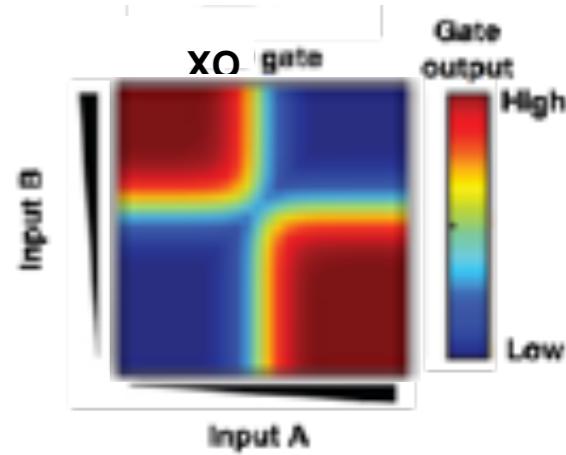
Biology can do better than electronics:
example with XOR



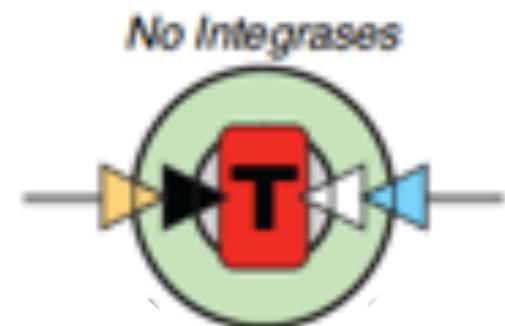
In electronic

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Transcriptor based logic

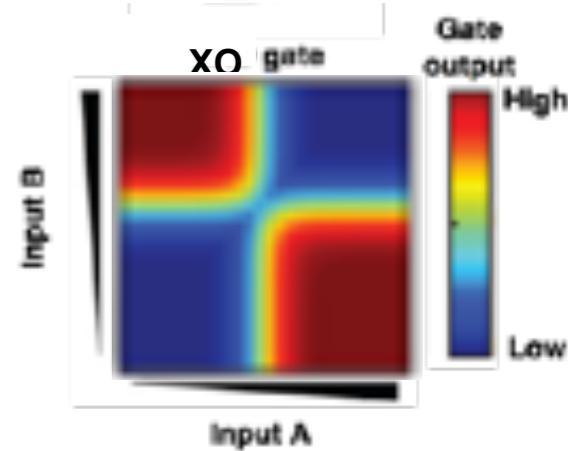
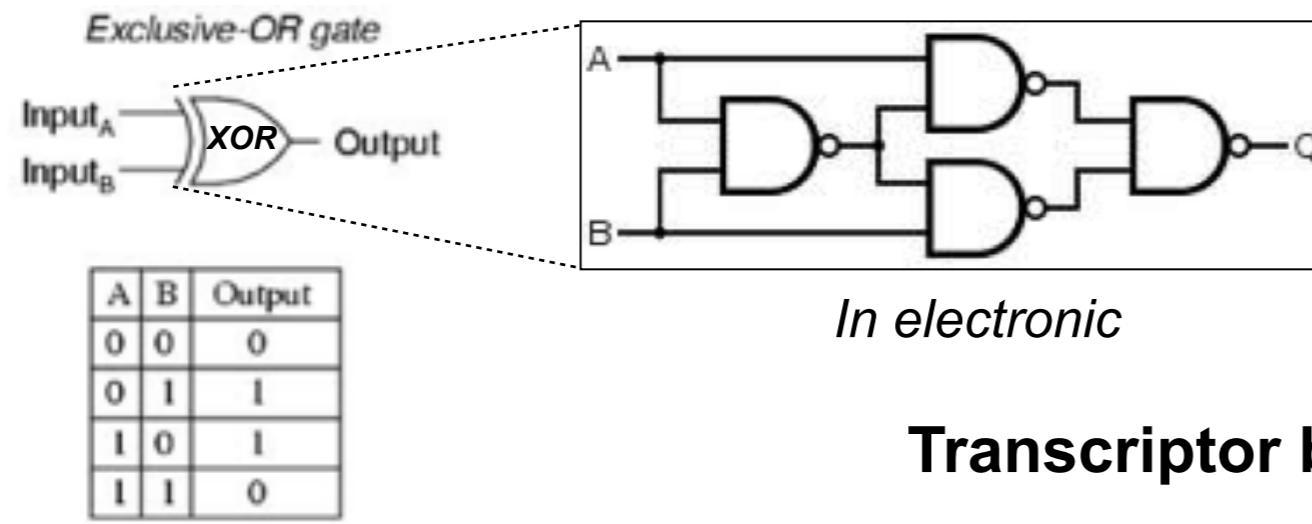


