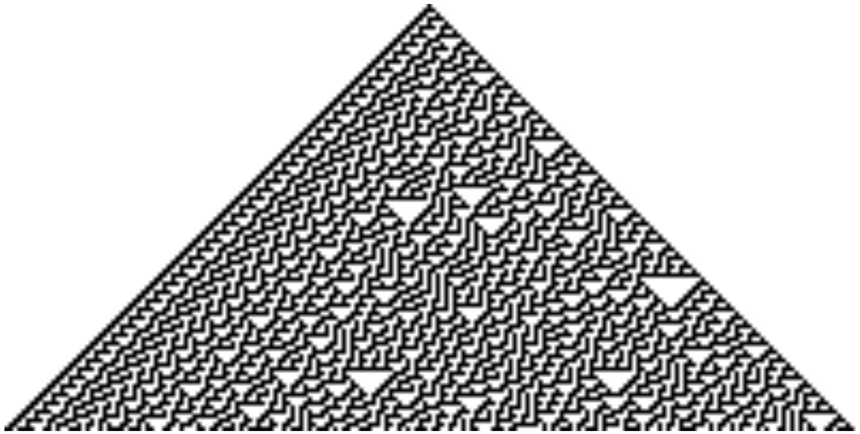
Gene Regulatory Models I

Dr. Michael Lones
Room EM.G31
M.Lones@hw.ac.uk

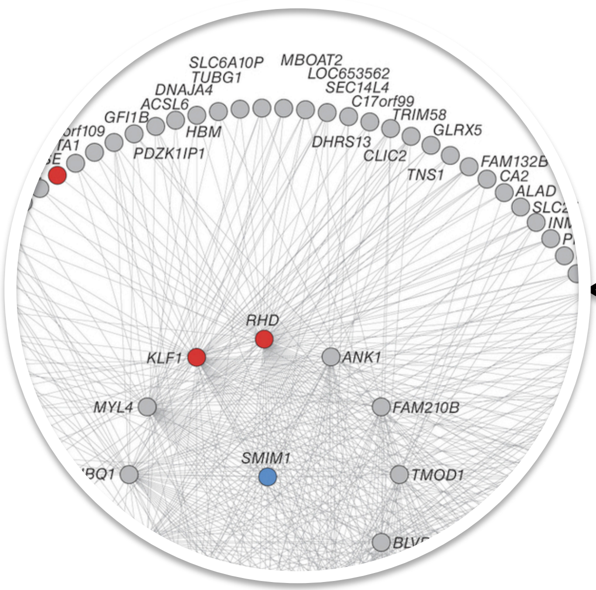
Previous Lecture

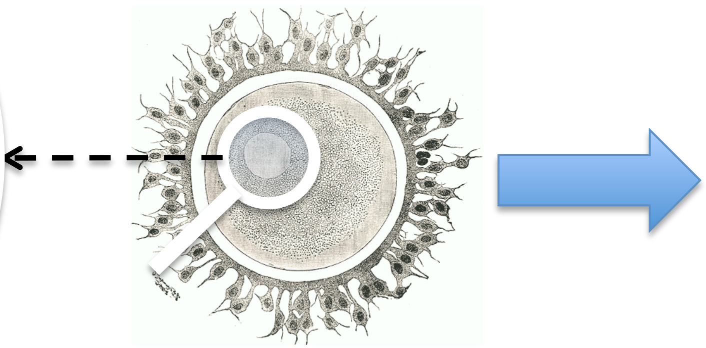
- Cellular automata
  - Distributed, bottom-up, emergent behaviour
  - Used to model natural systems
  - Complexity from simple systems
  - Can be used to compute

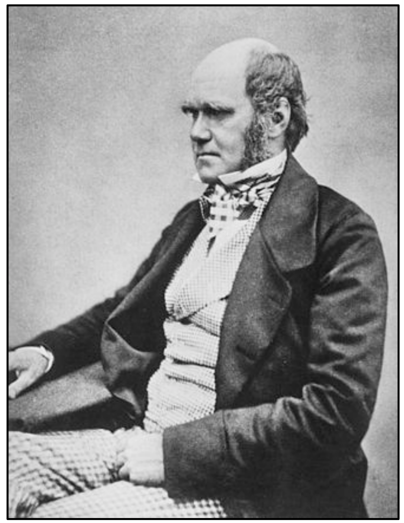


Today's Lecture

- How biological cells actually "compute"
  - Gene regulatory networks (GRNs)
  - Computational models of GRNs
  - Artificial development using GRNs

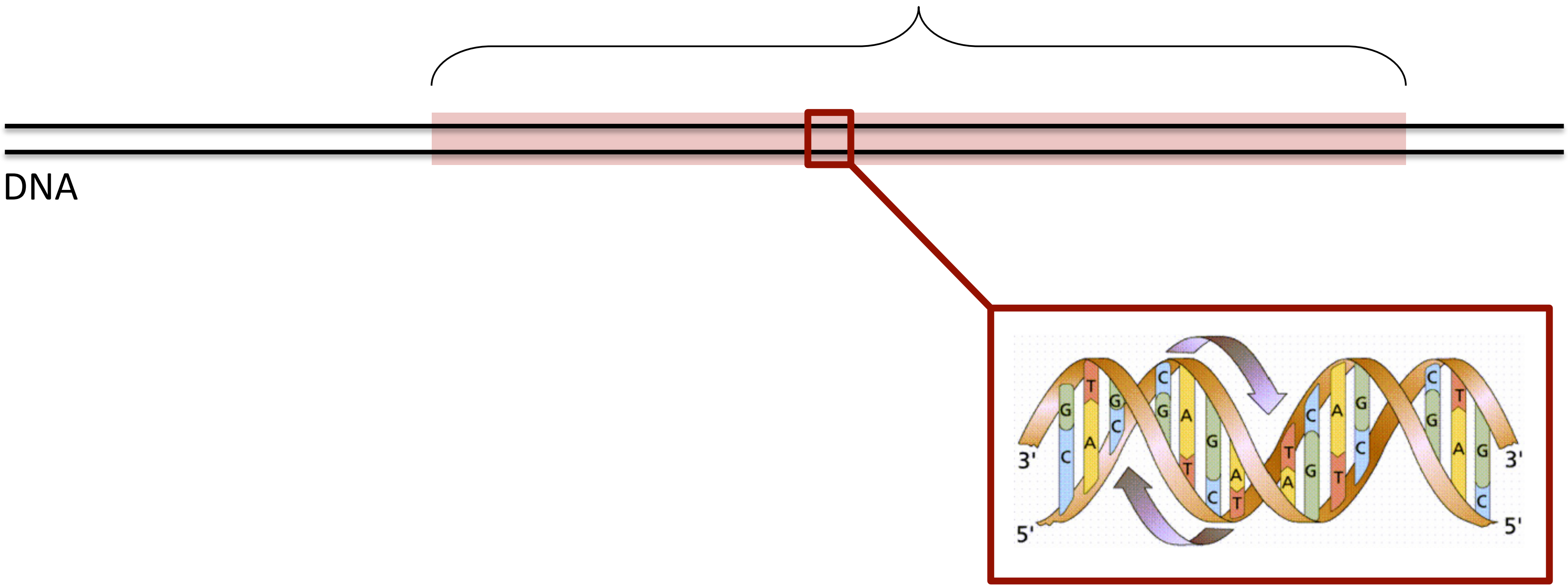






Gene Regulation in Biology

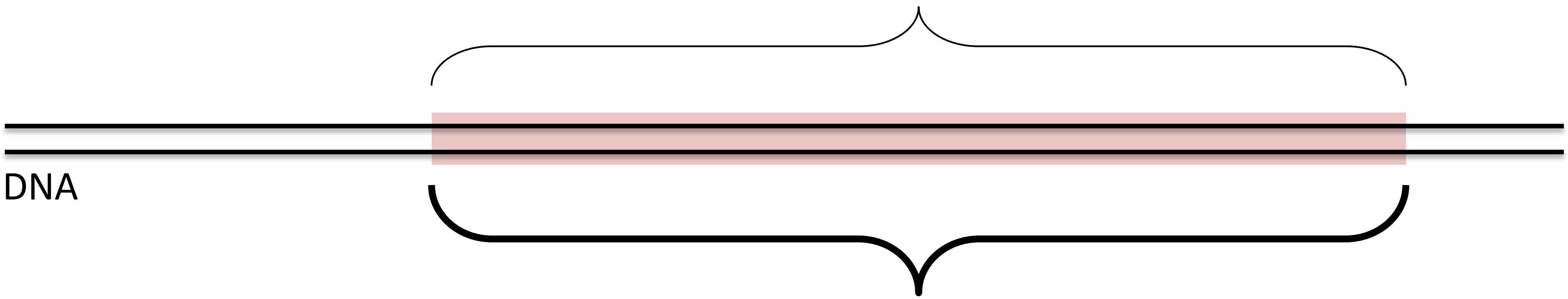
A **gene** is a contiguous region of DNA



