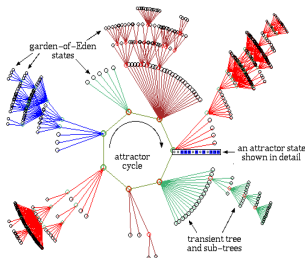


Boolean Networks as a Model of Gene Regulation

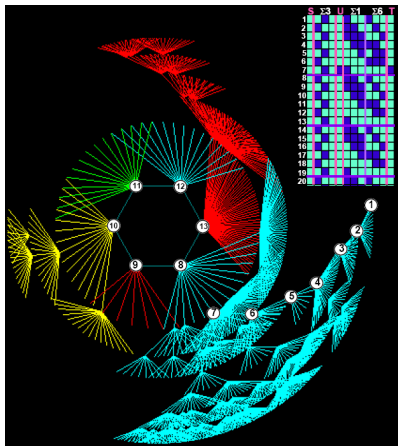
Dag Gillberg

Simon Fraser University

Feb 10, 2006



- Motivation
- Biological Framework
- Introduction of the “Kauffman Model” — Boolean Networks as a model for the (gene) regulation in cells
- Comparison with reality
- Current research and improvements to the model



Some numbers:

- The human body contains about 6×10^{13} cells

Some numbers:

- The human body contains about 6×10^{13} cells
- There are 256 different cell types that are quite different (e.g. brain cell, muscle cell, lung cell)

Some numbers:

- The human body contains about 6×10^{13} cells
- There are 256 different cell types that are quite different (e.g. brain cell, muscle cell, lung cell)
- Each cell contain a copy of the full DNA:
 - 2 meters of DNA curled up in chromosomes
 - 2.9 billion base pairs (A,T,C,G)
 - 3% of these make up 25,000 ative genes

Some numbers:

- The human body contains about 6×10^{13} cells
- There are 256 different cell types that are quite different (e.g. brain cell, muscle cell, lung cell)
- Each cell contain a copy of the full DNA:
 - 2 meters of DNA curled up in chromosomes
 - 2.9 billion base pairs (A,T,C,G)
 - 3% of these make up 25,000 ative genes
- Stringing all our DNA together makes a chain of length 800 AU
 - 20 times the distance between the sun and Pluto!

Some numbers:

- The human body contains about 6×10^{13} cells
- There are 256 different cell types that are quite different (e.g. brain cell, muscle cell, lung cell)
- Each cell contain a copy of the full DNA:
 - 2 meters of DNA curled up in chromosomes
 - 2.9 billion base pairs (A,T,C,G)
 - 3% of these make up 25,000 ative genes
- Stringing all our DNA together makes a chain of length 800 AU
 - 20 times the distance between the sun and Pluto!

Some Questions:

Some numbers:

- The human body contains about 6×10^{13} cells
- There are 256 different cell types that are quite different (e.g. brain cell, muscle cell, lung cell)
- Each cell contain a copy of the full DNA:
 - 2 meters of DNA curled up in chromosomes
 - 2.9 billion base pairs (A,T,C,G)
 - 3% of these make up 25,000 ative genes
- Stringing all our DNA together makes a chain of length 800 AU
 - 20 times the distance between the sun and Pluto!

Some Questions:

- What a mess! How are all these genes regulated?

Some numbers:

- The human body contains about 6×10^{13} cells
- There are 256 different cell types that are quite different (e.g. brain cell, muscle cell, lung cell)
- Each cell contain a copy of the full DNA:
 - 2 meters of DNA curled up in chromosomes
 - 2.9 billion base pairs (A,T,C,G)
 - 3% of these make up 25,000 ative genes
- Stringing all our DNA together makes a chain of length 800 AU
 - 20 times the distance between the sun and Pluto!

Some Questions:

- What a mess! How are all these genes regulated?
- How come cell types are so different when they all contain the same information?