A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0



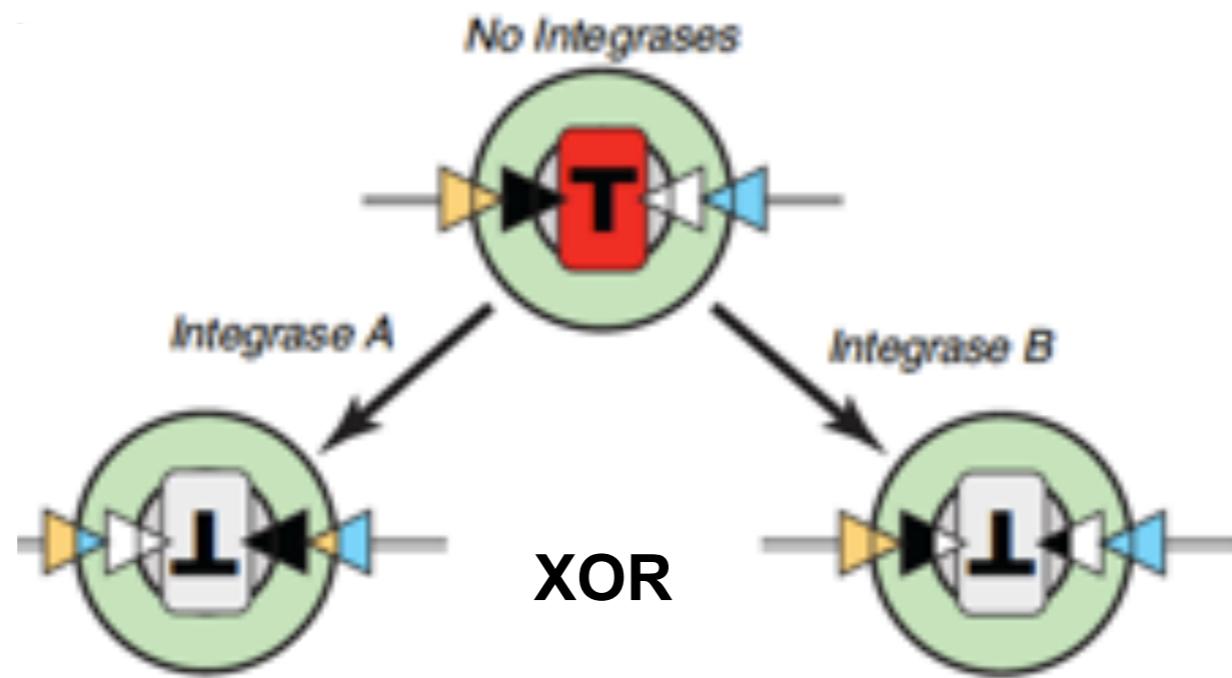
XOR

Biology can do better than electronics: example with XOR

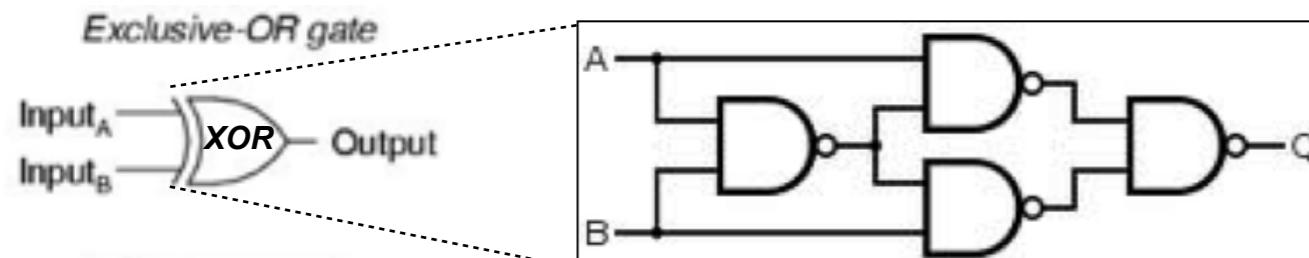


Transcriptor based logic

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0

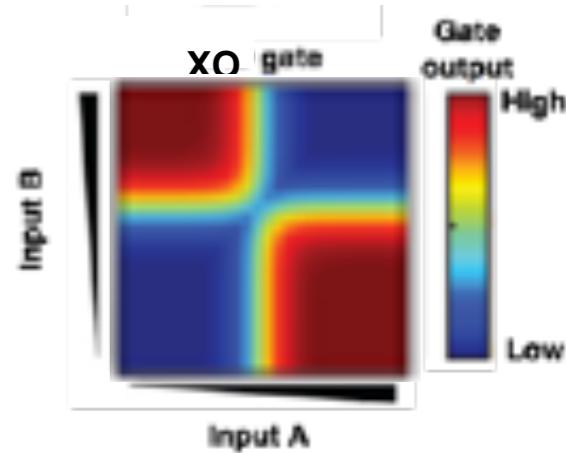


Biology can do better than electronics: example with XOR



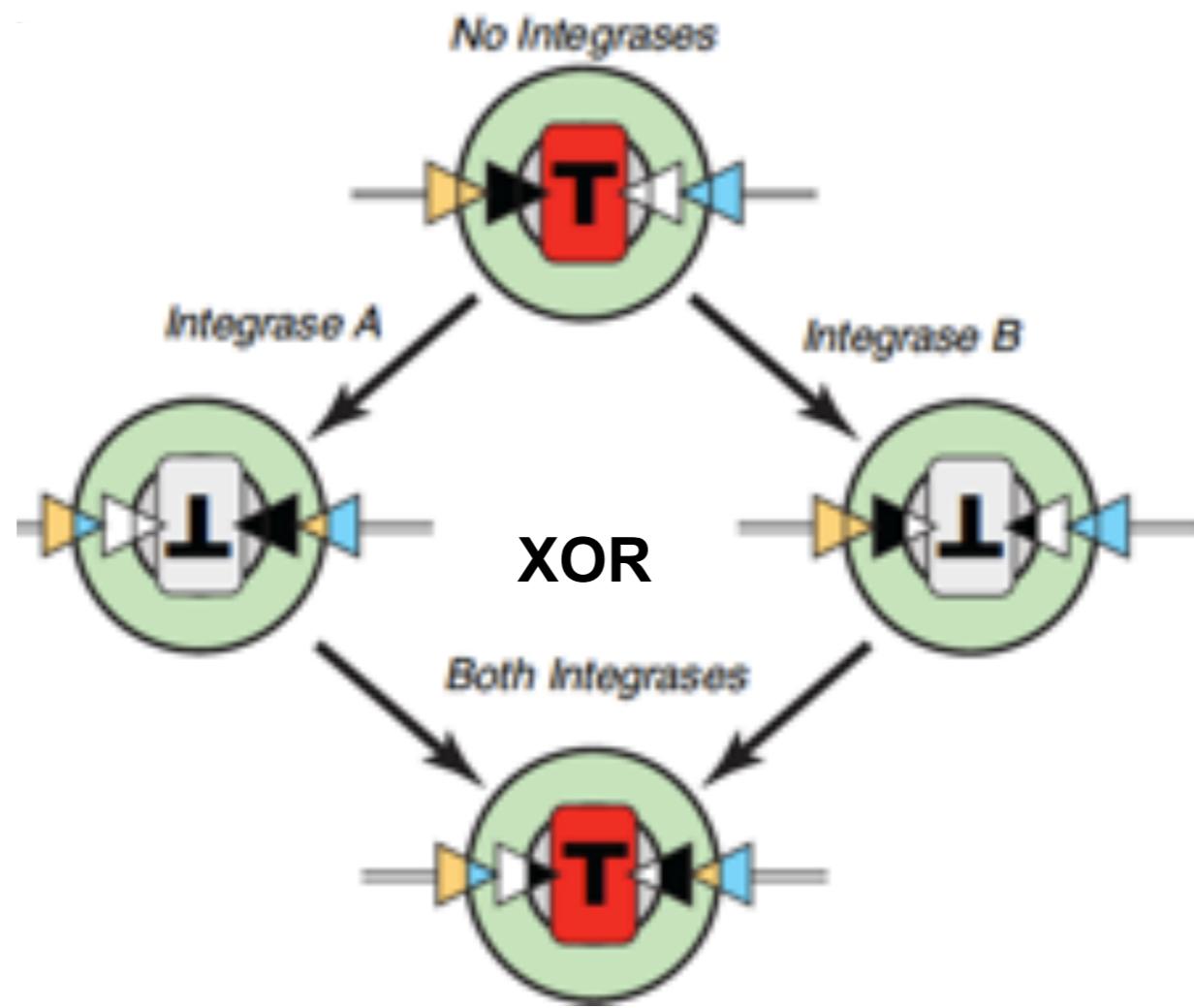
In electronic

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



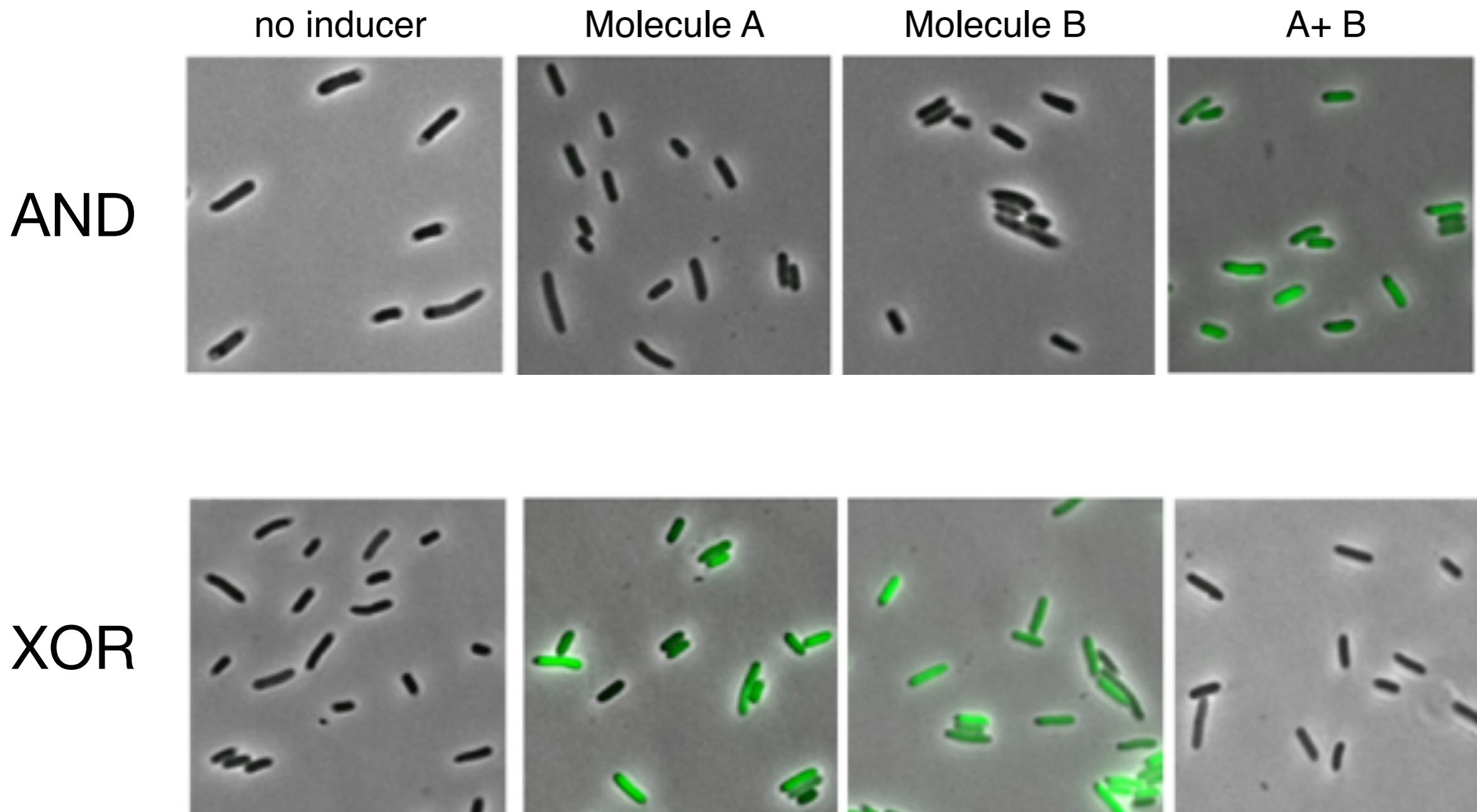
Transcriptor based logic

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0



Amplifying Genetic Logic Gates

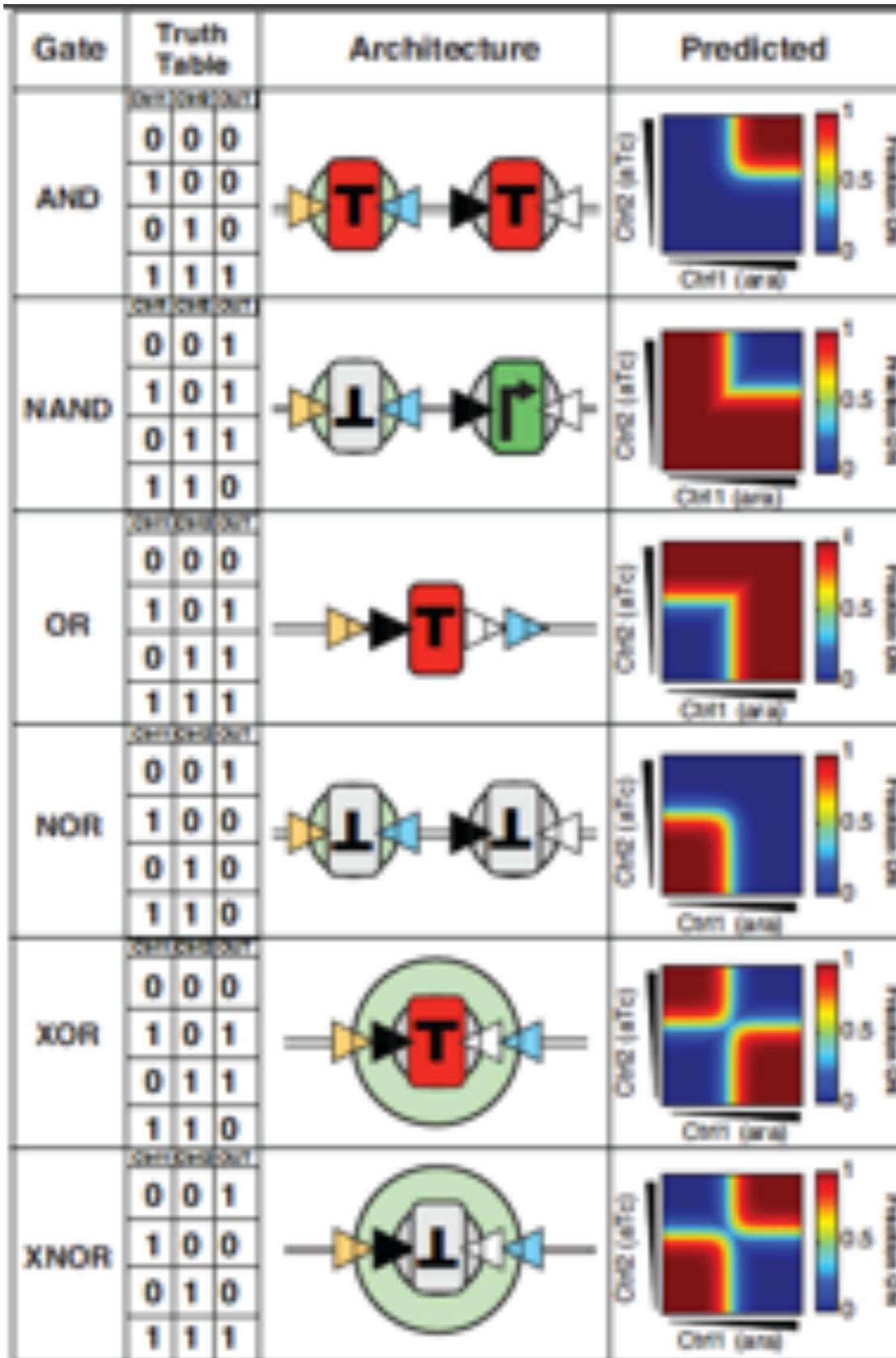
Jerome Bonnet, Peter Yin,* Monica E. Ortiz, Pakpoom Subsoontorn, Drew Endy†



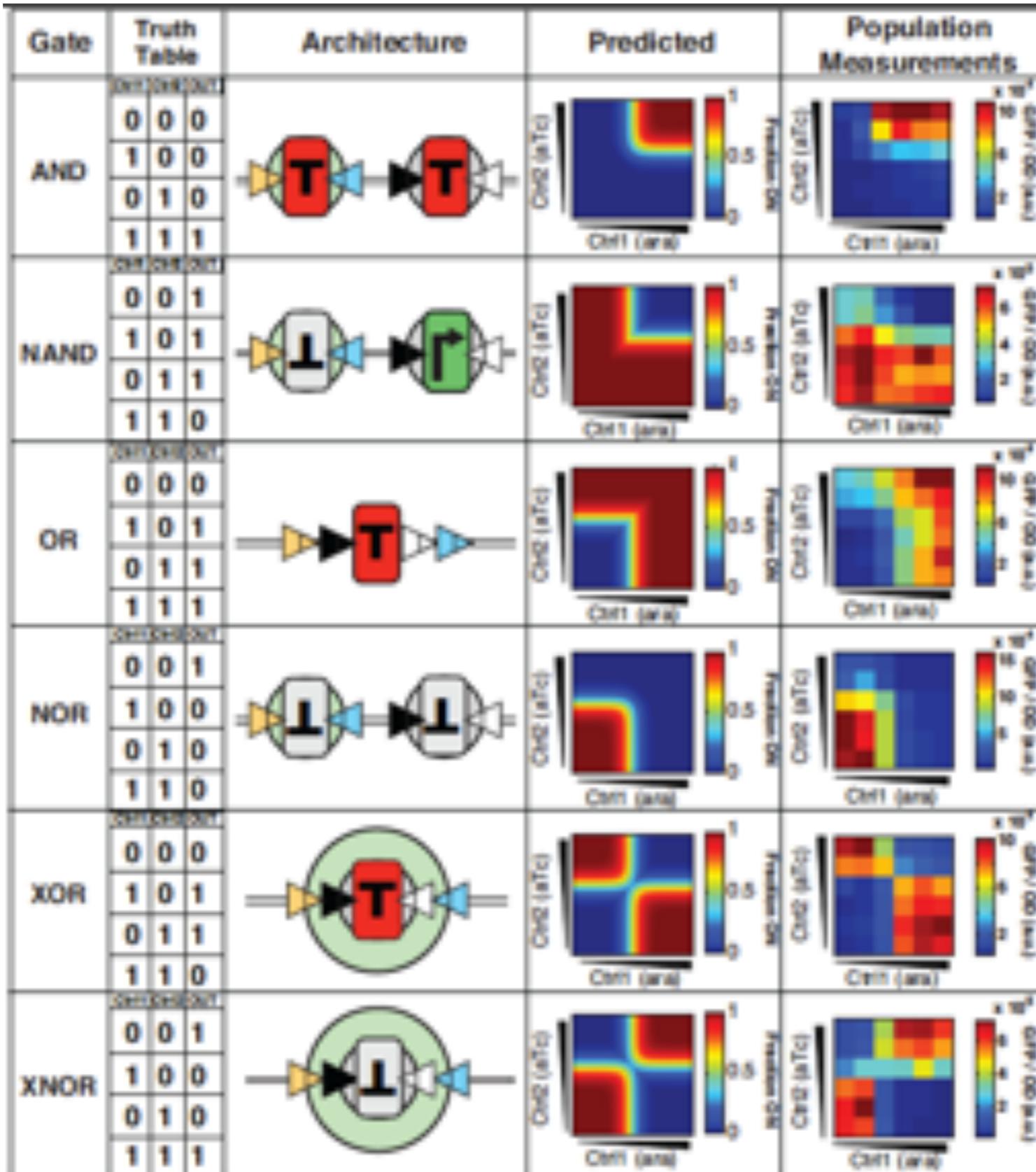
Boolean Integrase Logic (BIL) gates

Gate	Truth Table	Architecture												
AND	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	0	0	0	1	0	0	0	1	0	1	1	1	
0	0	0												
1	0	0												
0	1	0												
1	1	1												
NAND	<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	0	0	1	1	0	1	0	1	1	1	1	0	
0	0	1												
1	0	1												
0	1	1												
1	1	0												
OR	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	0	0	0	1	0	1	0	1	1	1	1	1	
0	0	0												
1	0	1												
0	1	1												
1	1	1												
NOR	<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	0	0	1	1	0	0	0	1	0	1	1	0	
0	0	1												
1	0	0												
0	1	0												
1	1	0												
XOR	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	0	0	0	1	0	1	0	1	1	1	1	0	
0	0	0												
1	0	1												
0	1	1												
1	1	0												
XNOR	<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	0	0	1	1	0	0	0	1	0	1	1	1	
0	0	1												
1	0	0												
0	1	0												
1	1	1												