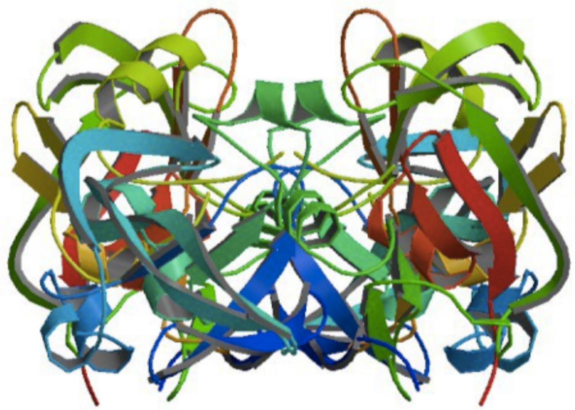
http://c2.plzcdn.com

Gene Regulation in Biology

A **gene** is a contiguous region of DNA

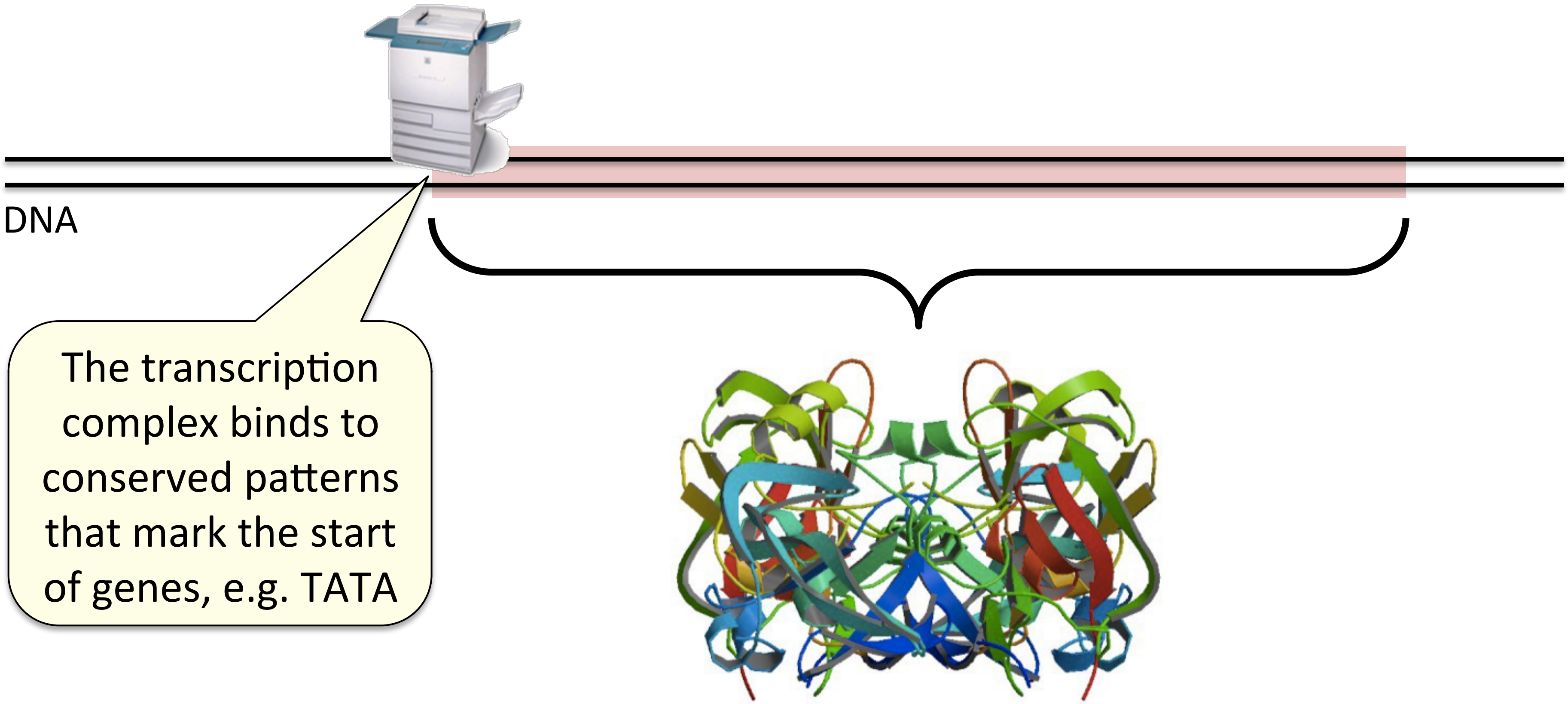


Each gene describes how to make a **protein**, which is a molecular machine



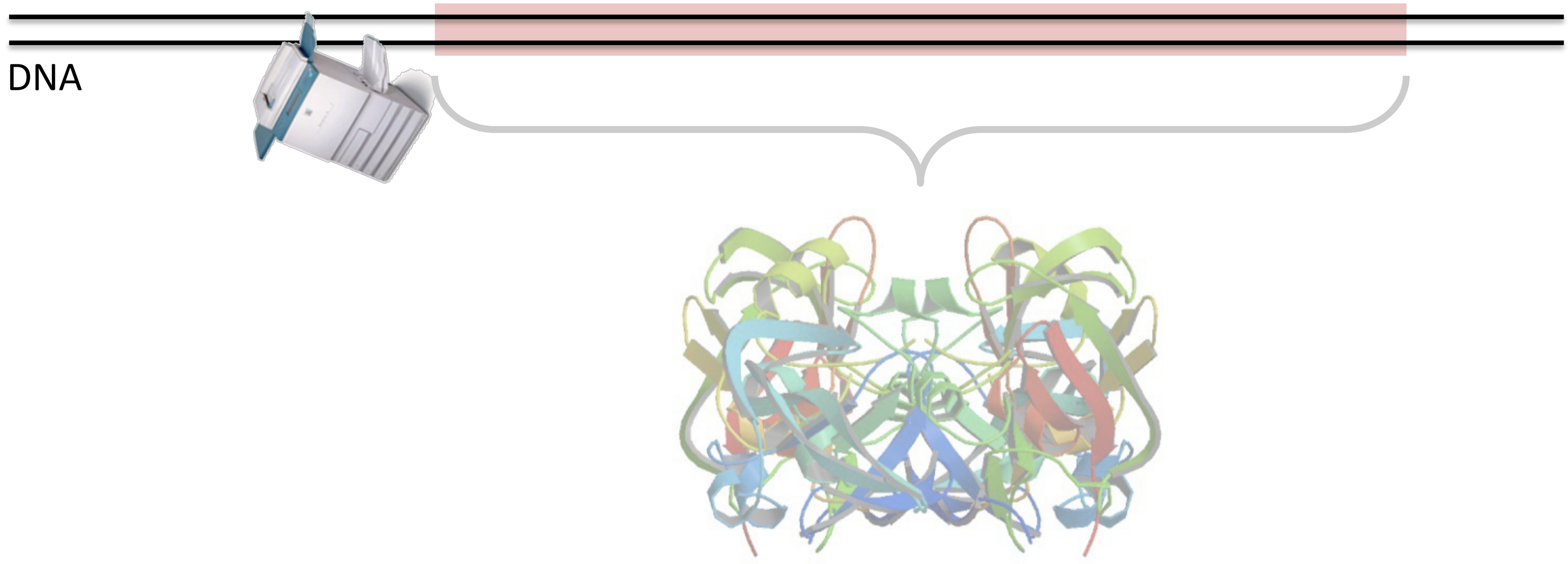
Gene Regulation in Biology

Genes are expressed when a **transcription complex** forms – this is a bit like a photocopier



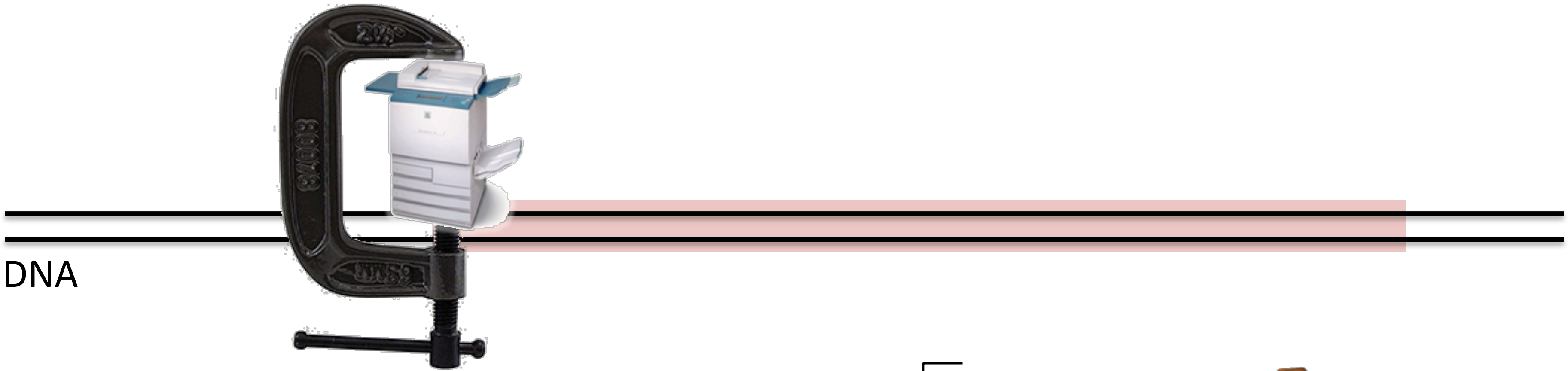
Gene Regulation in Biology

However, the transcription complex is unstable, and rarely copies genes by itself

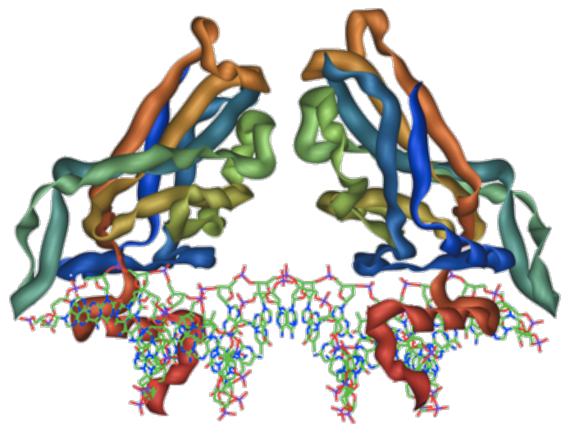


Gene Regulation in Biology

Instead, it must be helped out by
proteins called **transcription factors (TFs)**

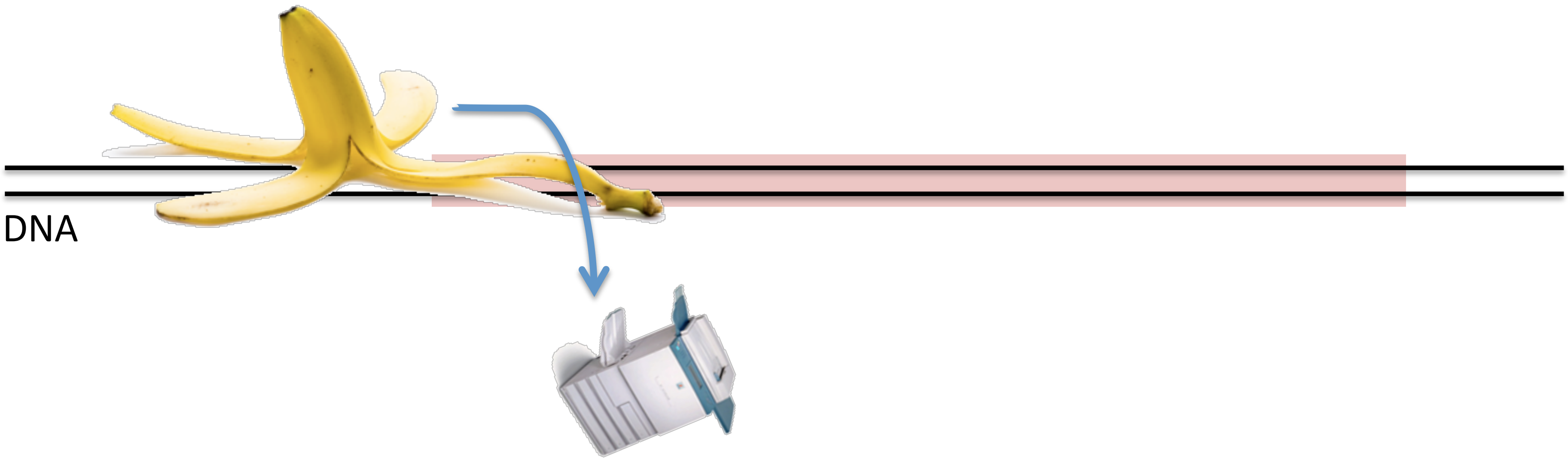


These bind to both the DNA and the transcription complex, holding everything in place – a bit like a clamp



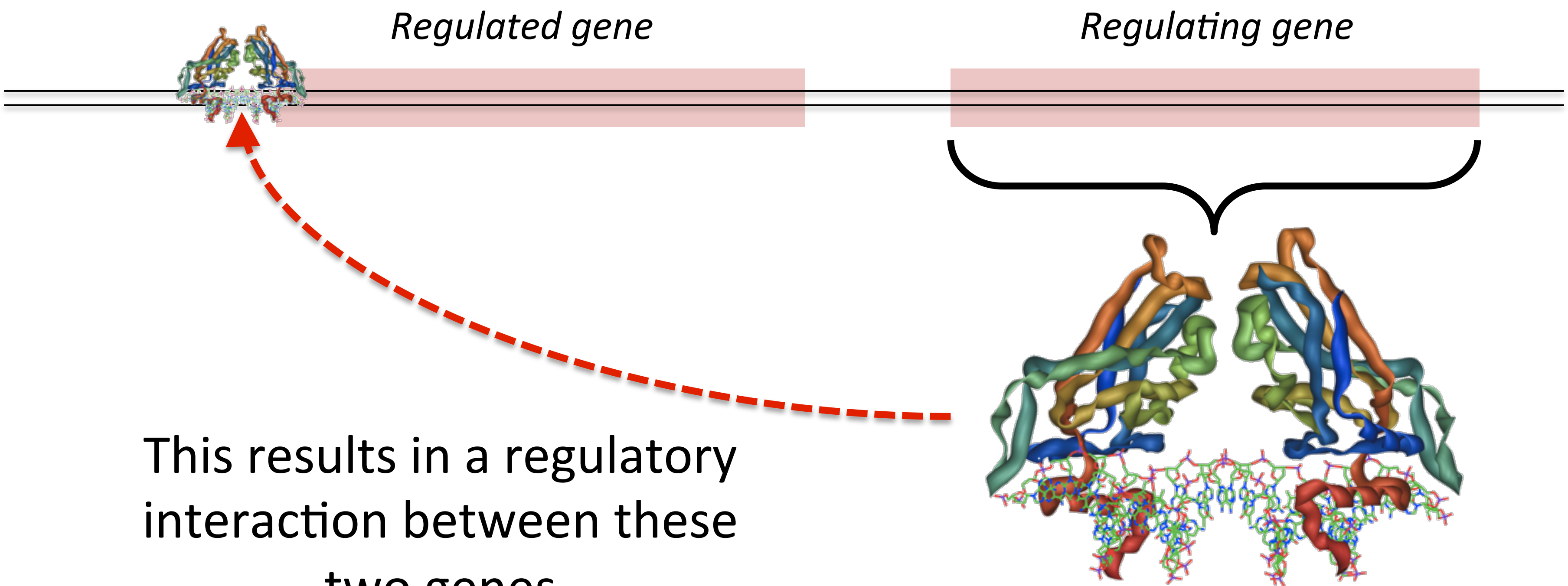
Gene Regulation in Biology

Some TFs are inhibitory and act to destabilise the transcription complex



Gene Regulation in Biology

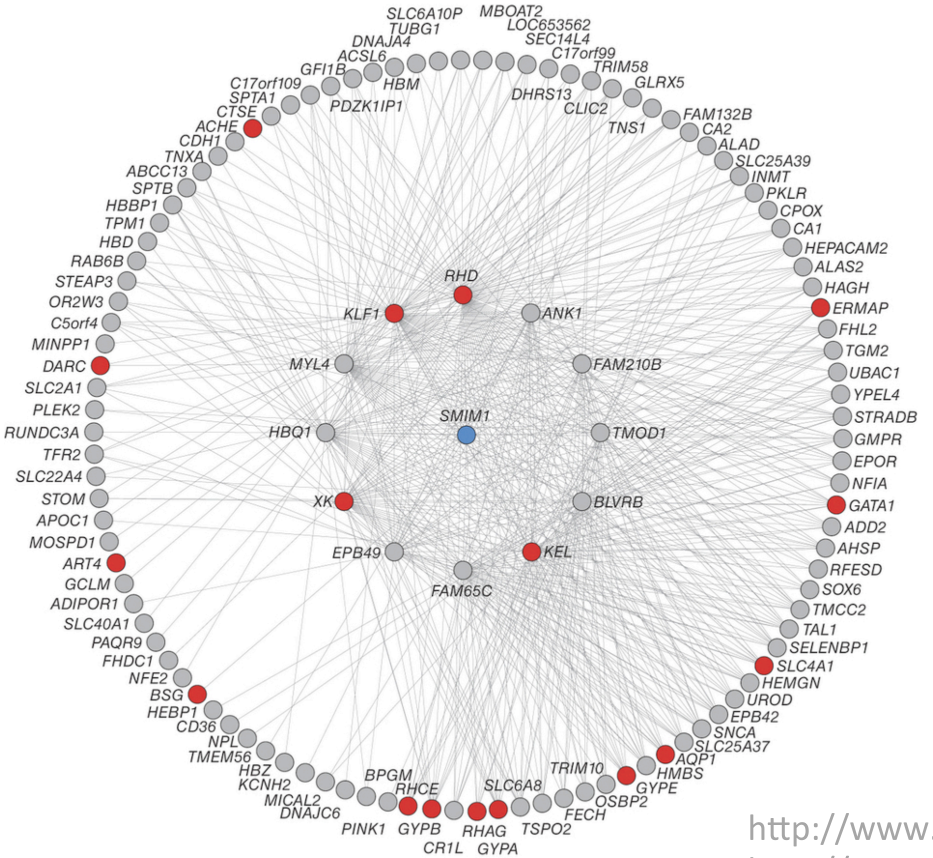
Since transcription factors are proteins, they must be produced by other genes …