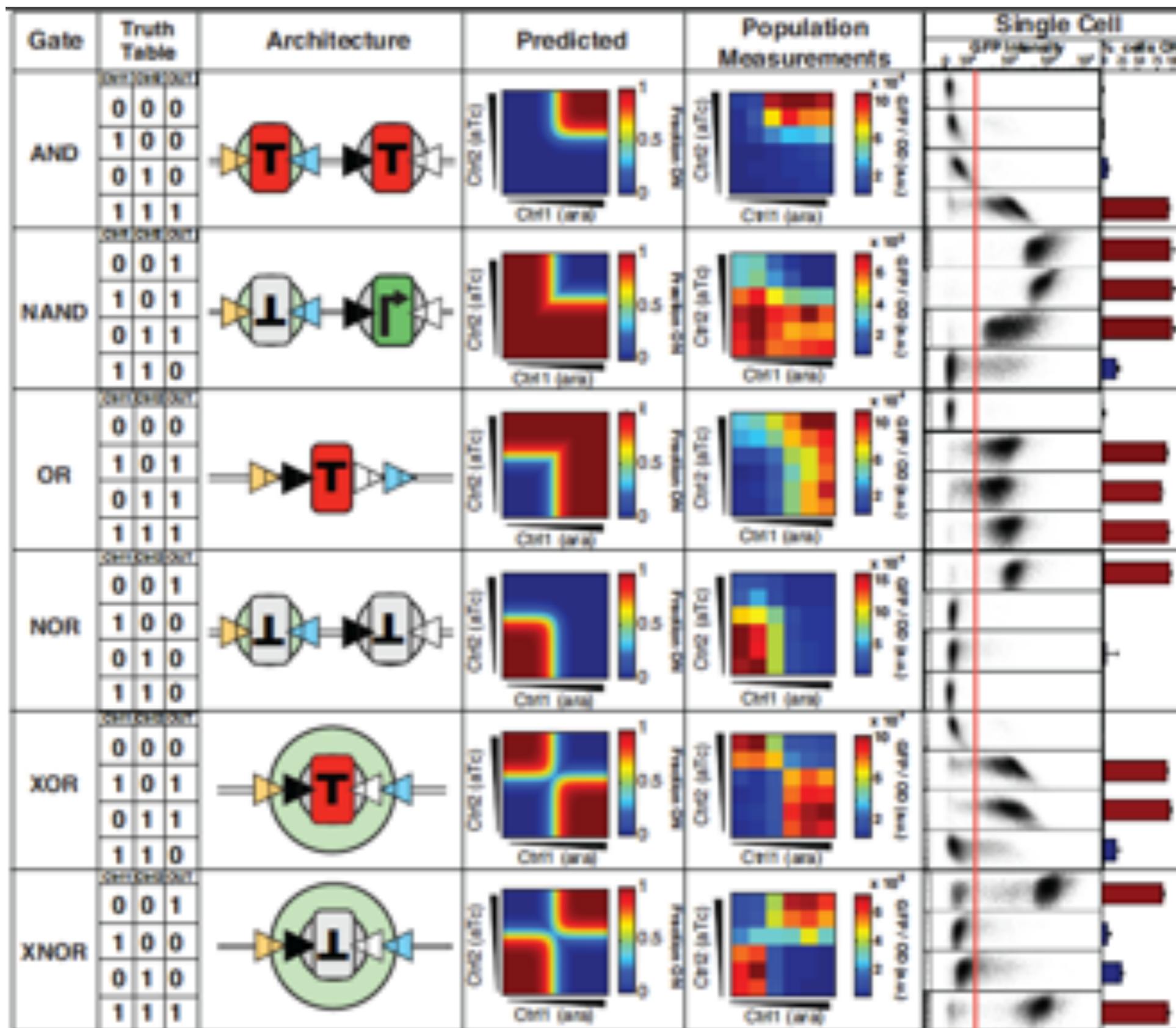
Boolean Integrase Logic (BIL) gates



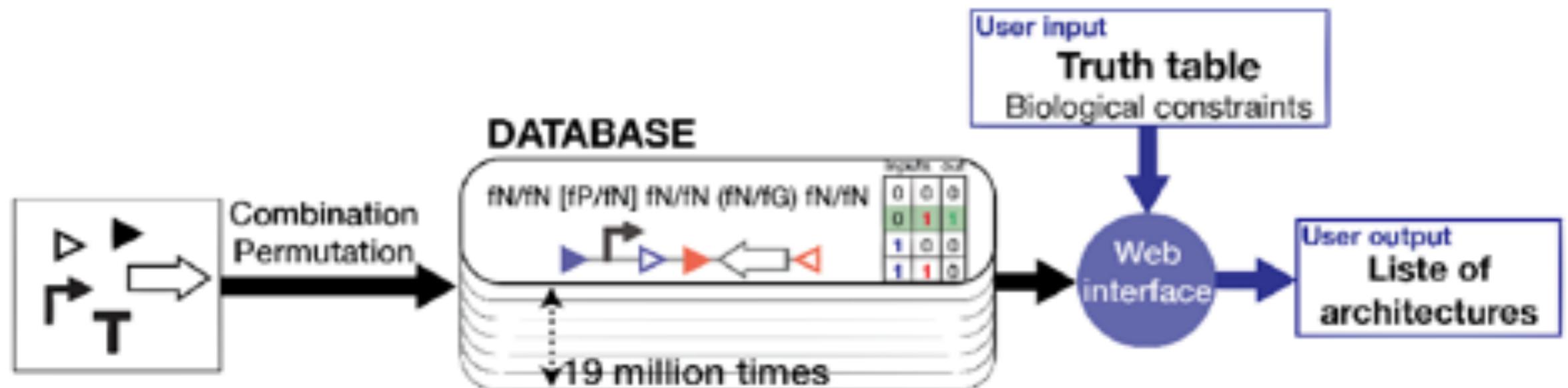
Boolean Integrase Logic (BIL) gates



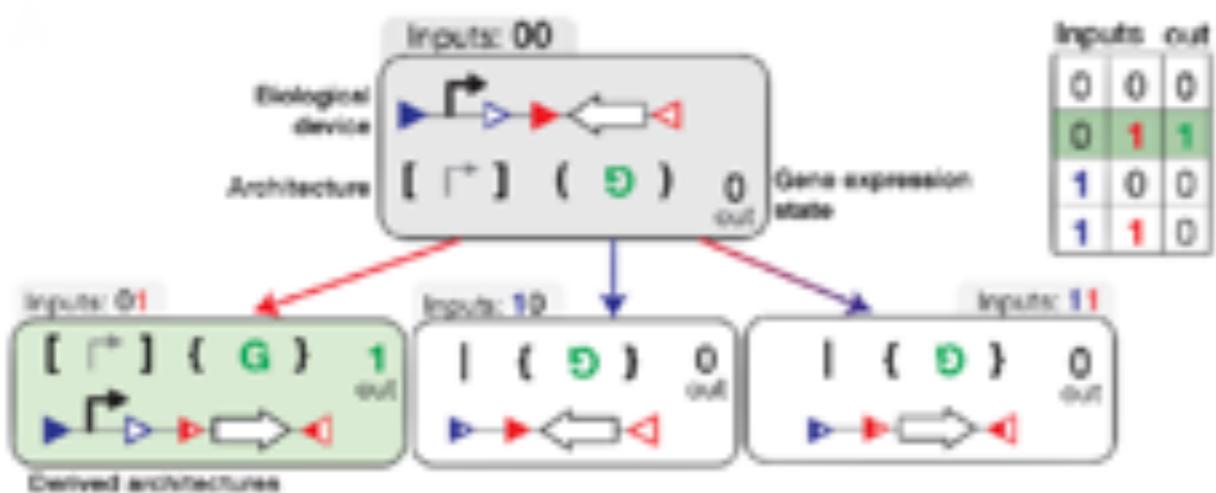
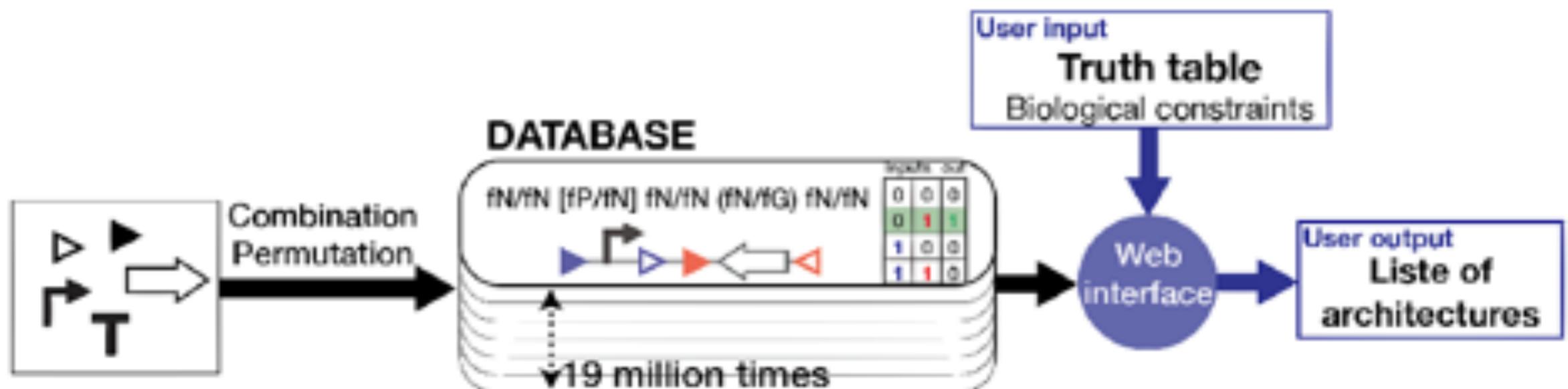
Boolean Integrase Logic (BIL) gates



Automated design of Boolean Integrase Logic



Automated design of Boolean Integrase Logic



In Nature : division of labor

(a)



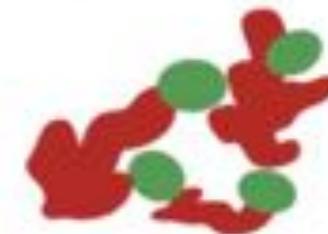
fungi alone



algae alone



complete lichen



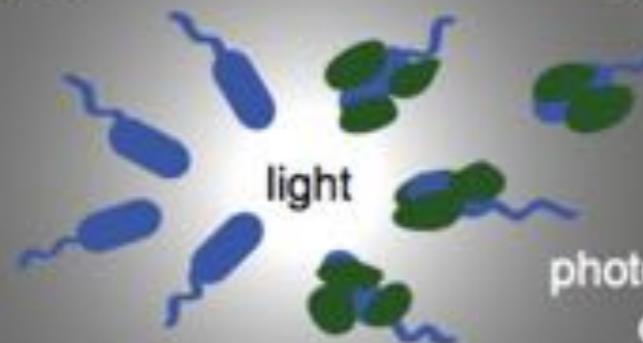
during dessication, starvation, and irradiation

(c)