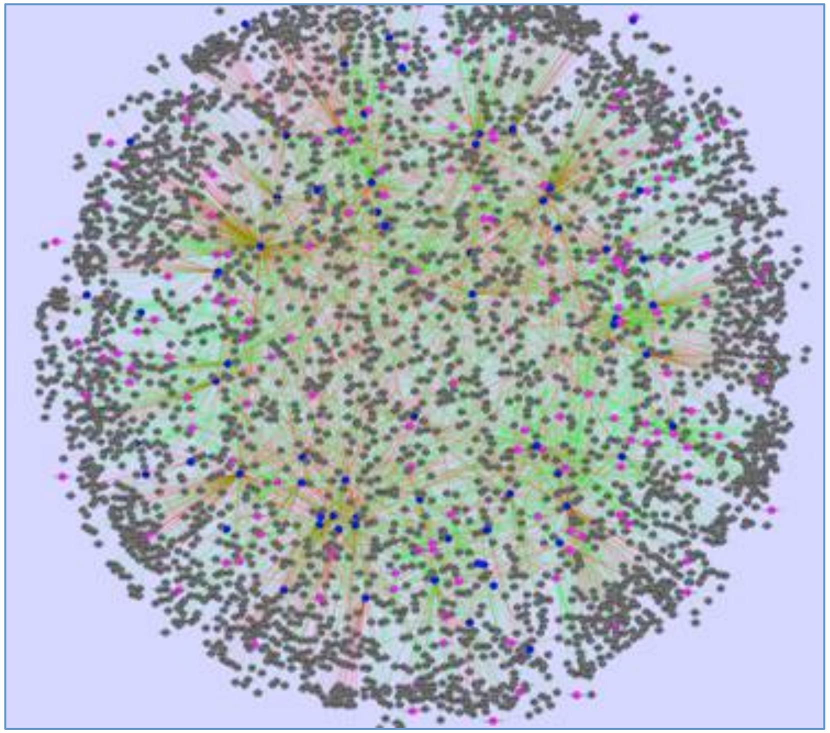


This results in a regulatory interaction between these two genes

Gene Regulation in Biology

- The pattern of interactions between all genes is known as the gene regulatory network (or GRN)

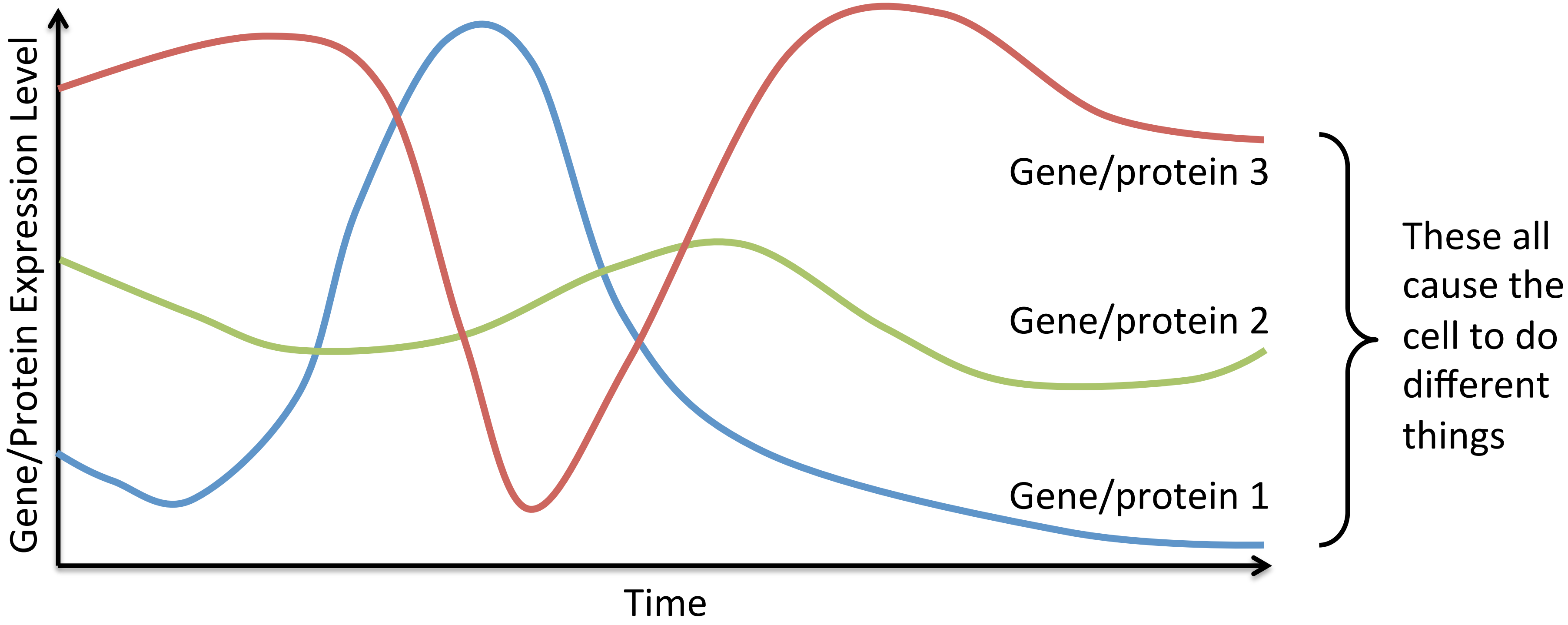




http://www.nature.com/ng/journal/v45/n5/fig_tab/ng.2600_F2.html
http://www.scbit.org/cgrnb/faq.htm

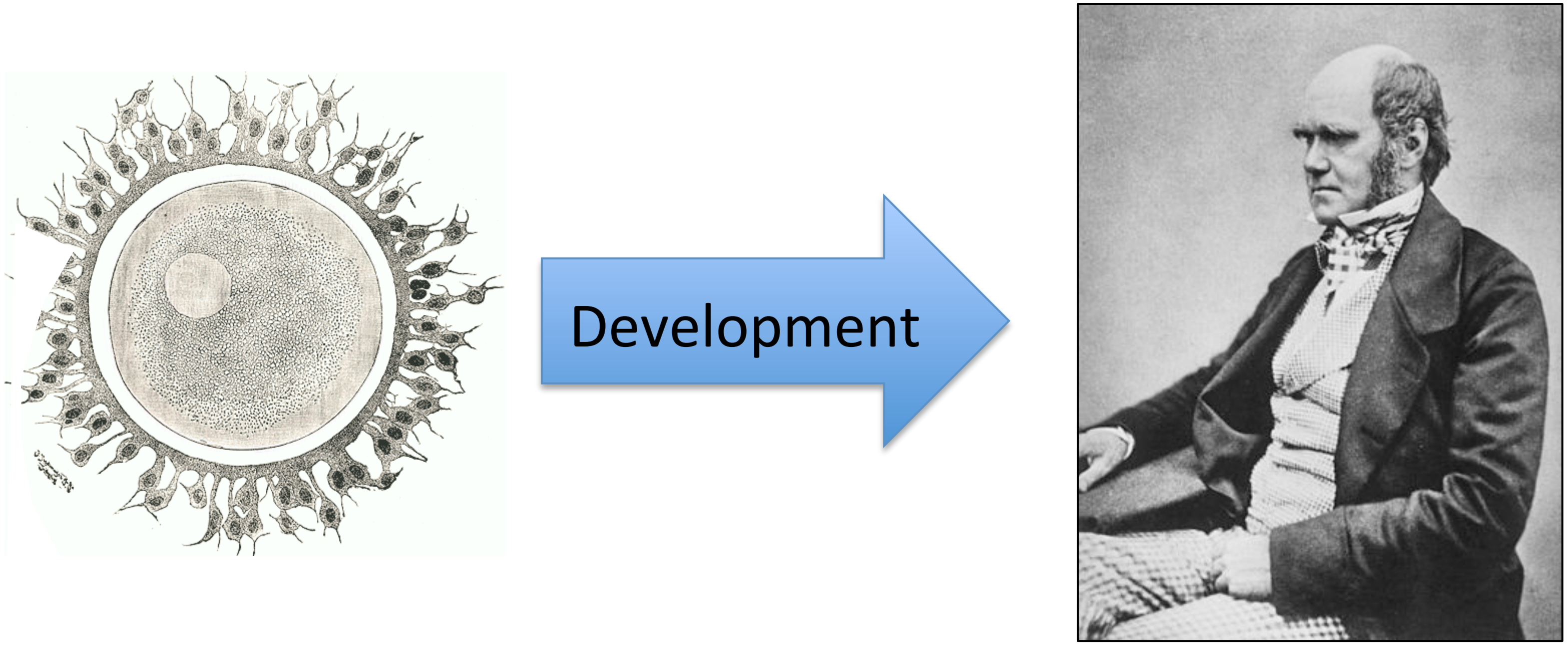
Gene Regulation in Biology

- The GRN controls the expression of proteins, and hence the behaviour of the cell, over time
  - It is, basically, the cell's computer



Gene Regulation in Biology

- The GRN also determines when cells divide and the kind of cells they will become
  - i.e. an organism's developmental process



Gene Regulation in Biology

- Why is gene regulation of computational interest?
  - GRNs underlie the complexity of biological systems, such as ourselves

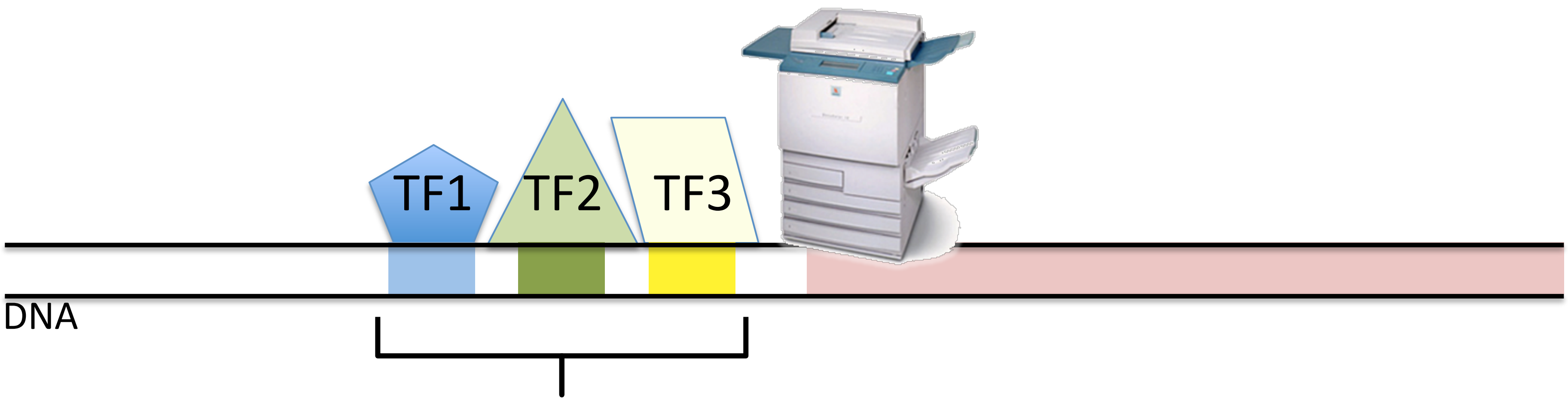
- Biological systems are **structurally complex**
  - This has led to an interest in how computational models of GRNs can be used to generate intricate structures