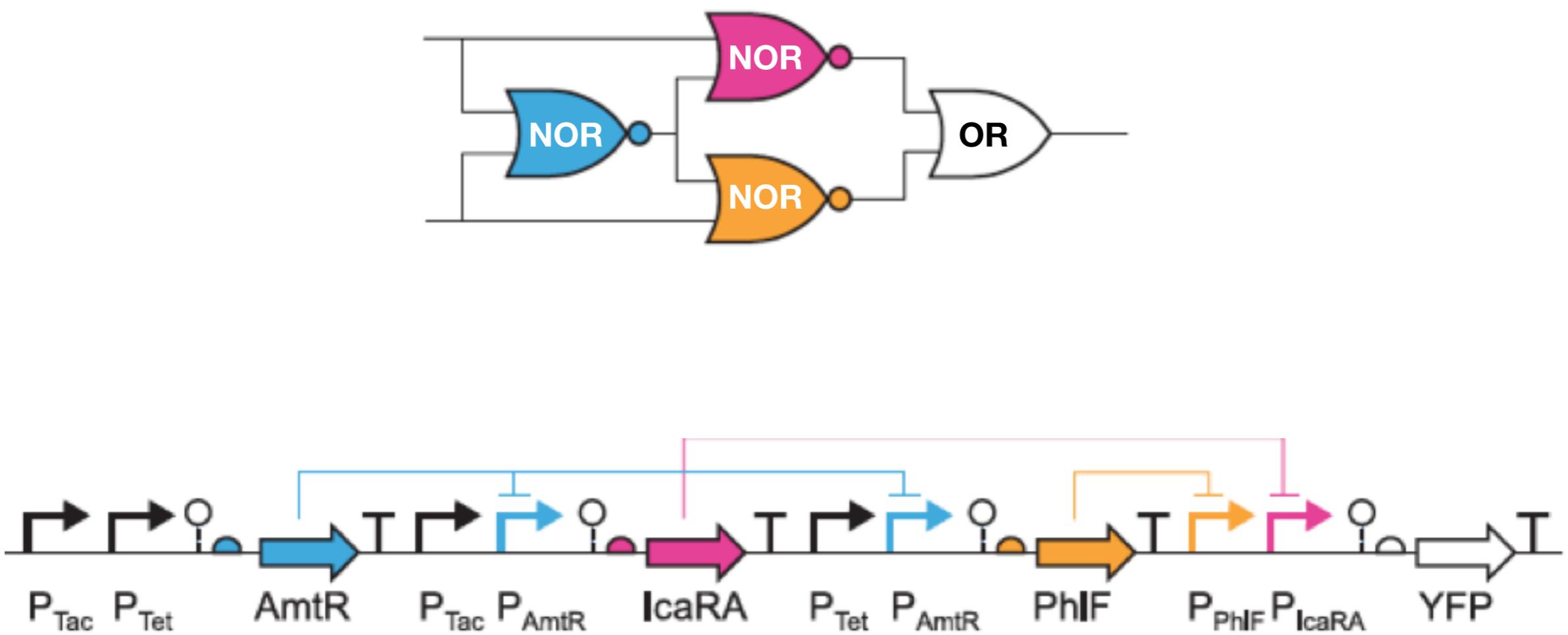
green sulfur
bacteria
photosynthesizer

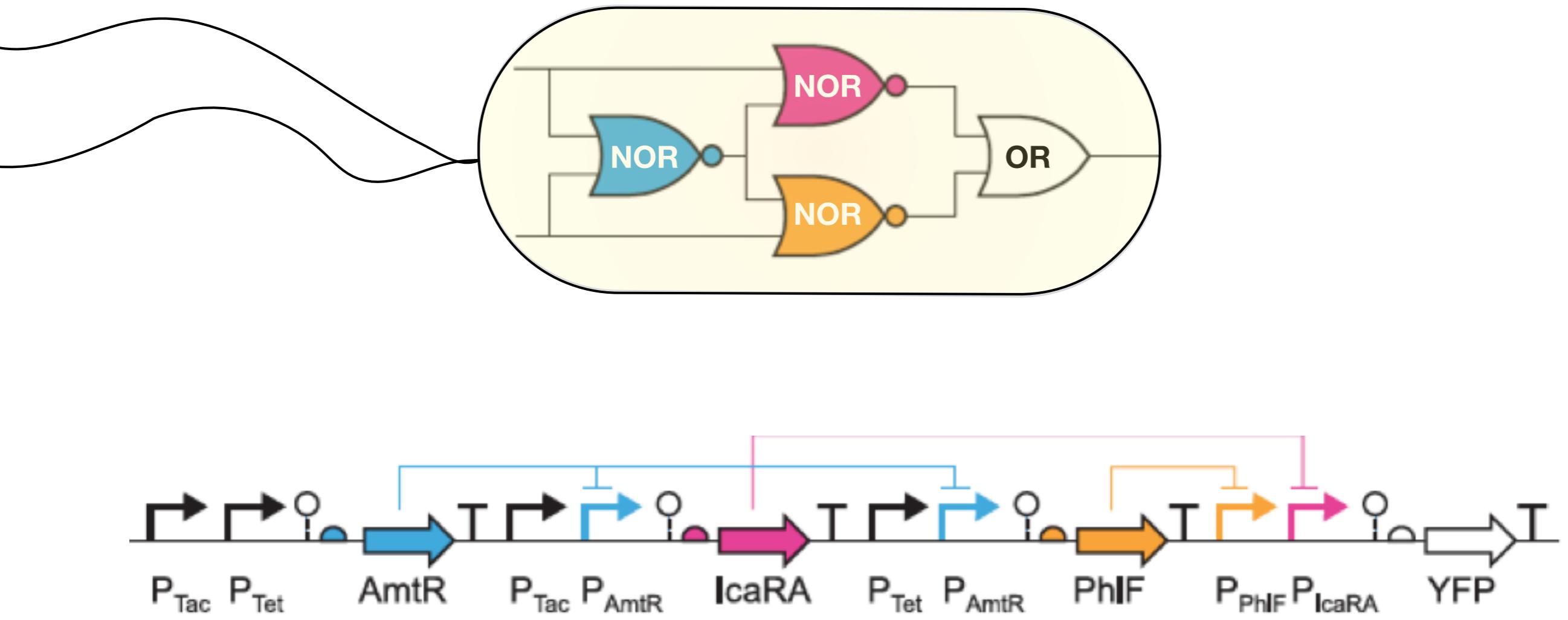
betaproteobacterium
motile

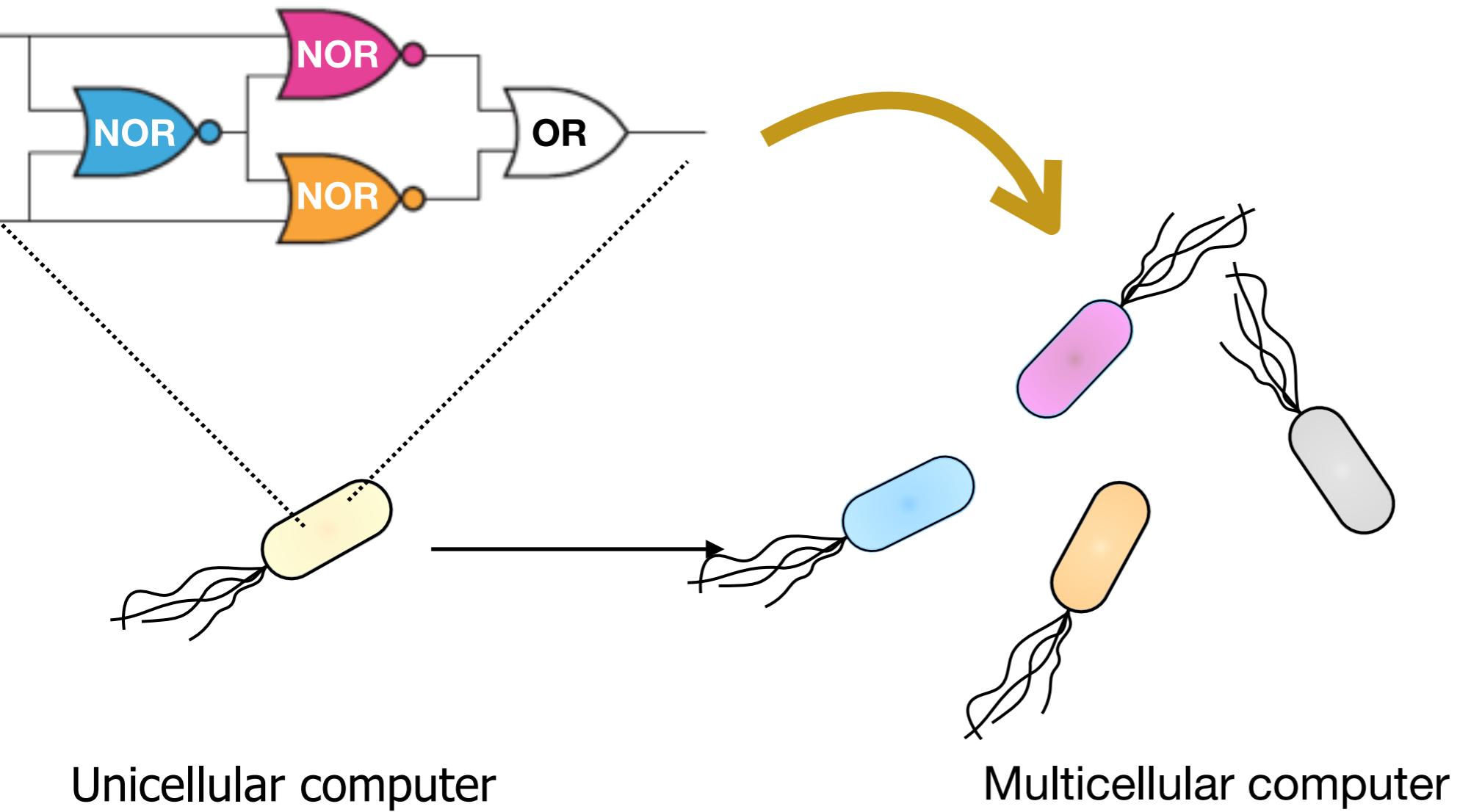


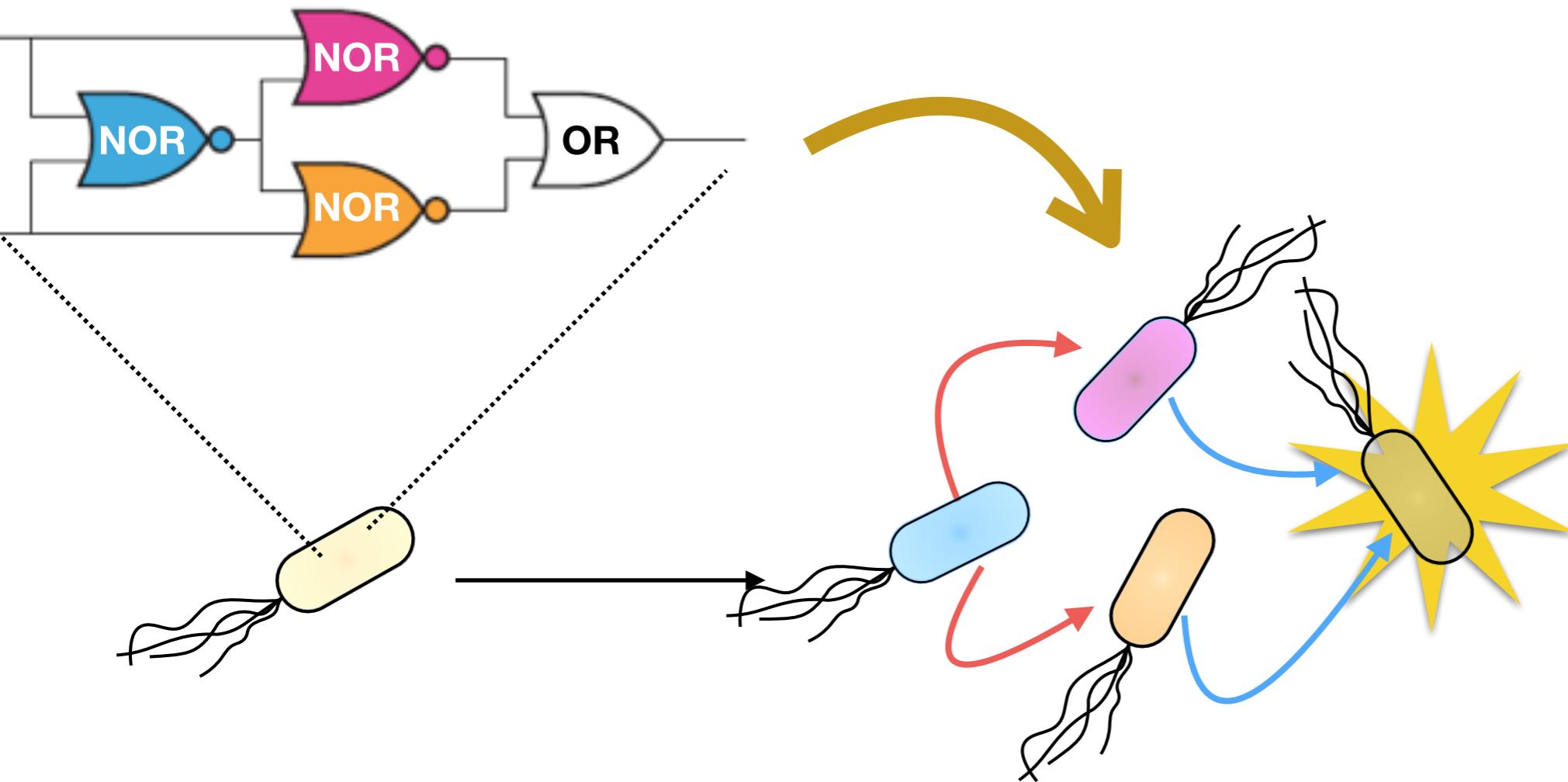
*Chlorochromatium
aggregatum*

motile
photosynthetic
community



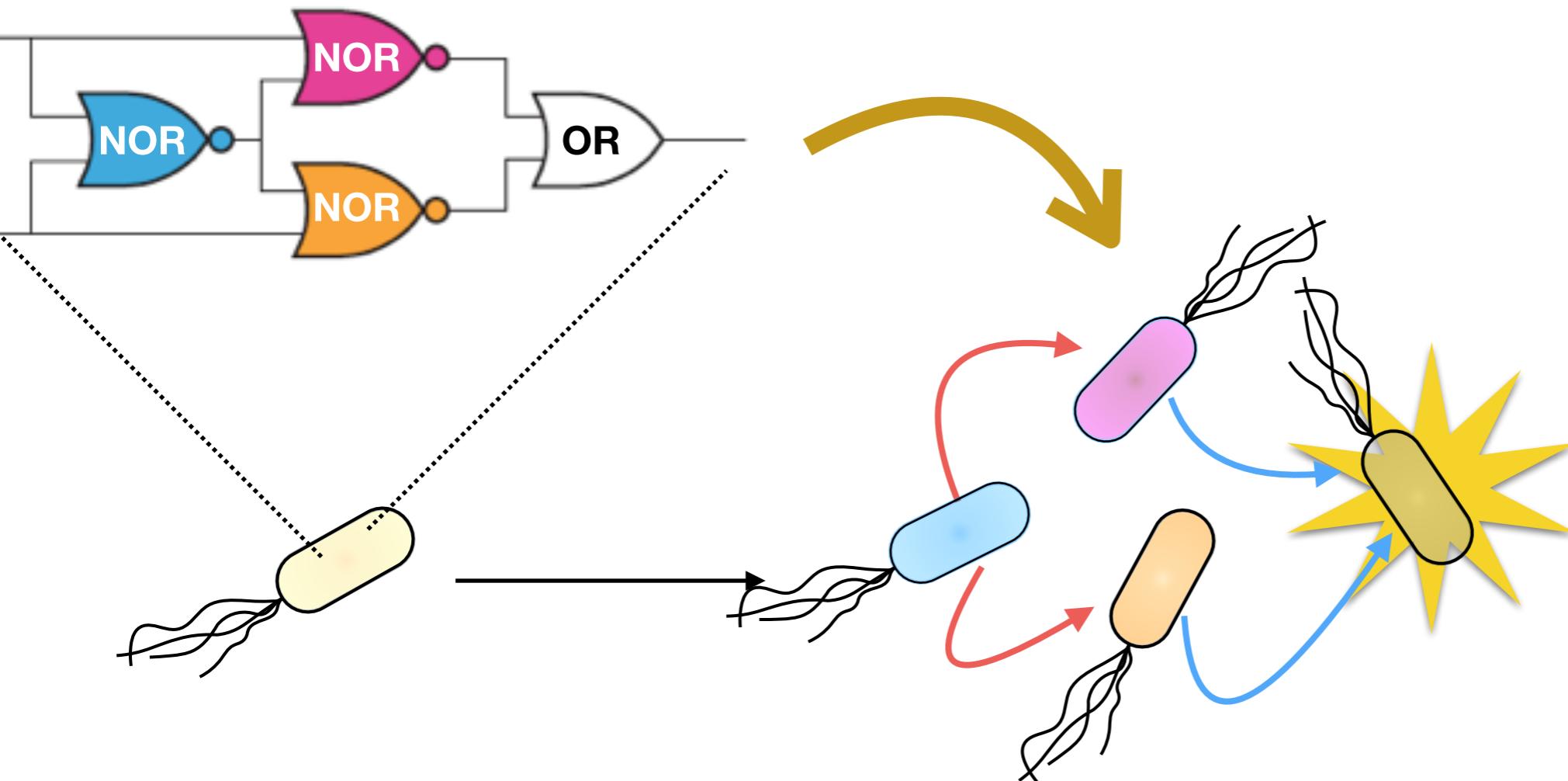






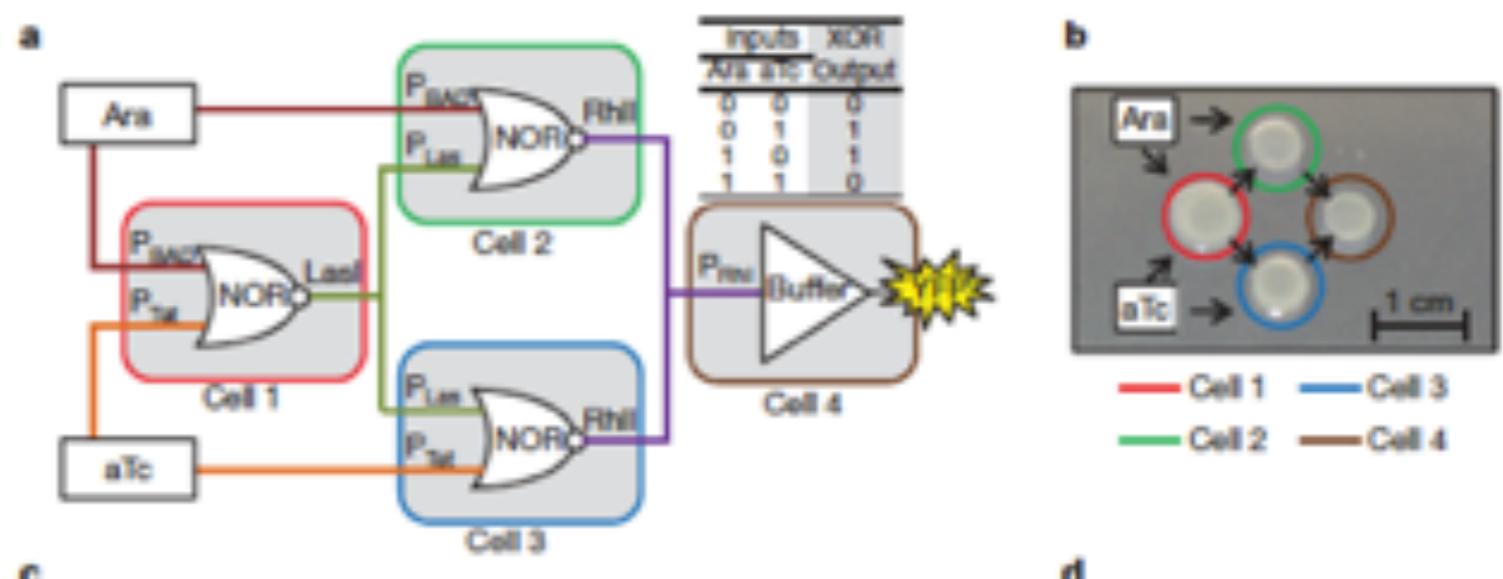
Unicellular computer

Multicellular computer



Unicellular computer

Multicellular computer