- Biological systems are **dynamically complex**
  - GRNs produce robust and intelligent responses, leading to interest in whether computational models can do the same, e.g. for controlling robots

Gene Regulation in Biology

- Also theoretical interest in **computability**
  - How do biological systems process information?
  - How does this differ from conventional computers?
  - Is it in some ways better? e.g. more compact and robust
- And from an EA perspective **evolvability**
  - GRNs are known to be evolvable, i.e. able to respond robustly to mutation and crossover
  - Especially in comparison to computer programs
  - Potentially an evolvable representation for GP

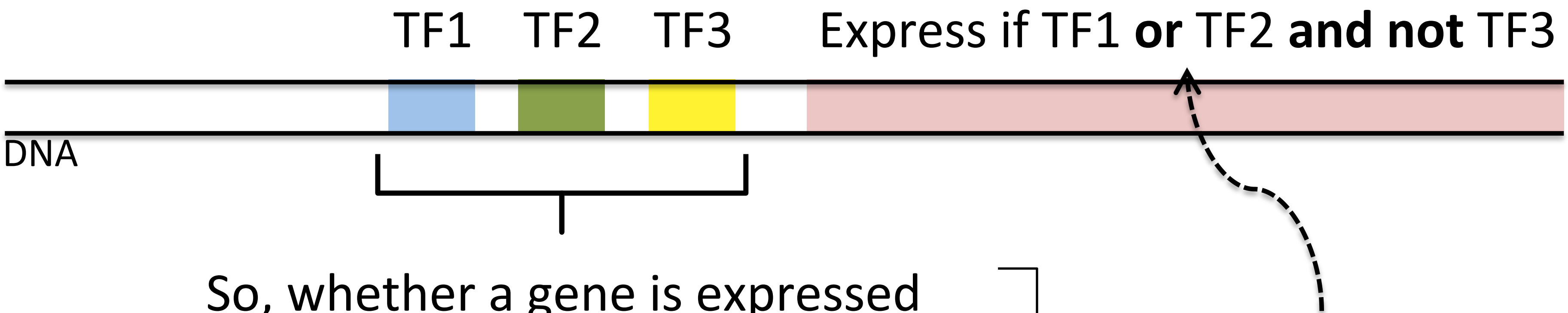
Gene Regulation in Biology



A number of different TFs are often involved in regulation, binding to different patterns in the **regulatory region** upstream of the gene

This pattern is different for every gene

Gene Regulation in Biology

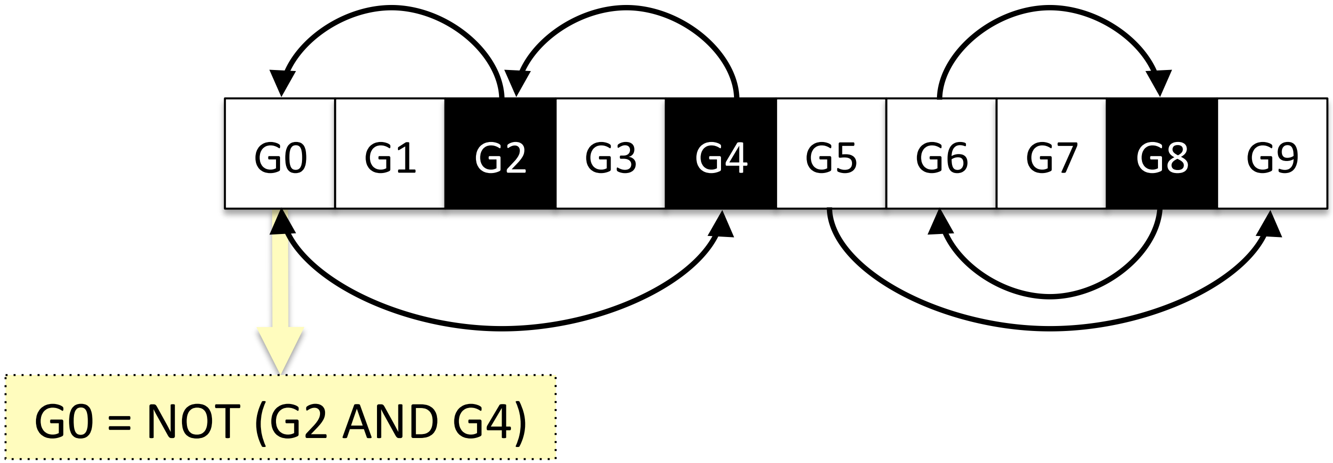


So, whether a gene is expressed depends on whether some other genes are expressed and whether some other genes are not expressed – this is the gene's **regulatory function**

Often this can be captured by a Boolean function

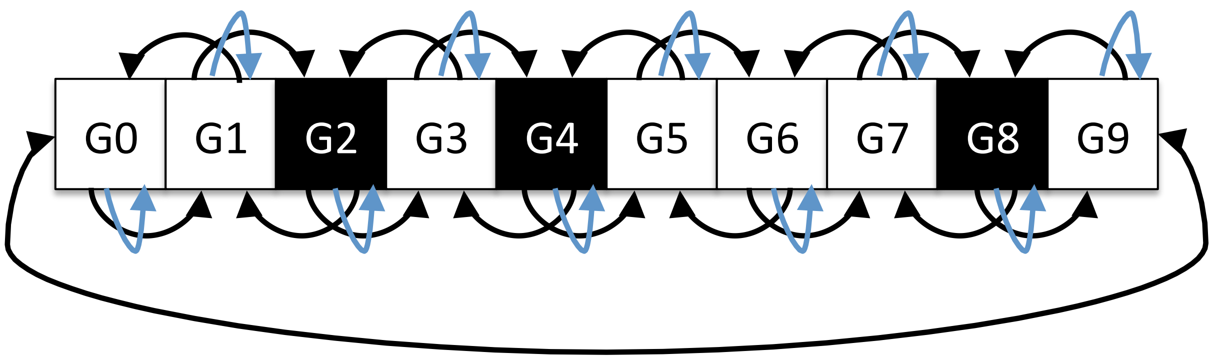
Boolean Network

- This idea can be modelled as a Boolean Network
  - A set of nodes (representing genes)
  - Each with a binary state (expressed or not expressed),
  - a set of input nodes (their regulating genes),
  - and a Boolean function (their regulatory function)
  - These are executed synchronously at each time step



Boolean Network

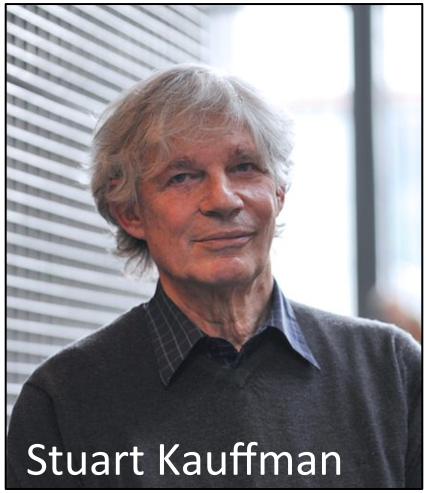
- Looks a bit like an elementary cellular automata
  - But without a fixed neighbourhood
  - And with a different update rule in each cell
  - In fact, a Boolean network is a generalisation of a CA
  - (i.e. a Boolean network can implement a CA ↓)
  - Therefore must be capable of universal computation



Random Boolean Networks

- Their behaviour can be studied statistically
  - By sampling and executing networks with particular sizes, connectivities and function sets
  - These are termed **Random Boolean Networks** (RBNs)

- Stuart Kauffman is known for this
  - He studied ***NK* networks**:
    - RBNs with *N* nodes
    - *K* inputs per node
    - A random function for each node
    - Also called Kauffman networks



http://en.wikipedia.org/wiki/File:Stuart_Kauffman.jpg

Boolean Functions

- For a particular value of K, there are $2^{(2^K)}$ functions
  - E.g. for K=2, there are 16 possible functions:

A	B	False	AND	A AND NOT B	A	NOT A AND B	B	XOR	OR
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

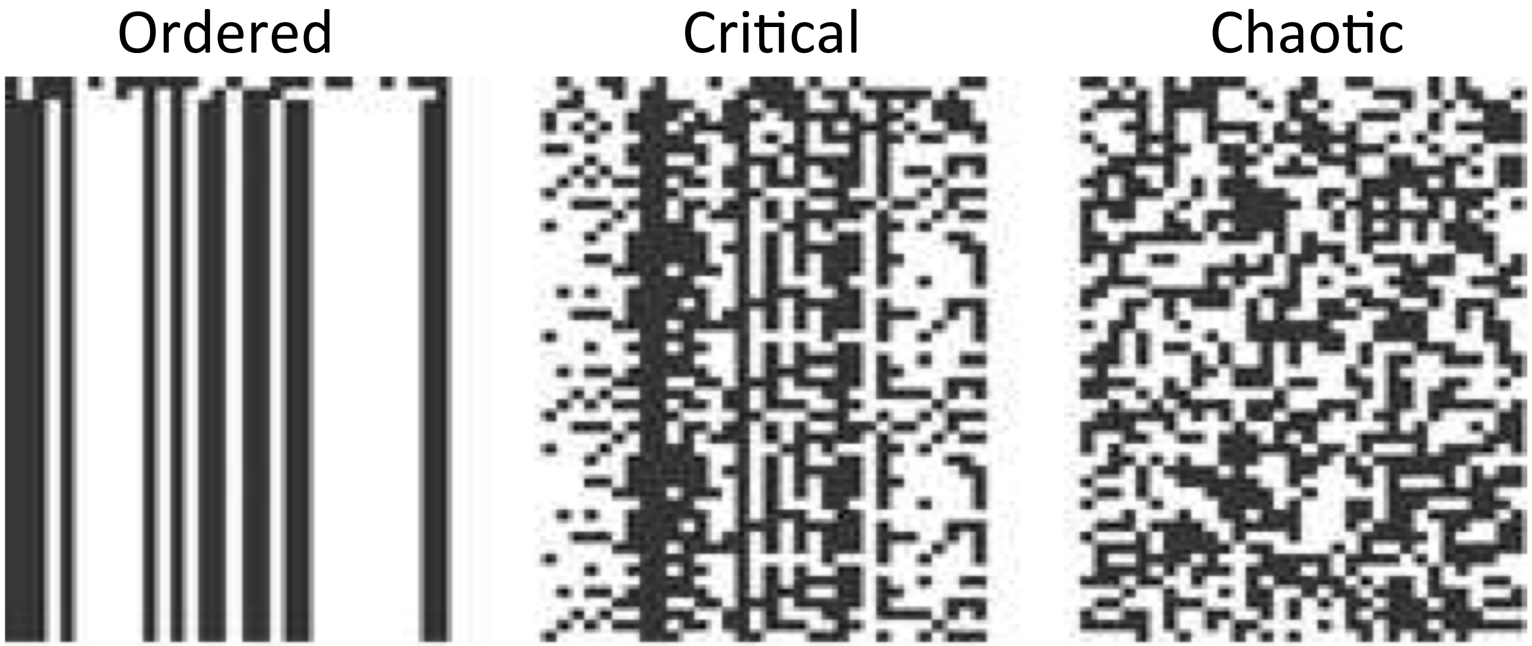
A	B	NOR	XNOR	NOT B	A OR NOT B	NOT A	NOT A OR B	NAND	True
0	0	1	1	1	1	0	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Kauffman's NK Networks

- There are *very* many networks for each NK
  - In fact, there are $$\left(\frac{2^{2^K} N!}{(N-K)!}\right)^N$$
    - for K=2, N=10, there are ~2 million
    - for K=3, N=10, there are ~6x10^{15}, which is *a lot*
  - This is why they are studied statistically
    - Too many to study them exhaustively, as Wolfram did with elementary CAs

Kauffman's NK Networks

- The behaviour of an RBN falls into 3 categories:

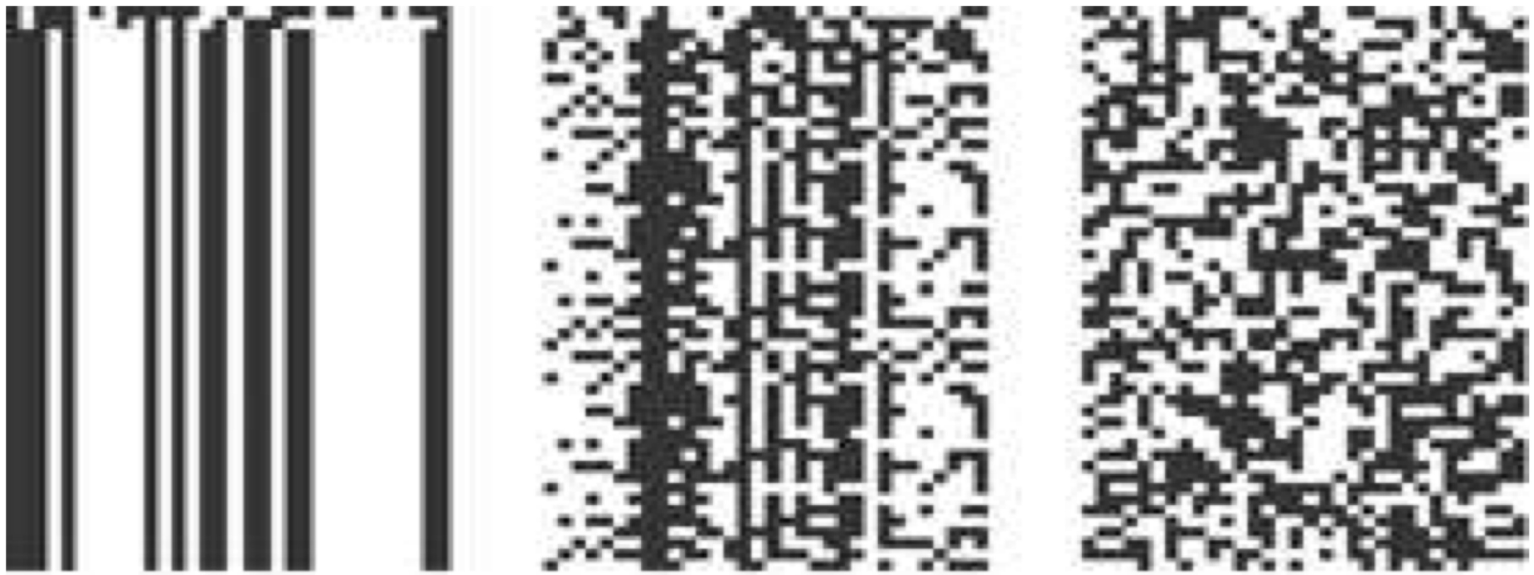


From [Gershenson, 2004] Introduction to Random Boolean Networks
http://uk.arxiv.org/abs/nlin.AO/0408006

Kauffman's NK Networks

- Kauffman observed that, *on average**

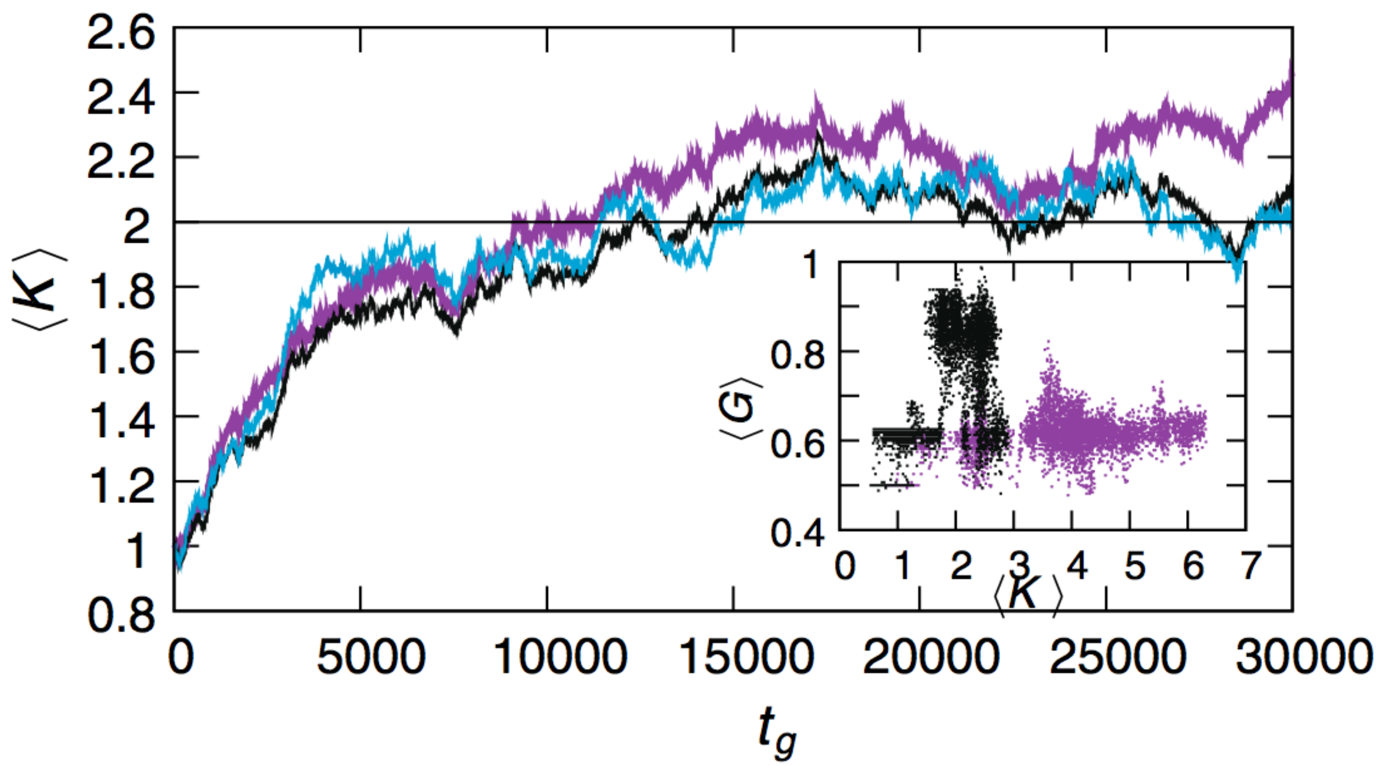
Ordered when K<2 | Critical when K=2 | Chaotic when K>2



*But note this doesn't mean that **all** K=2 networks are critical, or that critical networks can't be found for K>2

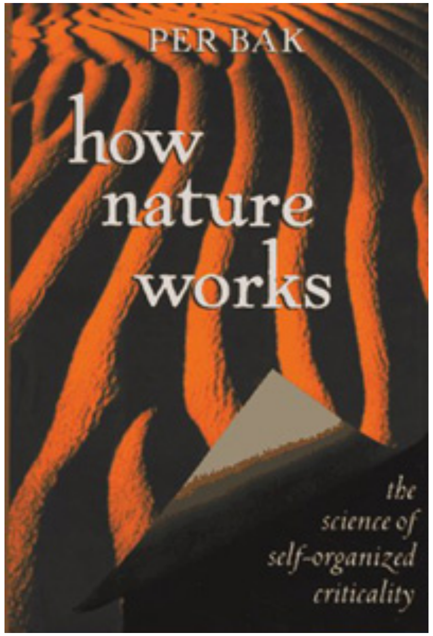
Evolved Criticality

- Evolved Boolean networks appear to favour criticality
  - Several studies have shown this, with K tending to 2
  - K=2 also promotes learning and generality [Goudarzi'12]:



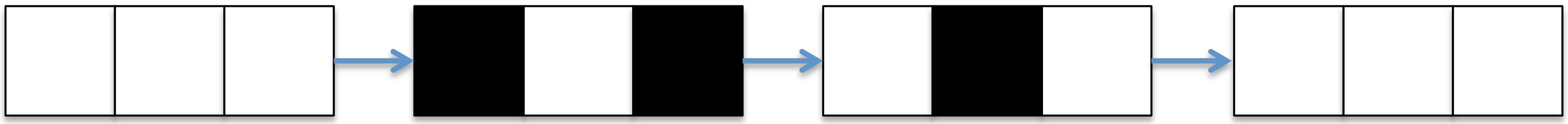
Edge of Chaos

- Criticality is also known as the ‘edge of chaos’
  - Dynamics are neither ordered nor chaotic, but complex
  - Hypothesised to be the sweet spot for computation
  - This is where CA Rule 110 (universality) is found
- Appears frequently in natural systems
  - Genes, brains, proteins, flocks …
  - Evolution may select for criticality
  - Per Bak, “How Nature Works” ➔
  - “Are biological systems poised at criticality?” [http://arxiv.org/pdf/1012.2242v1.pdf]



Attractors

- Attractors are an important concept for RBNs
  - A finite number of nodes means that states must repeat
  - An attractor of length *L* repeats every *L* steps
  - e.g. *L*=3:

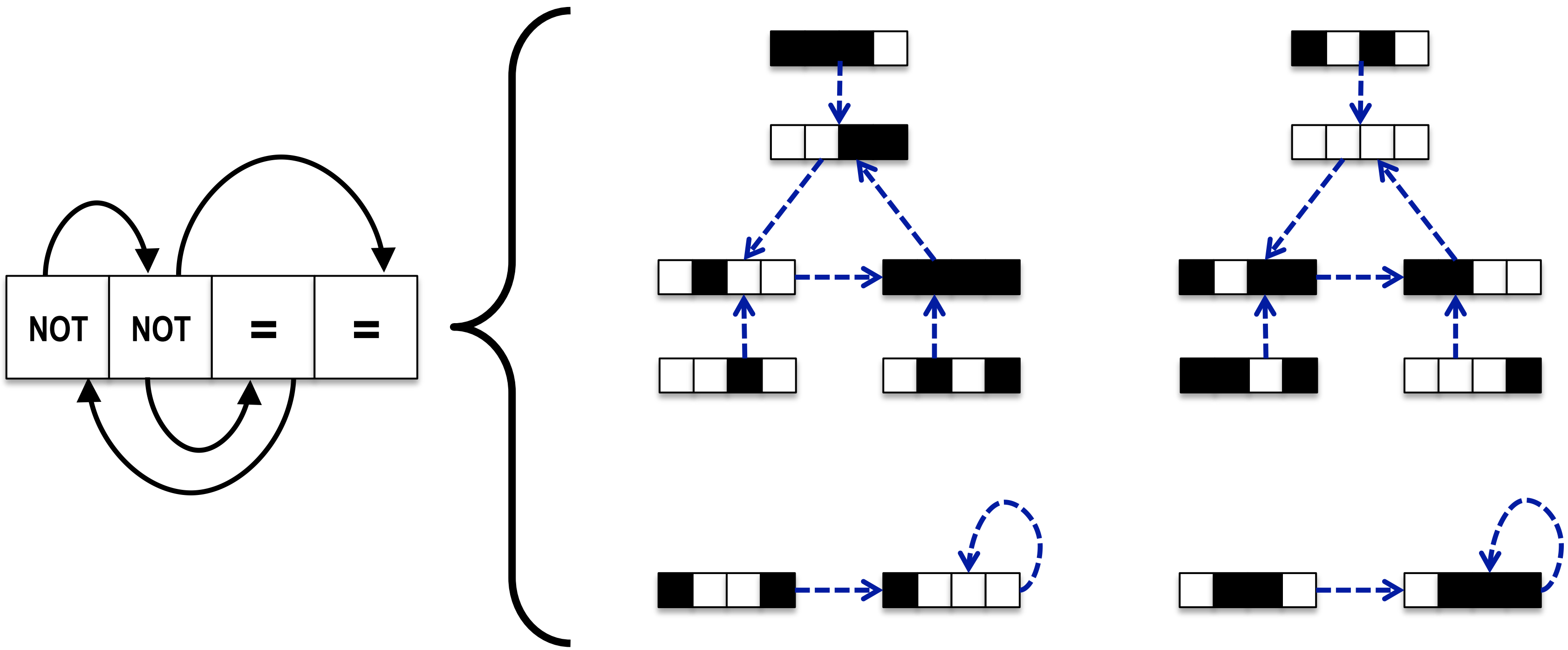


  - An attractor of length 1 is termed a point attractor

- Transients occur before an attractor is reached
  - e.g. “Acorn” is a transient leading to a stable attractor