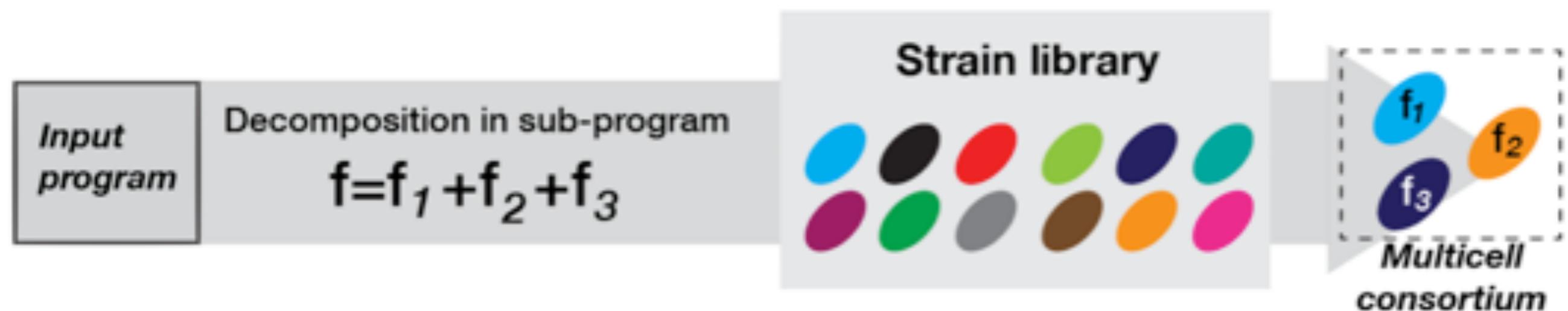


see also: Macia and Solé, 2010, 2016

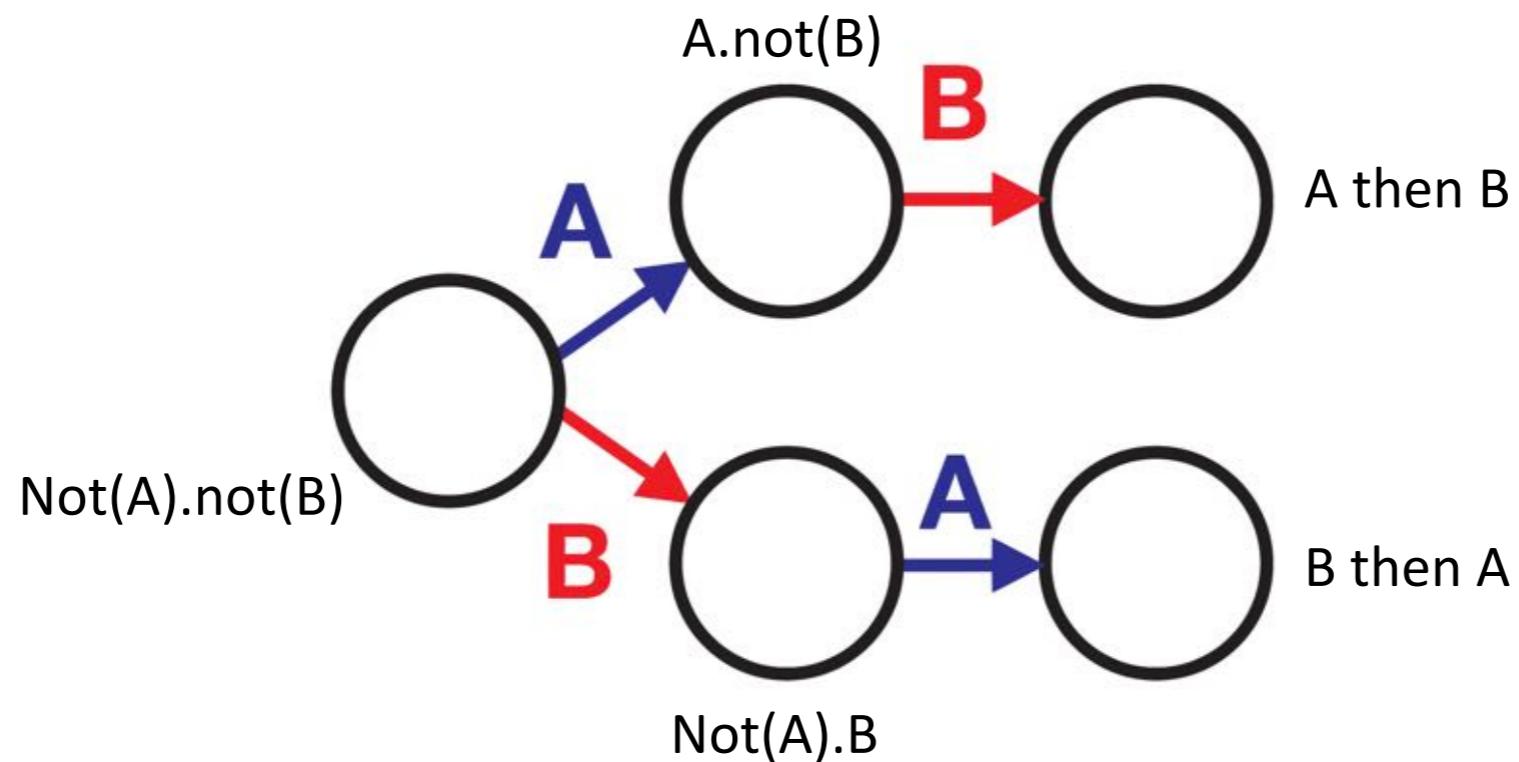
Tamsir et al, 2010

Multicellular recombinase logic.

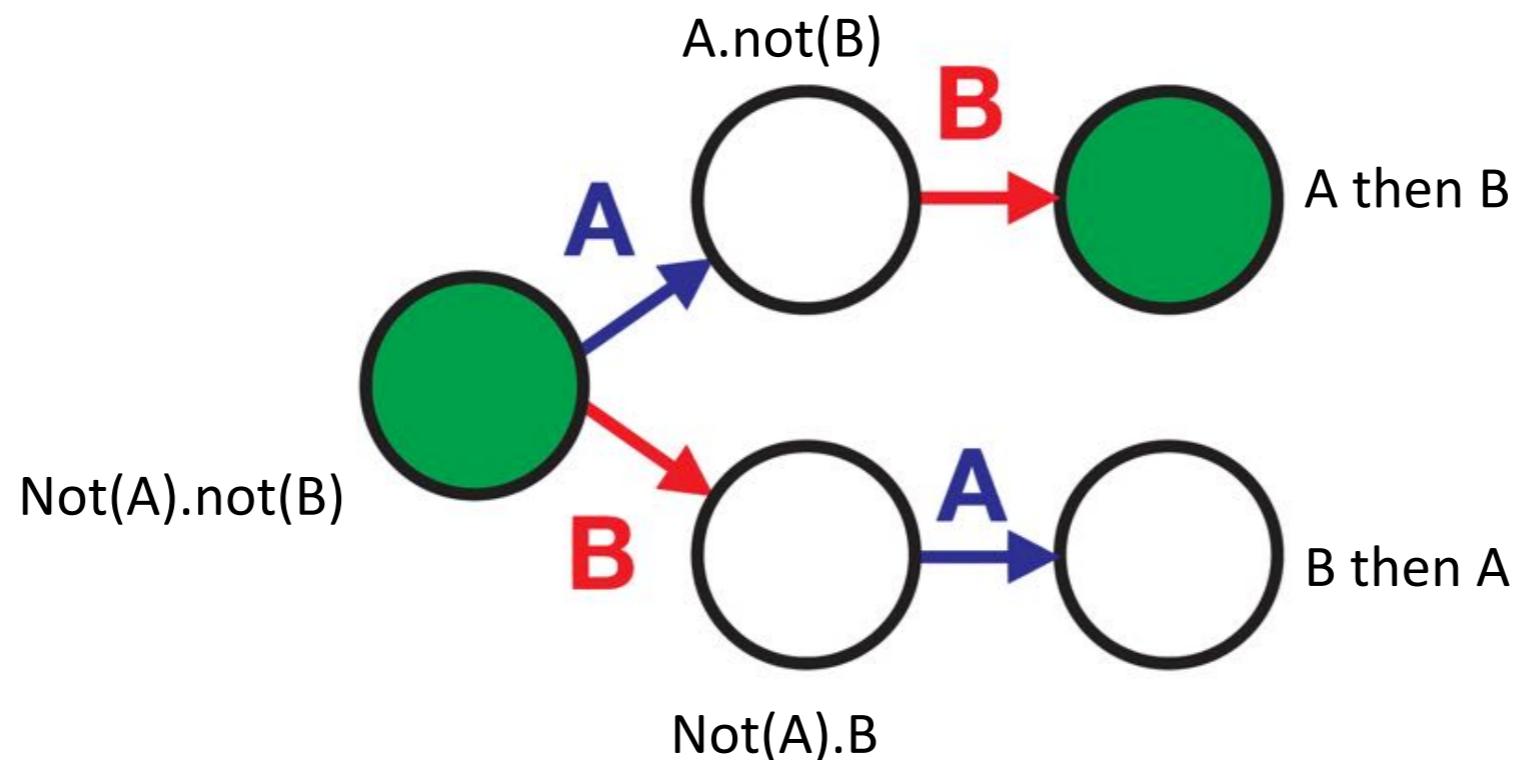
A reduced library of highly characterized logic devices that can be combined to achieve all Boolean logic functions.



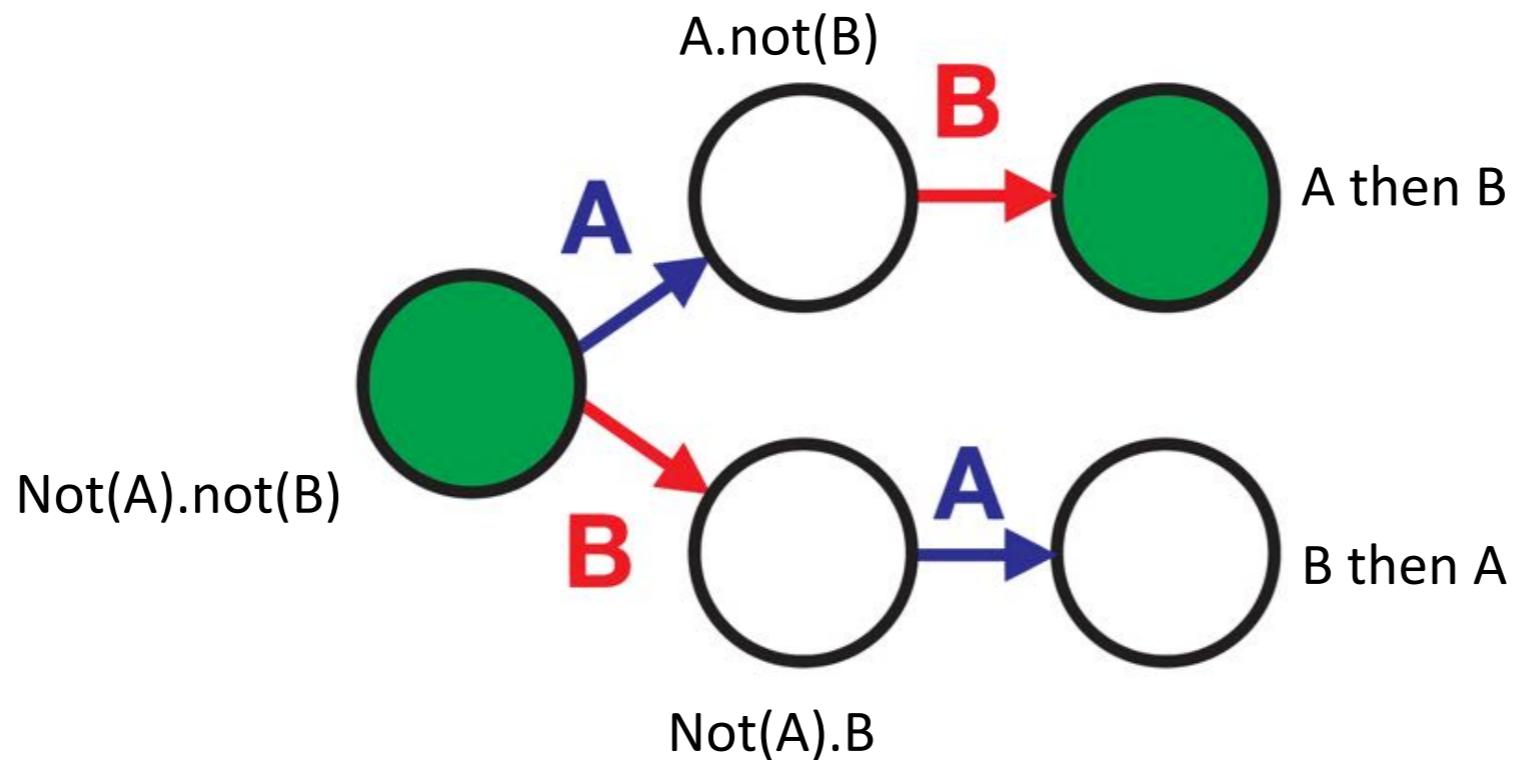
History-dependent logic.



History-dependent logic.

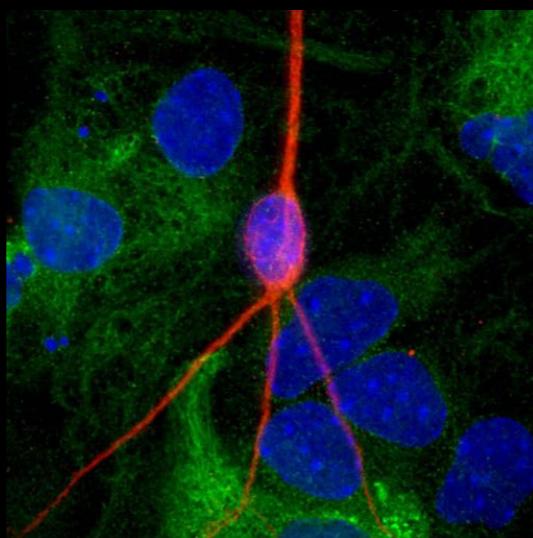


History-dependent logic.