Attractors

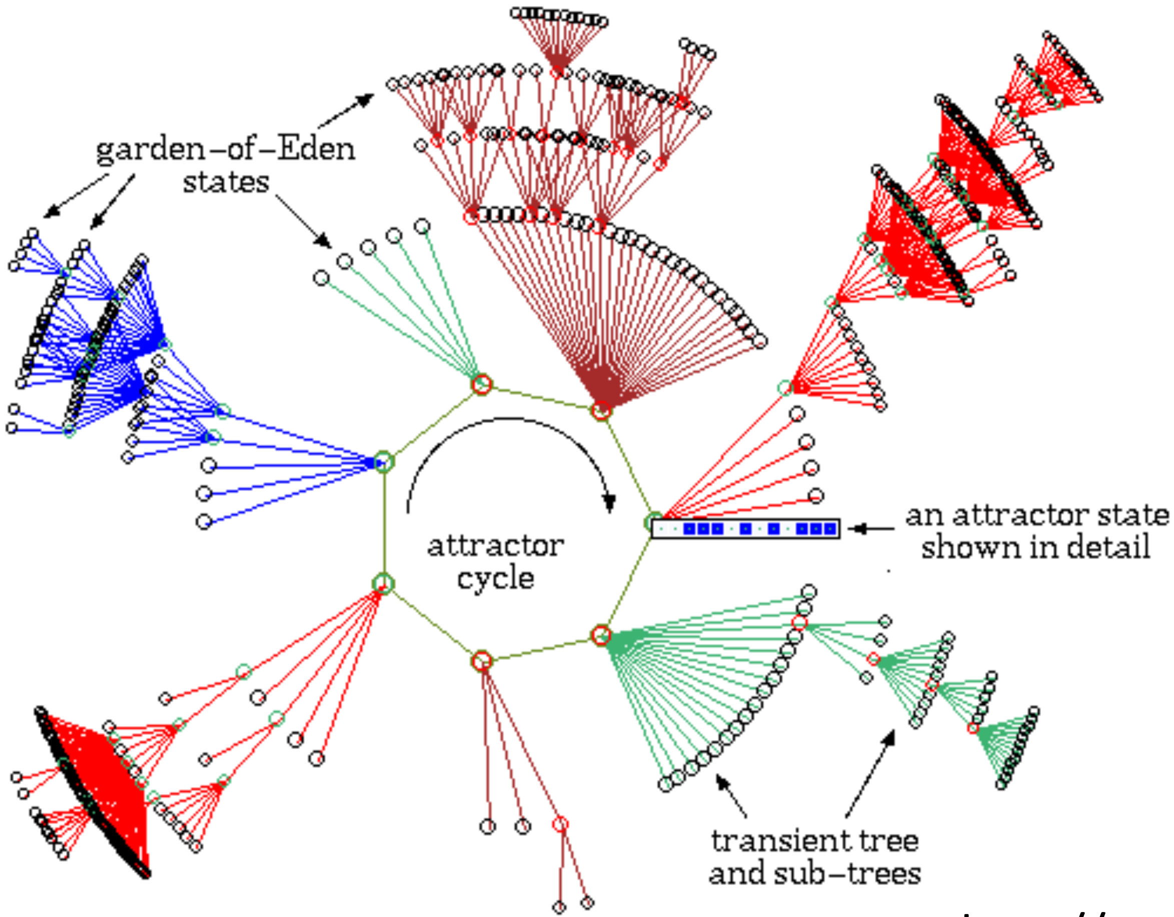
- Complete attractor map for a small network



  - 2 attractors of length 3, and 2 point attractors

Attractors

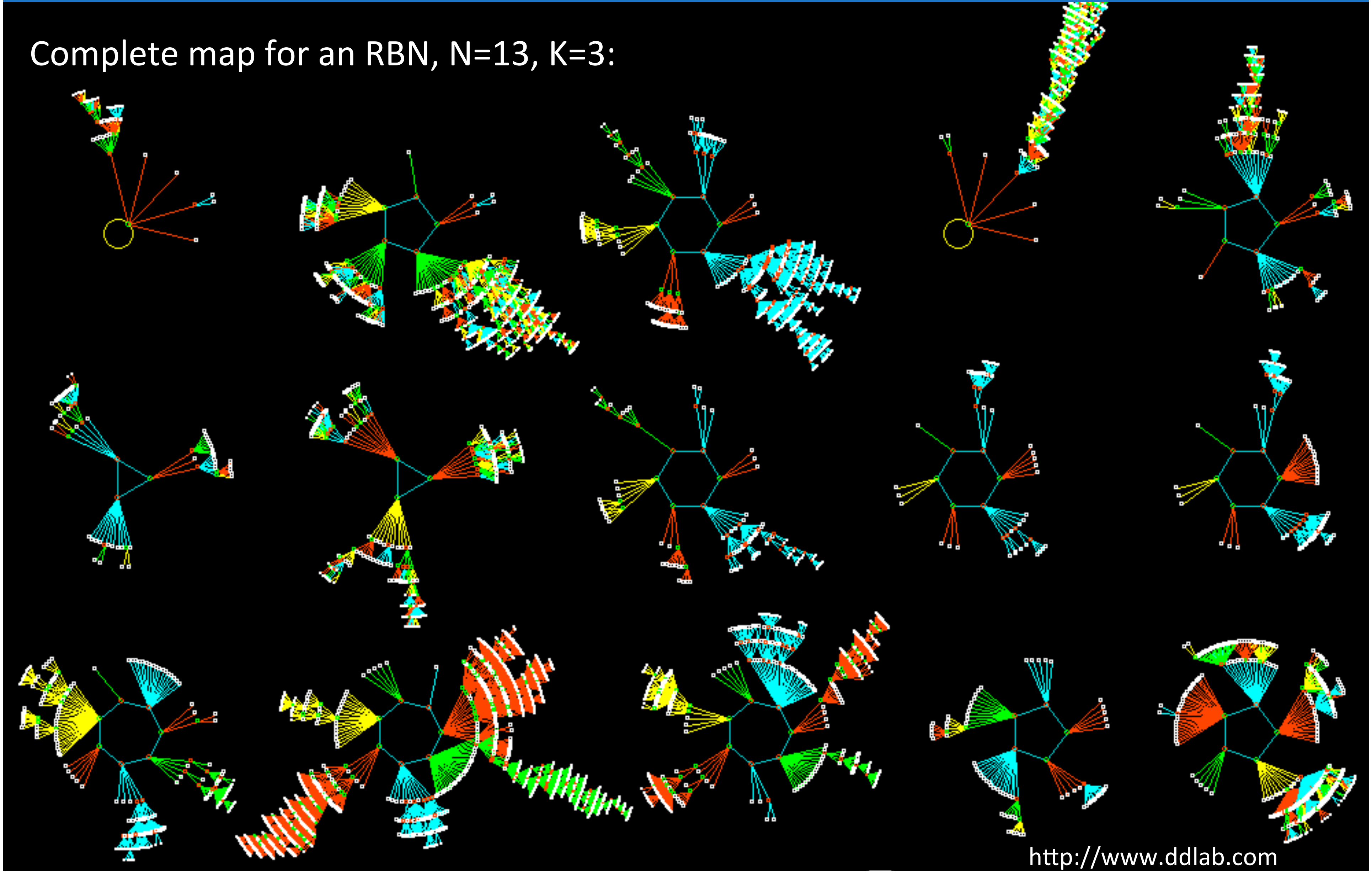
- Attractors often have large **basins of attraction**



http://www.ddlab.com

Attractors

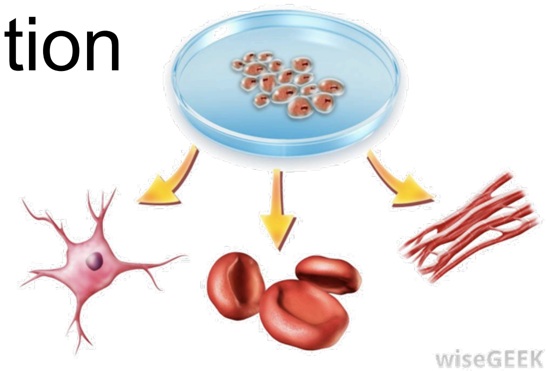
Complete map for an RBN, N=13, K=3:



http://www.ddlab.com

Attractors

- Attractors have a biological interpretation
  - Stable states for a cell
    e.g. cell types: neuron, liver, skin, …
  - Cancer can also be seen as an attractor

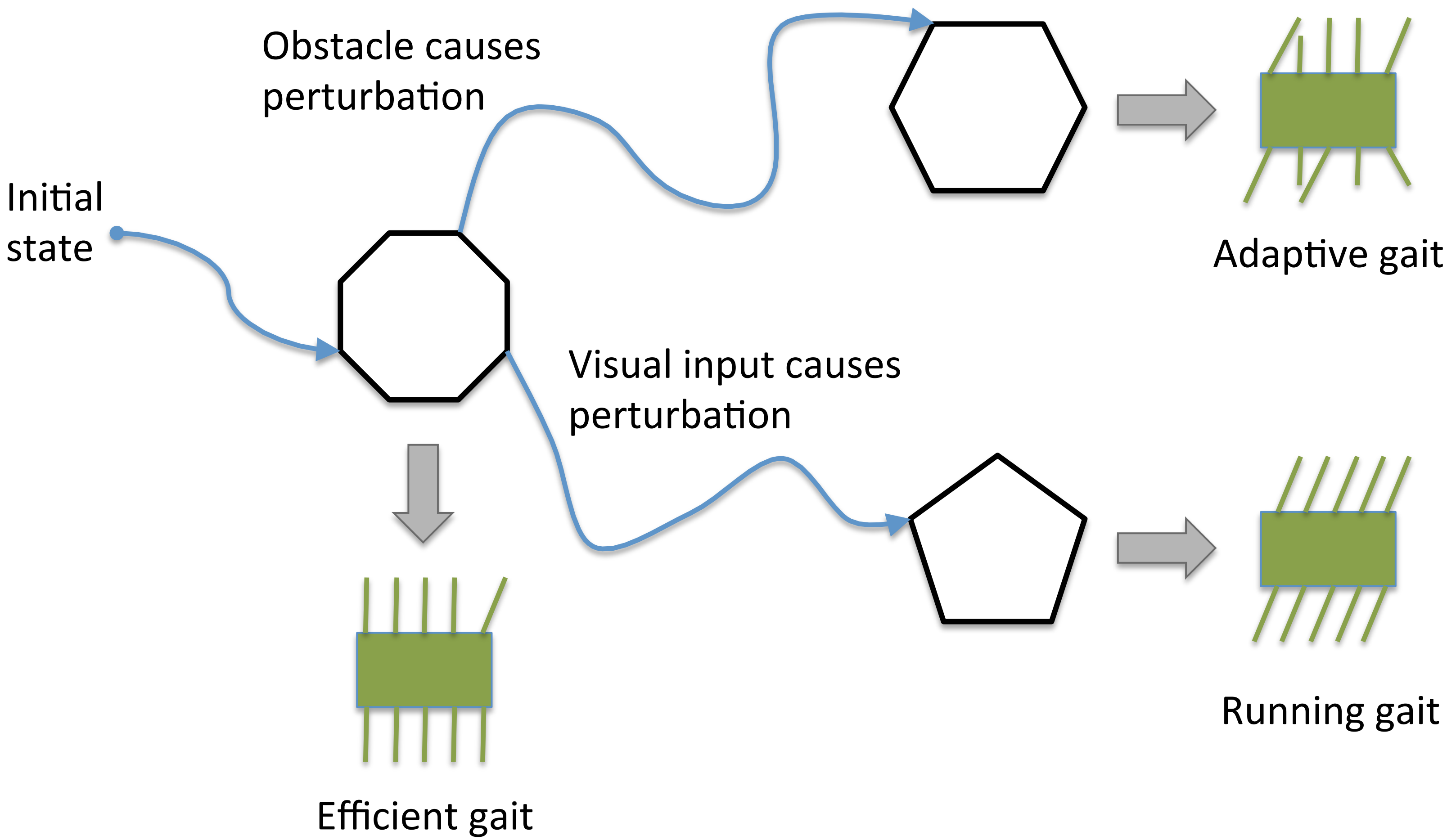


- Boolean networks are used to study bio GRNs
  - Captures their qualitative dynamics
  - Usually using NOT, OR and AND functions
  - See "Boolean modeling of biological regulatory networks: A methodology tutorial": http://www.sciencedirect.com/science/article/pii/S1046202312002770

Attractors

- And a computational interpretation?
  - An attractor can be seen as a computational state
  - e.g. a point attractor represents a particular ‘output’
- A cyclic attractor generates a repeating sequence
  - Which can be interpreted as a sequence of actions
  - e.g. robot actuator movements
- Transients can be seen as computing the attractor
  - Either from the initial state
  - Or between attractors (especially if inputs are allowed)