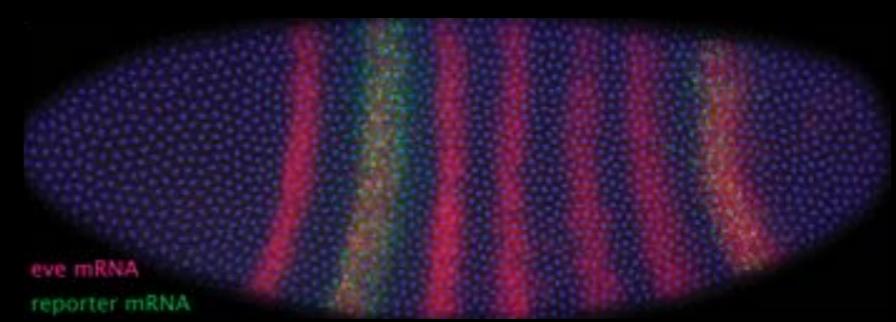


Differentiation
Development

Neuron derived from neural stem cells

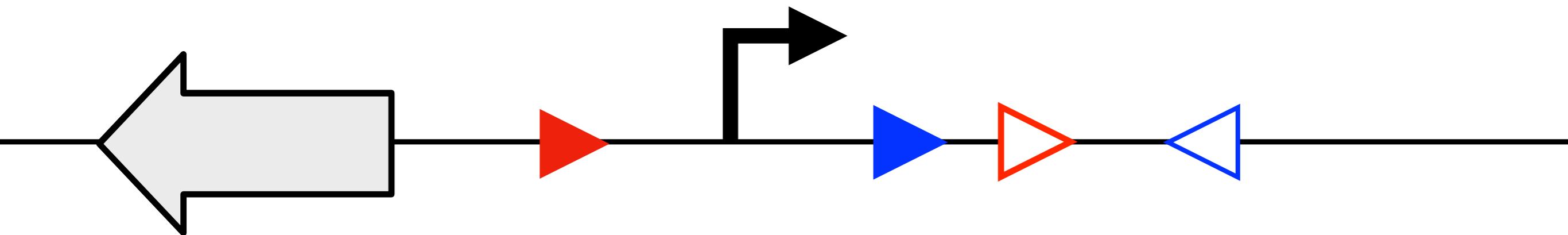


Paul Knoepfler, University of California, Davis

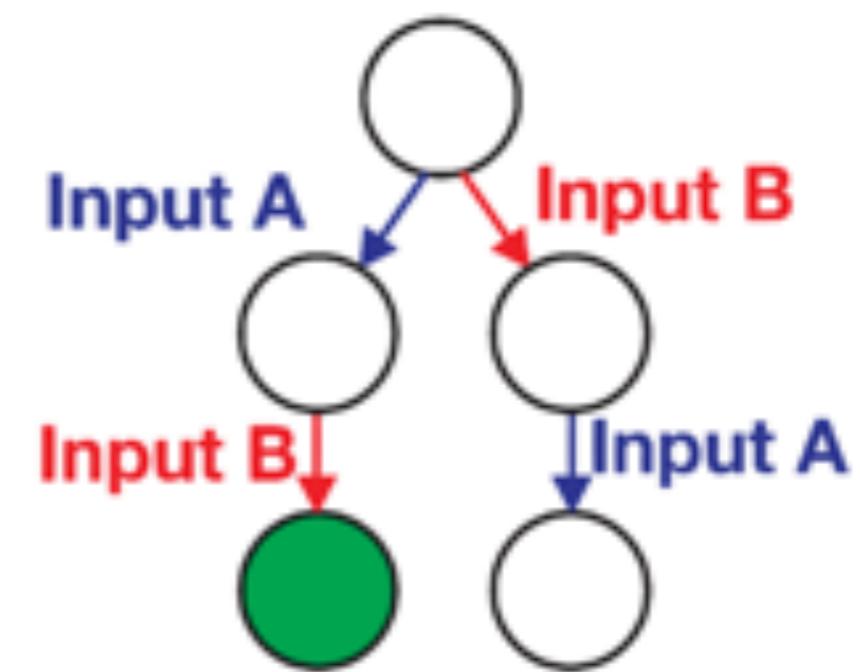


Drosophila embryo with tagged mRNA
Bothma et al., PNAS 2014

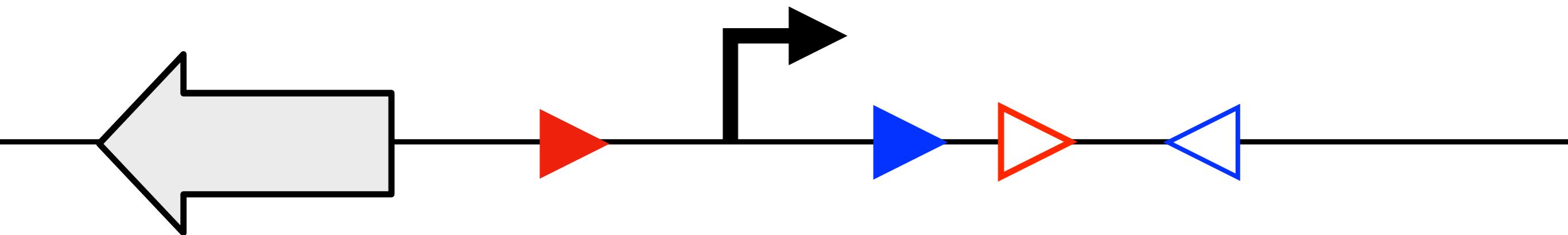
History-dependent logic using recombinases.



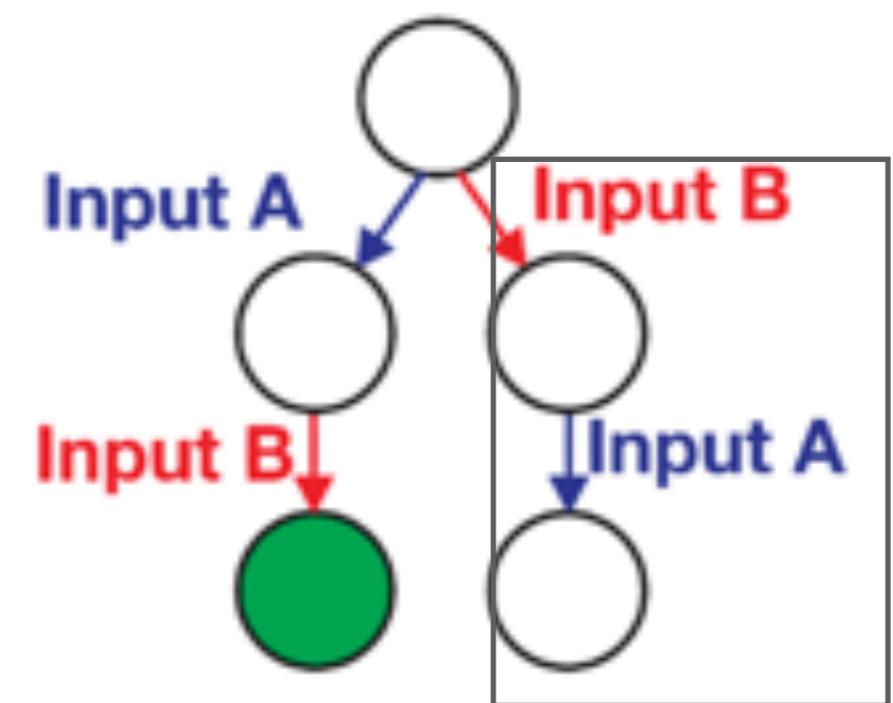
$f = A \text{ then } B$



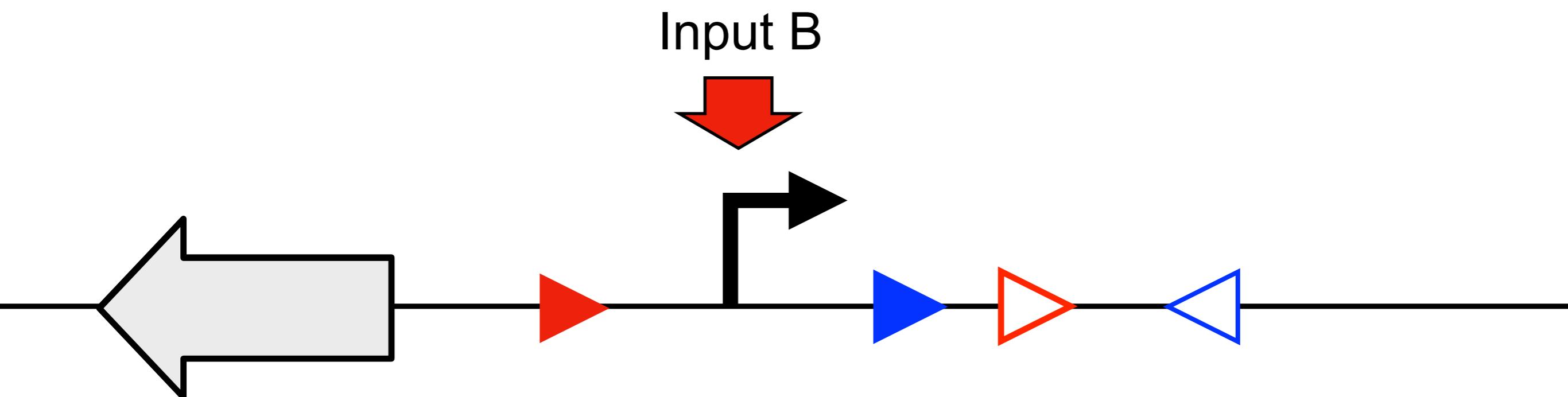
History-dependent logic using recombinases.



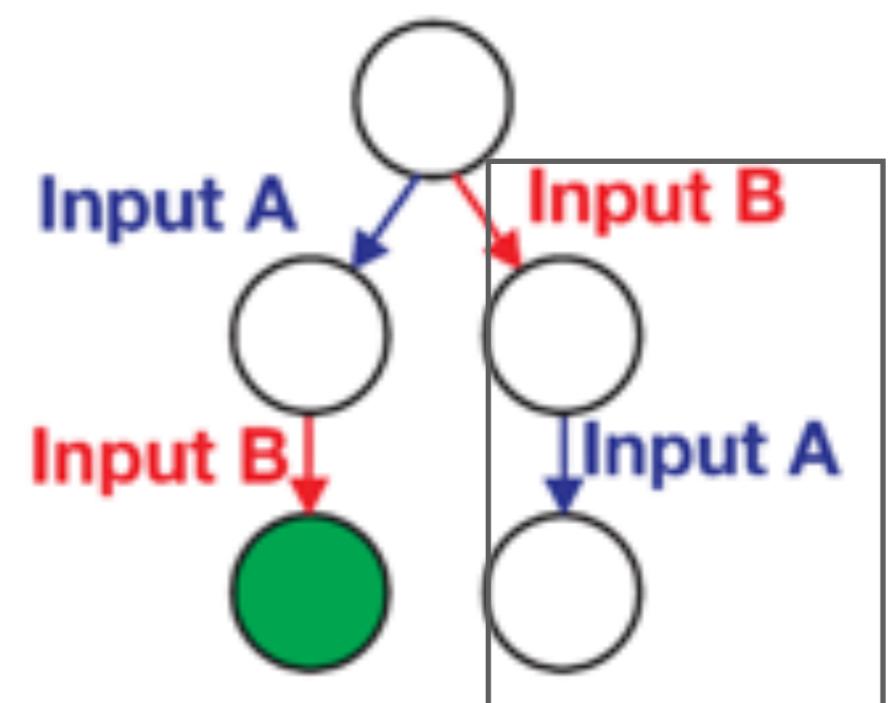
$f = A \text{ then } B$



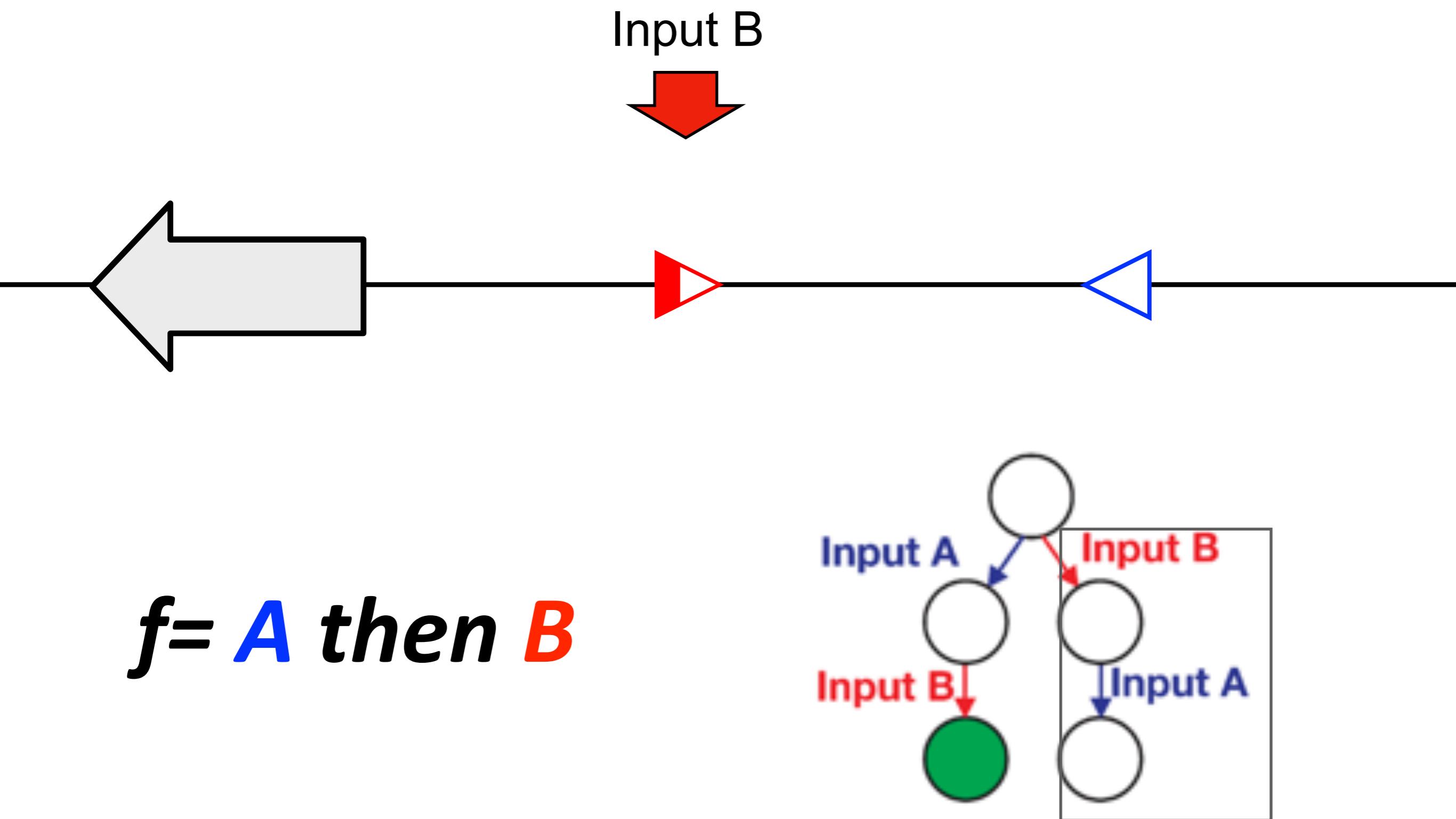
History-dependent logic using recombinases.