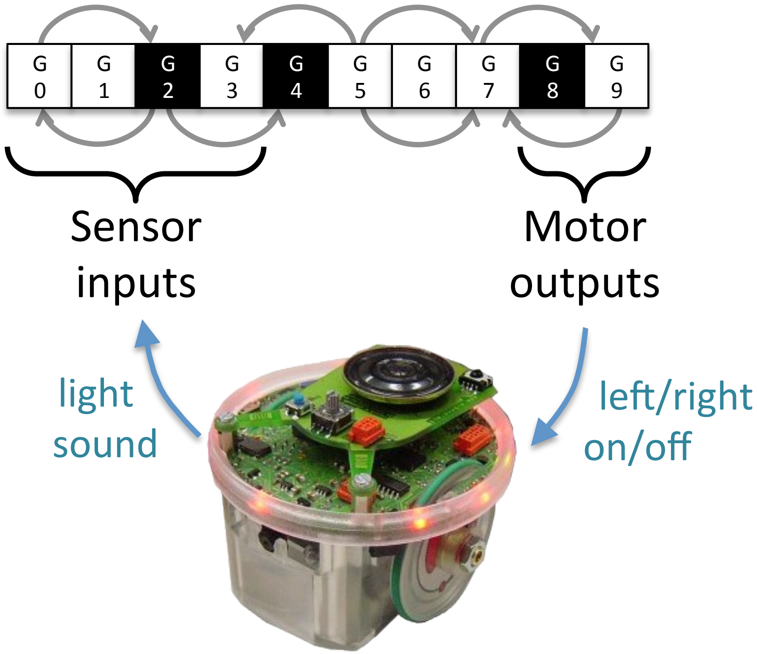
Illustrative Example



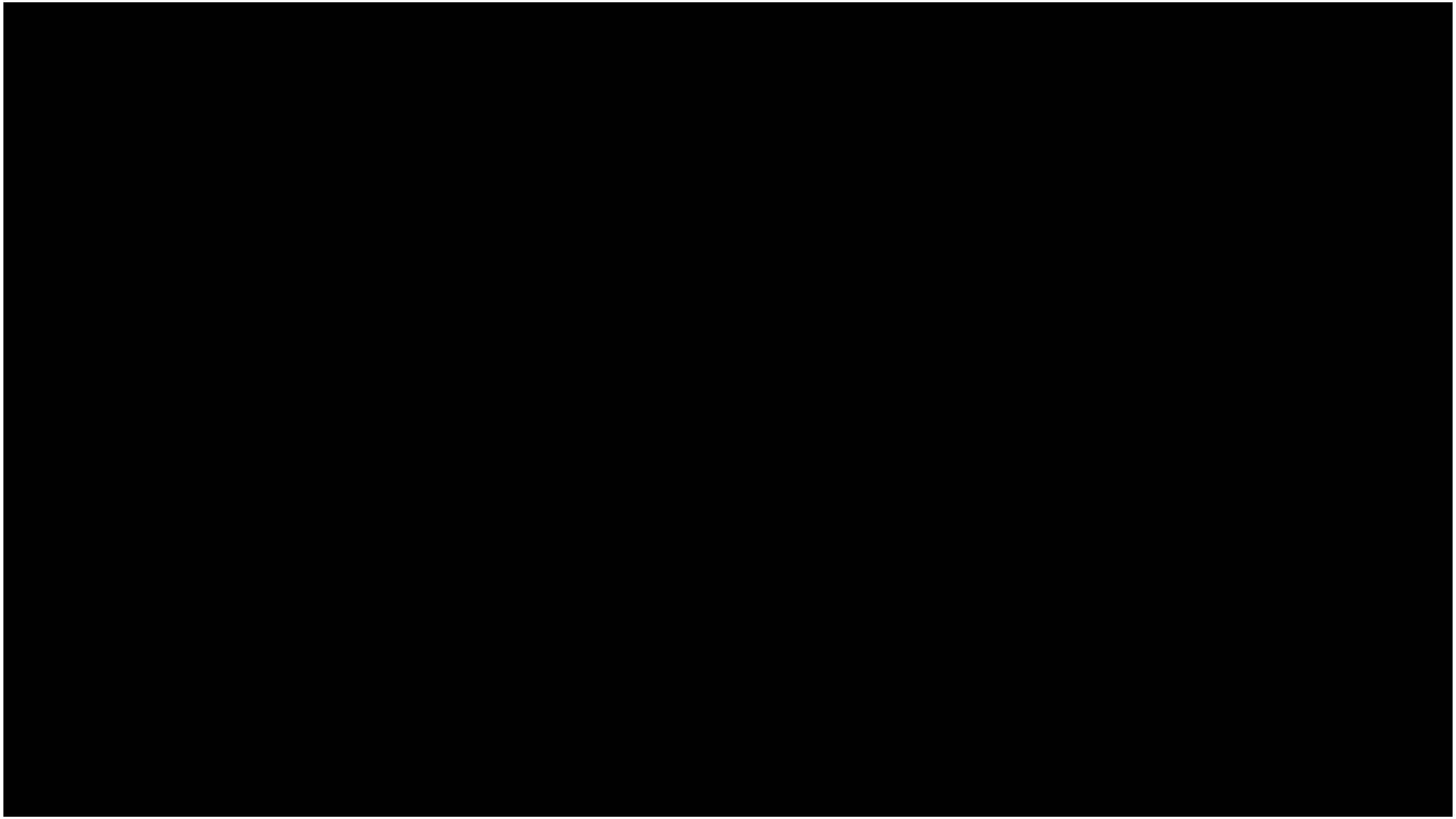
Evolving Boolean Networks

- Using EAs to find useful Boolean networks
  - Networks that solve a particular computational problem
  - Exhaustive search is not an option in this case
  - Another form of genetic programming

- Robot controllers [Roli, 2011]
  - Used inputs and outputs ➔
  - Searched for networks with light following behaviour
  - Controlled a real ePuck
  - Networks were robust to sensor noise



Roli et al.



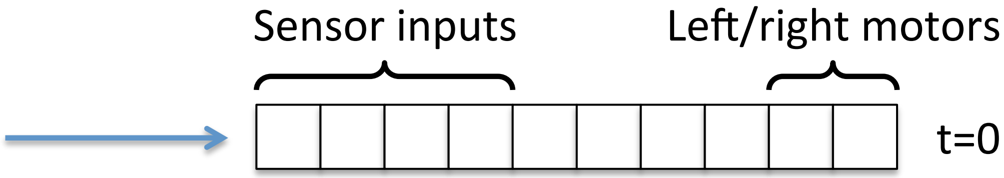
https://www.youtube.com/watch?v=6ZF9Ijpwkd8

Inputs and Outputs

Initial State

No sensor signals

Not moving



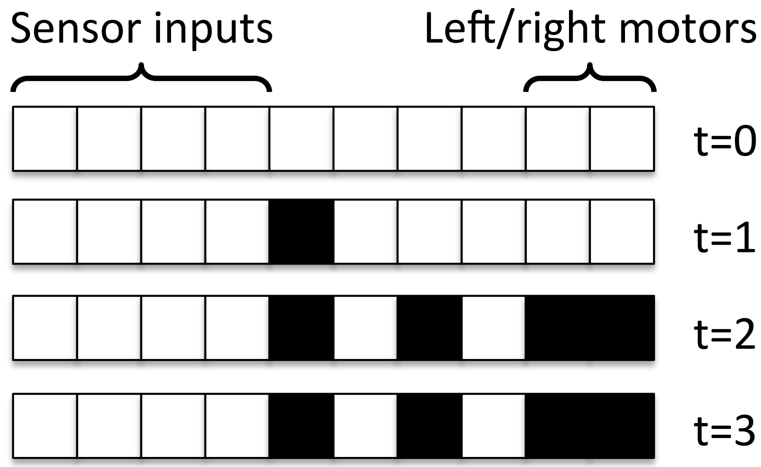
Note: This is a simplified example. See paper for full details of how this works.

Inputs and Outputs

Initial State

No sensor signals

Not moving

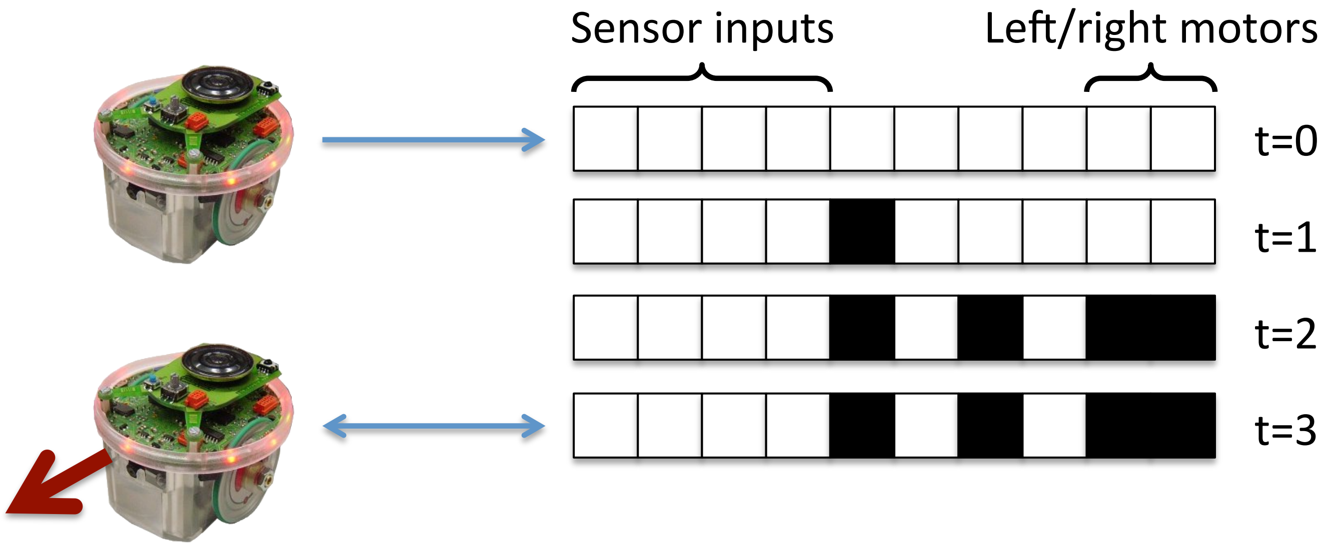


Note: This is a simplified example. See paper for full details of how this works.

Inputs and Outputs

Initial State
No sensor signals
Not moving

First Update
No sensor signals
Moving forward



Note: This is a simplified example. See paper for full details of how this works.

Inputs and Outputs

Initial State

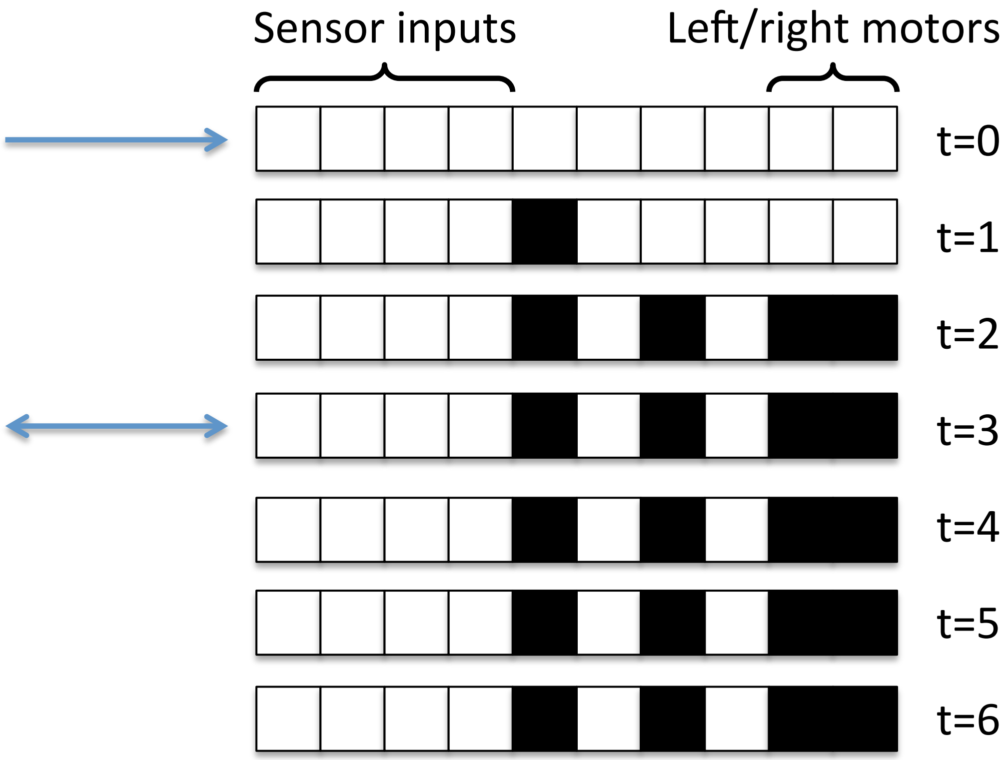
No sensor signals

Not moving

First Update

No sensor signals

Moving forward



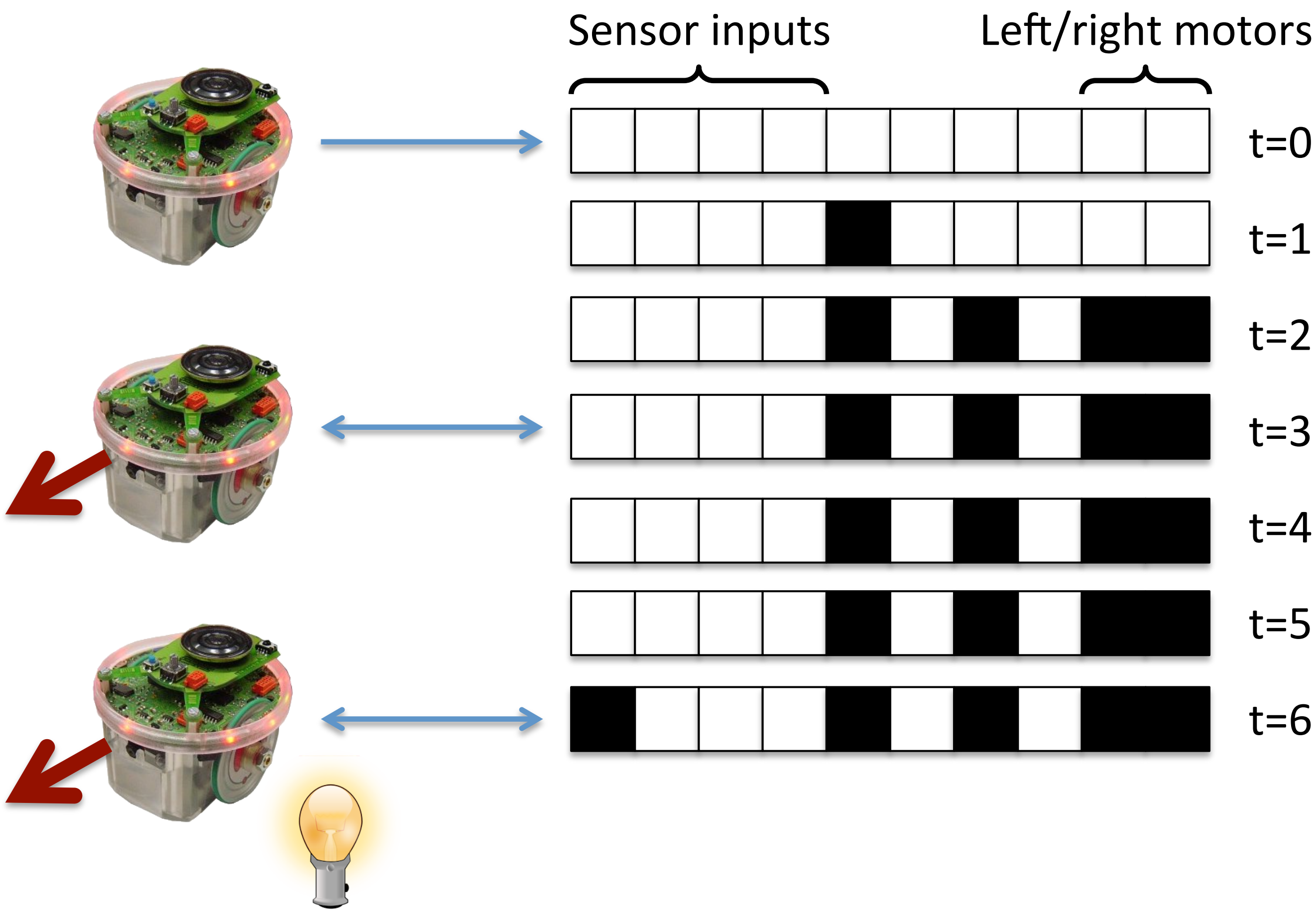
Note: This is a simplified example. See paper for full details of how this works.

Inputs and Outputs

Initial State
No sensor signals
Not moving

First Update
No sensor signals
Moving forward

Second Update
Light on left
Moving forward



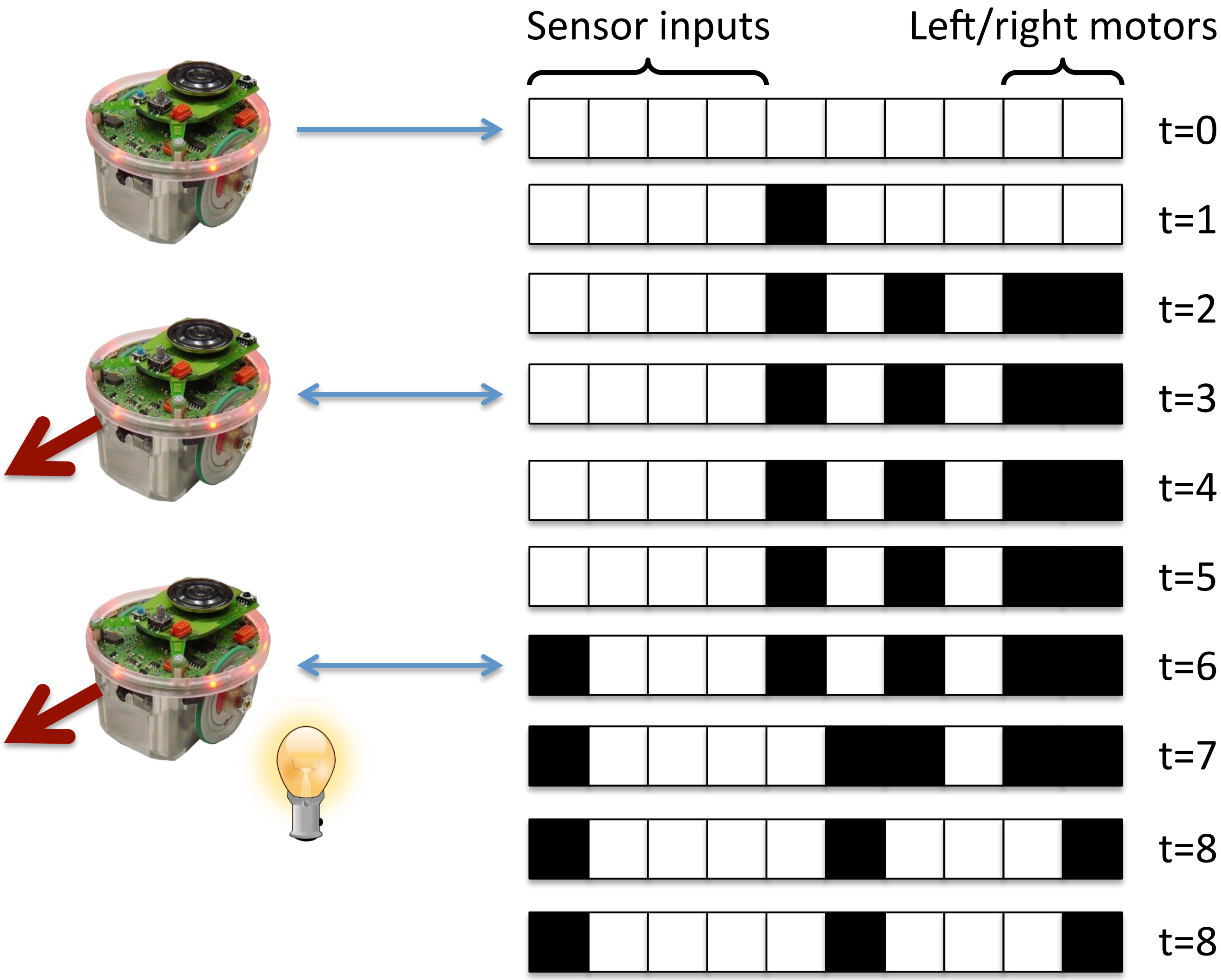
Note: This is a simplified example. See paper for full details of how this works.

Inputs and Outputs

Initial State
No sensor signals
Not moving

First Update
No sensor signals
Moving forward

Second Update
Light on left
Moving forward



Note: This is a simplified example. See paper for full details of how this works.

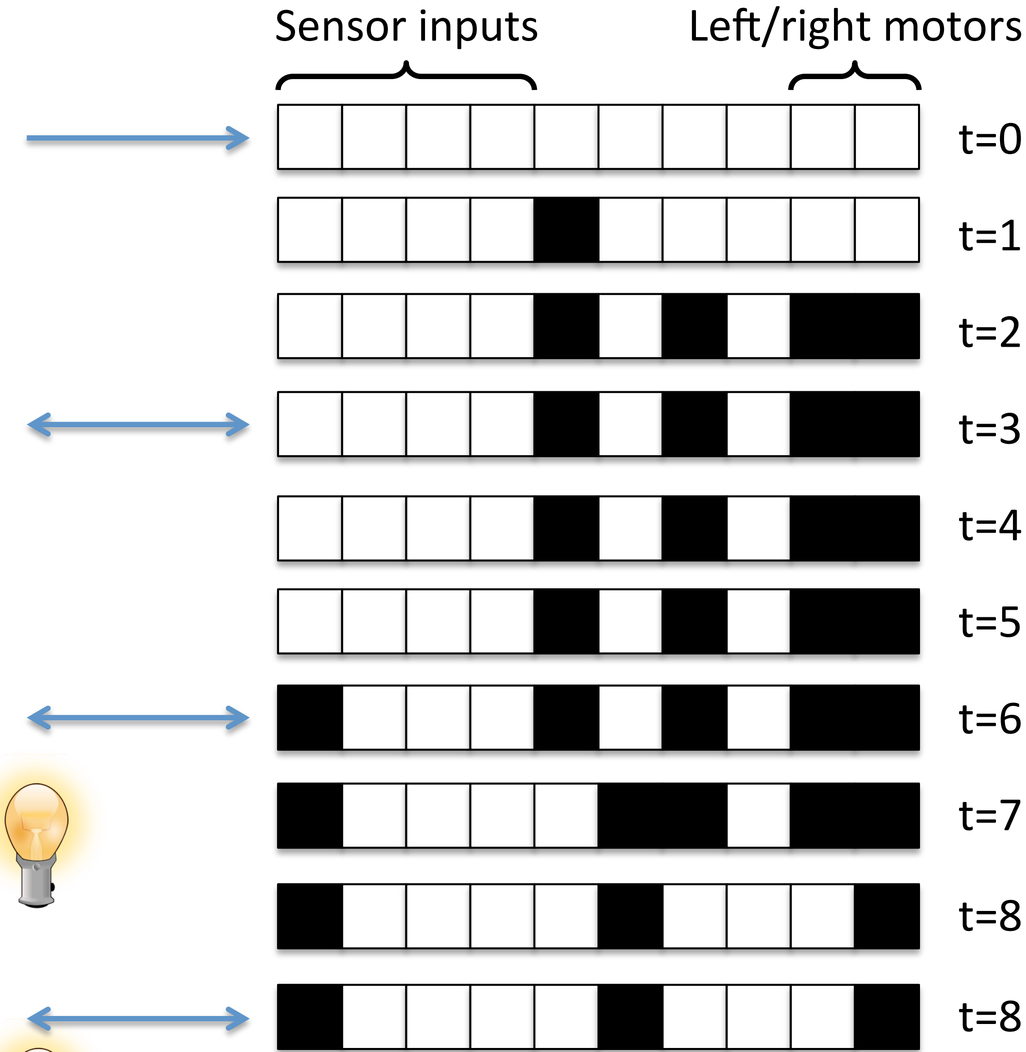
Inputs and Outputs

Initial State
No sensor signals
Not moving

First Update
No sensor signals
Moving forward

Second Update
Light on left
Moving forward

Third Update
Light on left
Turning left



Note: This is a simplified example. See paper for full details of how this works.

Limitations of Boolean Networks

- From a practical perspective
  - Inputs and outputs must be binary encoded
  - Difficult to handle large/continuous/many values
  - E.g. Pi in binary: 01000000010010010000111111111001

- From a biological perspective
  - Gene expression levels are not discrete
  - Regulatory functions are not always Boolean functions

  - *However, Boolean networks are computationally efficient and can be implemented directly in hardware*

To be continued…

- GRN models that look more biological
- Using GRN models for development
- Interesting applications

Bibliography

- W. Banzhaf. On the dynamics of an artificial regulatory network. Lecture Notes in Computer Science 2801, chapter 24, pages 217–227. Springer, 2003. http://link.springer.com/chapter/10.1007%2F978-3-540-39432-7_24
- S. Cussat-Blanc and J. B. Pollack, Using Pictures to Visualize the Complexity of Gene Regulatory Networks, ALIFE 2012. http://mitpress.mit.edu/sites/default/files/titles/content/alife13/ch064.html
- N. Flann et al., Biological development of cell patterns: Characterizing the space of cell chemistry genetic regulatory networks, in *Advances in Artificial Life*, LNCS 3630, 2005. http://dx.doi.org/10.1007/11553090_7
- C. Gershenson, Introduction to random Boolean networks, 2004. http://arxiv.org/abs/nlin/0408006
- A. Goudarzi et al, Emergent Criticality through Adaptive Information Processing in Boolean Networks, *Physical Review Letters* 108(12), 2012. http://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1109&context=ece_fac
- Y. Jin and Y. Meng. Morphogenetic Robotics. *Morphogenetic Engineering*, 2012. http://link.springer.com/chapter/10.1007%2F978-3-642-33902-8_3
- M. Joachimczak et al, Evolving morphologies and controllers for soft-bodied multicellular animats using gene regulatory networks and artificial embryogenesis. GECCO'12, 2012 http://doursat.free.fr/docs/Joachimczak_et_al_2012_animats_GECCO.pdf
- A. Roli et al, Boolean network robotics: a proof of concept, 2011 http://arxiv.org/abs/1101.6001
- S. Sanchez and S. Cussat-Blanc. Gene regulated car driving: using a gene regulatory network to drive a virtual car. *Genetic Programming and Evolvable Machines* 15(4):477–511, 2014. http://link.springer.com/article/10.1007%2Fs10710-014-9228-y
- M. Trefzer et al., Image compression of natural images using artificial gene regulatory networks, GECCO'10, 2010 http://www-users.york.ac.uk/~mt540/pdfs/trefzer2010_gci.pdf