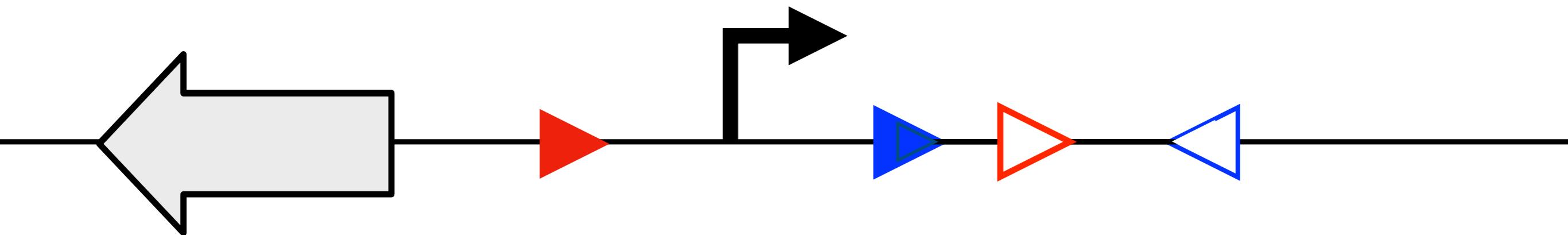
$f = A \text{ then } B$



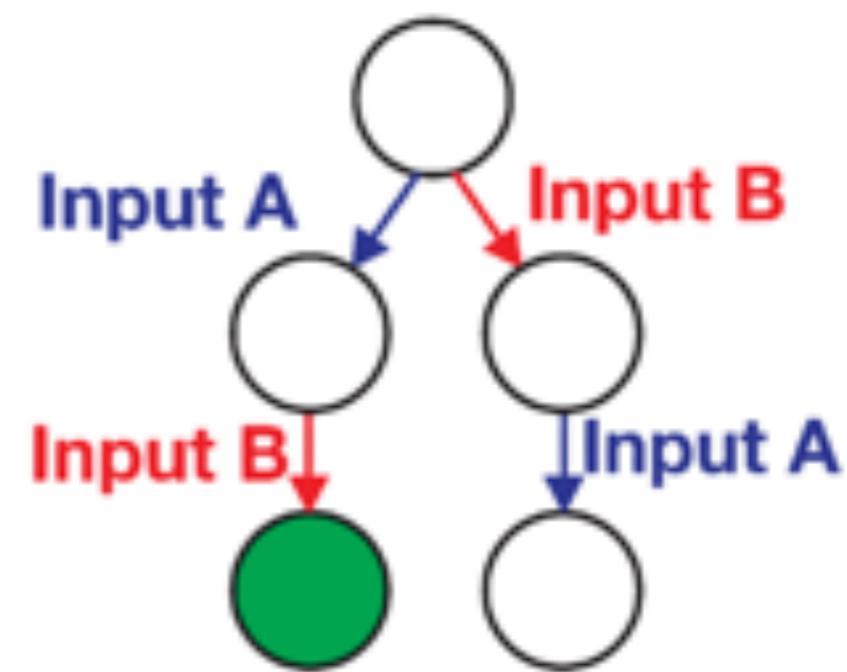
History-dependent logic using recombinases.



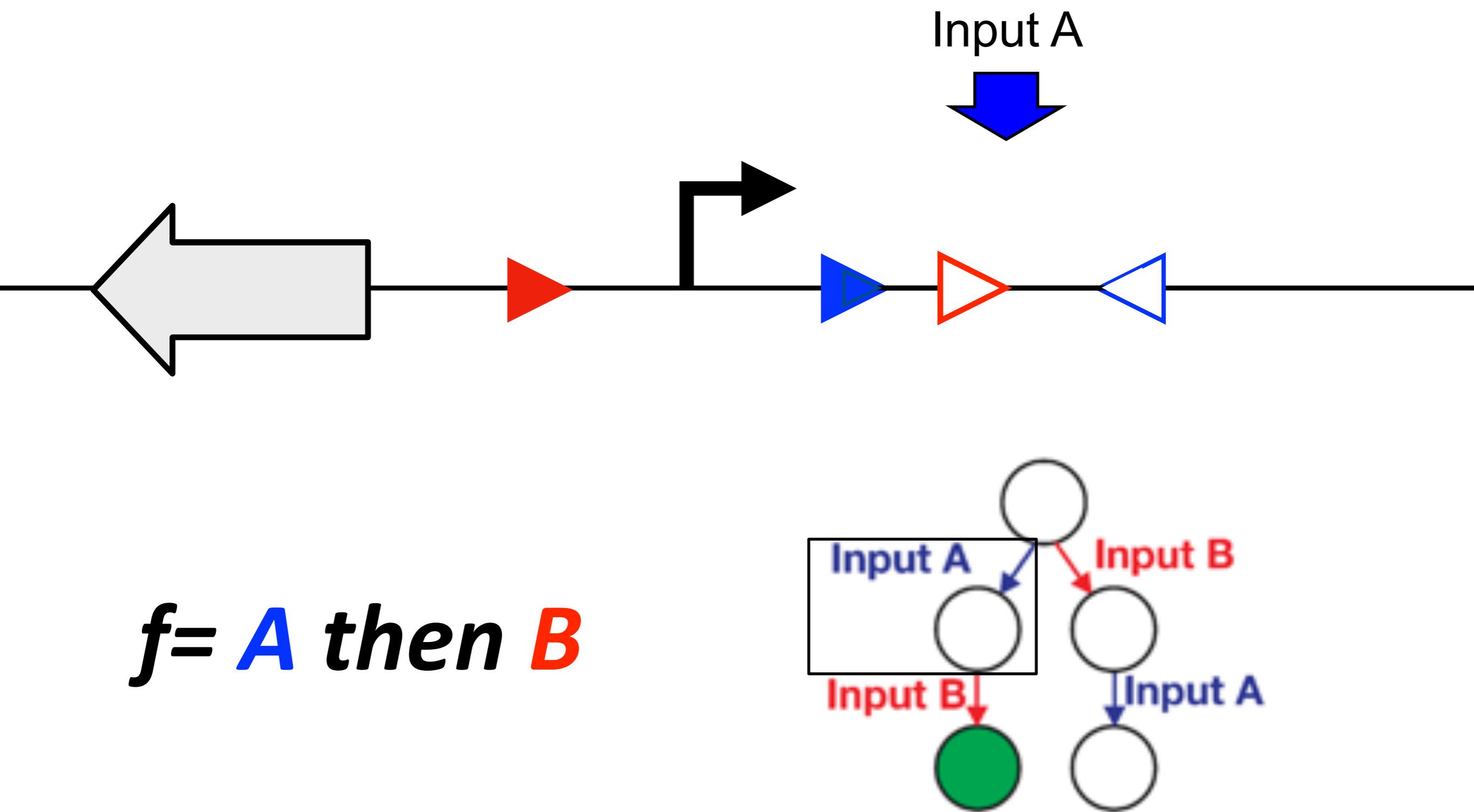
History-dependent logic using recombinases.



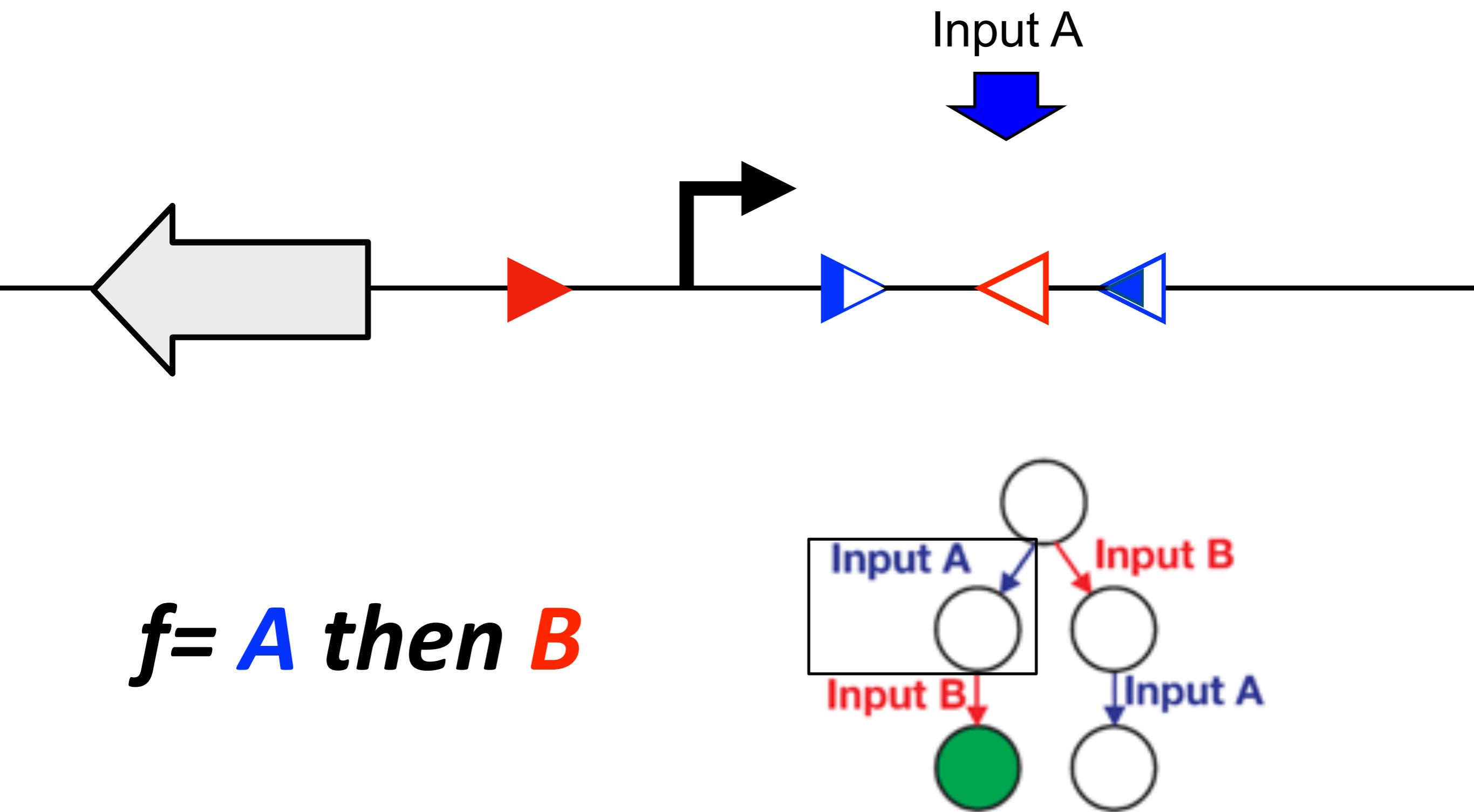
$f = A \text{ then } B$



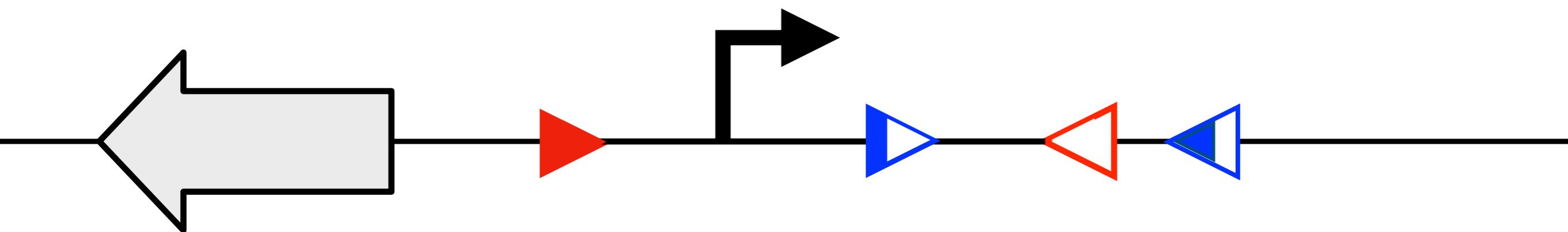
History-dependent logic using recombinases.



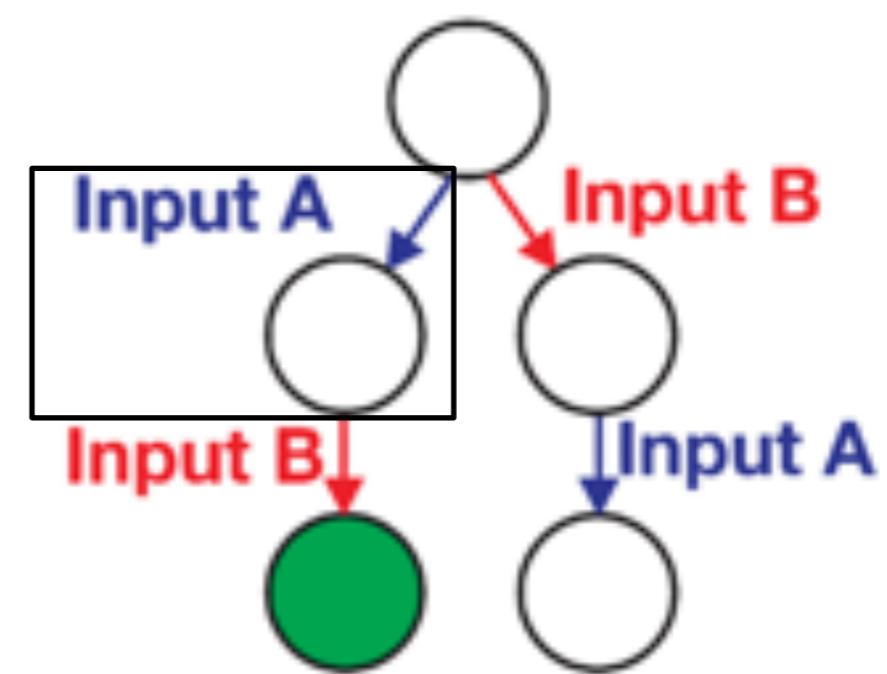
History-dependent logic using recombinases.



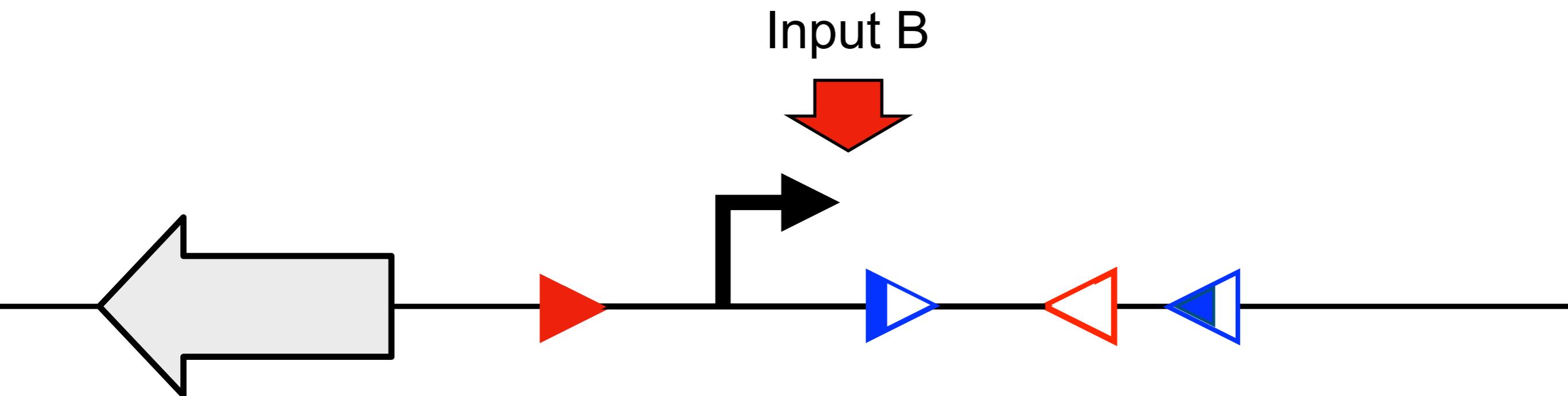
History-dependent logic using recombinases.



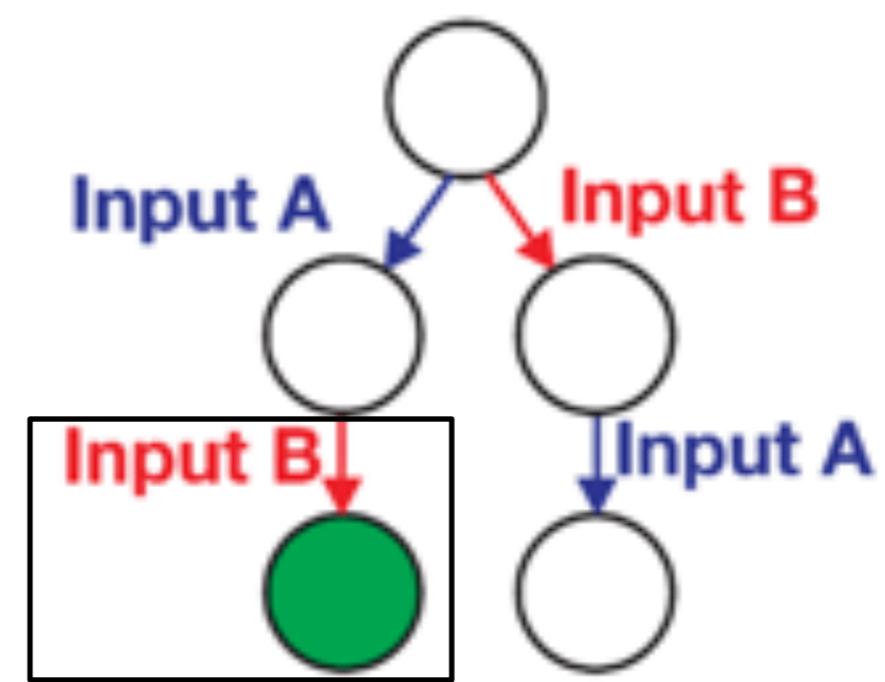
$f = A \text{ then } B$



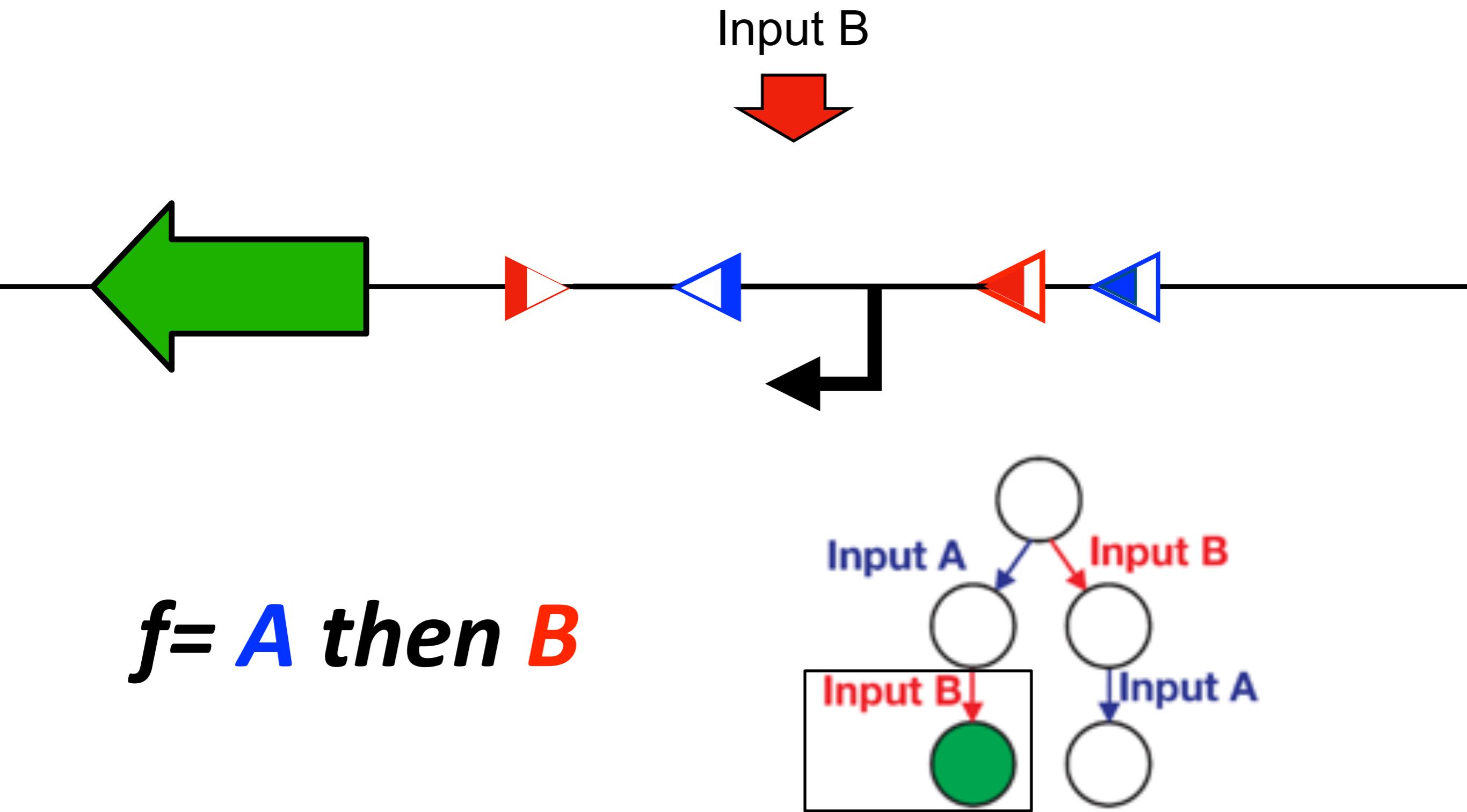
History-dependent logic using recombinases.



$f = A \text{ then } B$

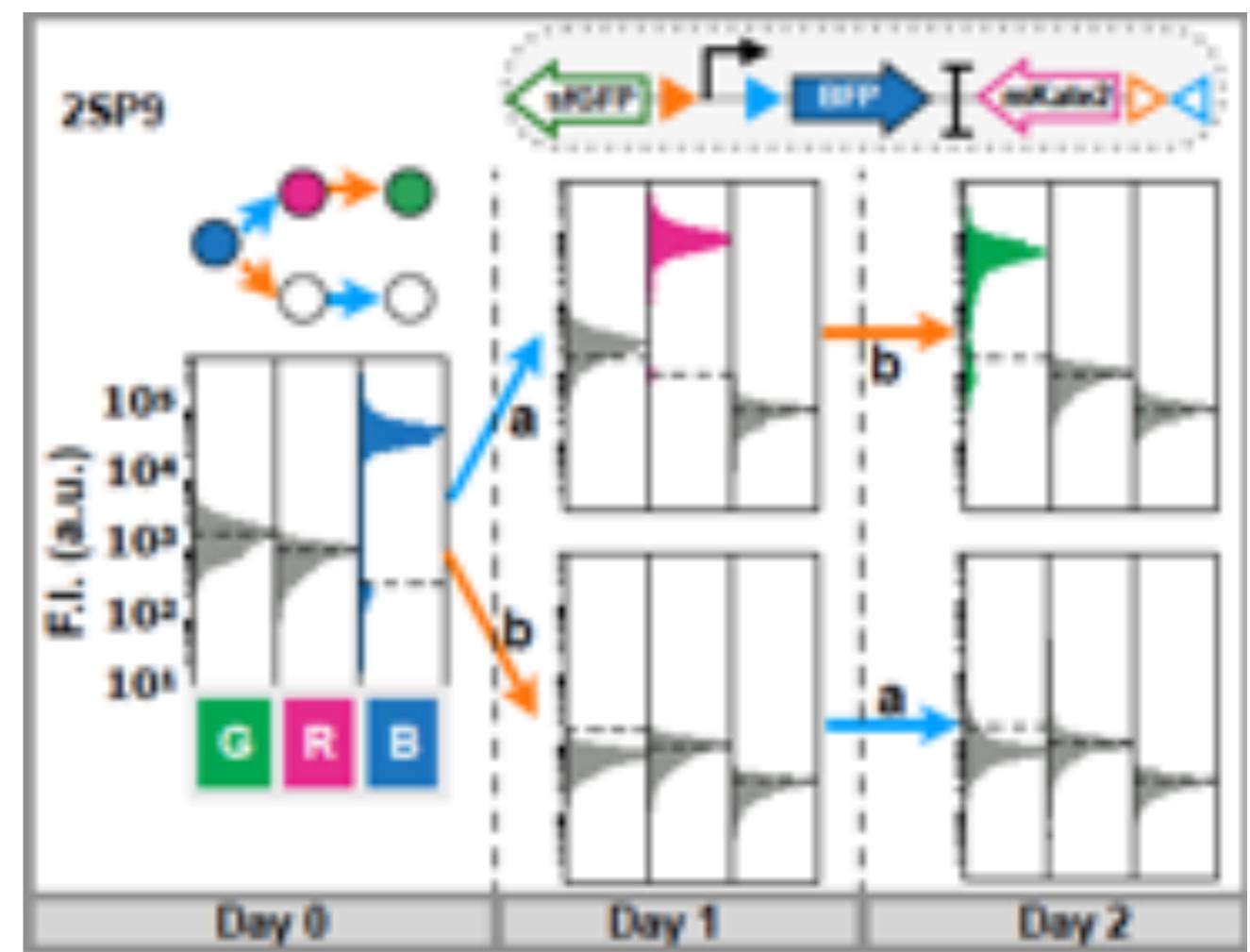
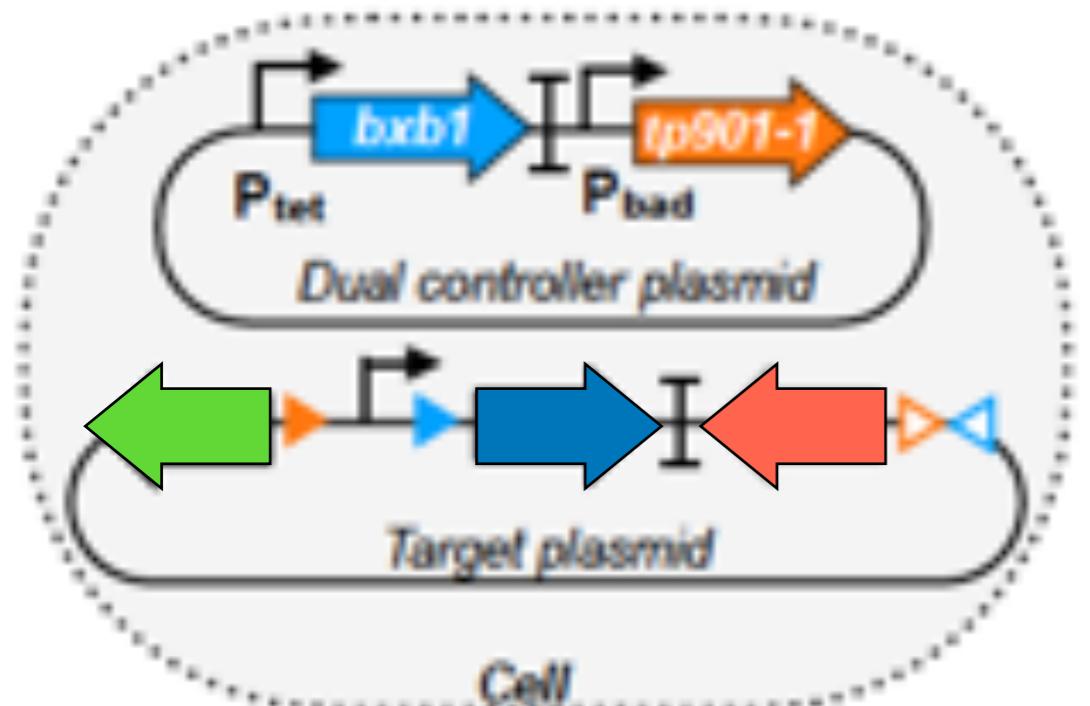


History-dependent logic using recombinases.



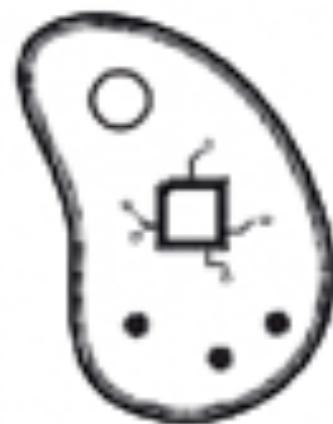
A 2-input scaffold for history-dependent gene expression programs

Co-transformation of target with integrase plasmid.
Sequential induction.

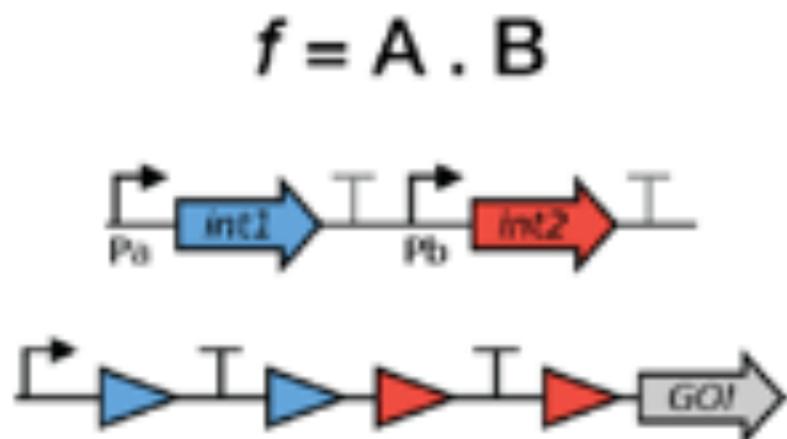


CALIN

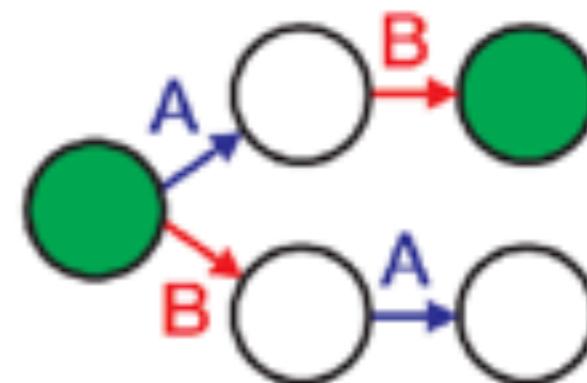
Composable Asynchronous Logic
using Integrase Networks



Asynchronous Boolean logic



History-dependent logic



synbio.cbs.cnrs.fr/calin