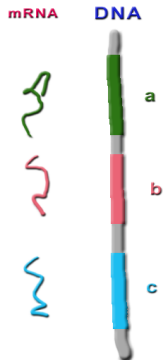
Gene Expression

- 3 active genes
- Each gene *transcribed* to mRNA (~ 1 s)
- The mRNA is *translated* to proteins (~ 1 min)
- $\rightarrow$ high concentration of proteins A, B and C in the cell



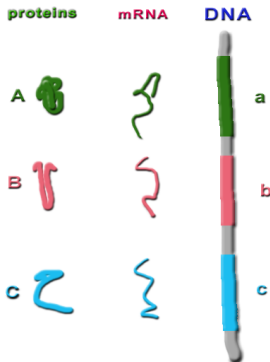
Gene Expression

- 3 active genes
- Each gene *transcribed* to mRNA (~ 1 s)
- The mRNA is *translated* to proteins (~ 1 min)
- $\rightarrow$ high concentration of proteins A, B and C in the cell



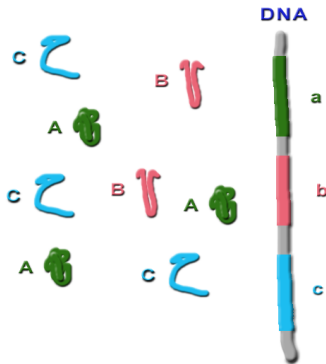
Gene Expression

- 3 active genes
- Each gene *transcribed* to mRNA (~ 1 s)
- The mRNA is *translated* to proteins (~ 1 min)
- $\rightarrow$ high concentration of proteins A, B and C in the cell



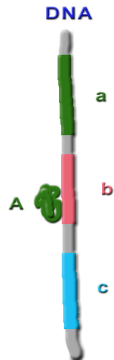
Gene Expression

- 3 active genes
- Each gene *transcribed* to mRNA (~ 1 s)
- The mRNA is *translated* to proteins (~ 1 min)
- → high concentration of proteins A, B and C in the cell



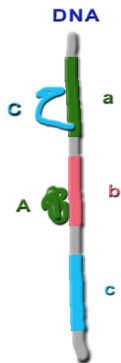
Gene Regulation (Example)

- protein A binds to gene b
→ gene b in-activated
→ no more production of protein B
- protein C want to bind to gene a, however..
...protein B inhibits C from binding to a
→ gene a still expressed!
- gene c not regulated (not realistic)



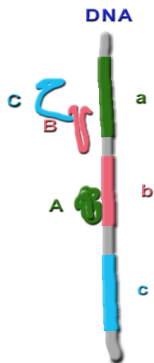
Gene Regulation (Example)

- protein A binds to gene b
→ gene b in-activated
→ no more production of protein B
- protein C want to bind to gene a, however..
...protein B inhibits C from binding to a
→ gene a still expressed!
- gene c not regulated (not realistic)



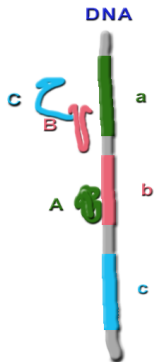
Gene Regulation (Example)

- protein A binds to gene b
→ gene b in-activated
→ no more production of protein B
- protein C want to bind to gene a, however..
...protein B inhibits C
from binding to a
→ gene a still expressed!
- gene c not regulated (not realistic)



Gene Regulation (Example)

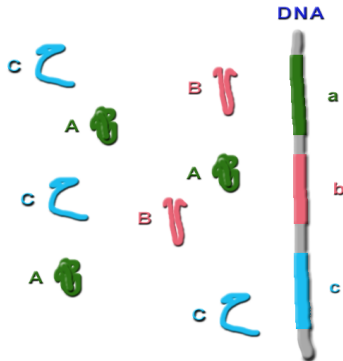
- protein A binds to gene b
→ gene b in-activated
→ no more production of protein B
- protein C want to bind to gene a, however..
...protein B inhibits C from binding to a
→ gene a still expressed!
- gene c not regulated (not realistic)



Biological Framework

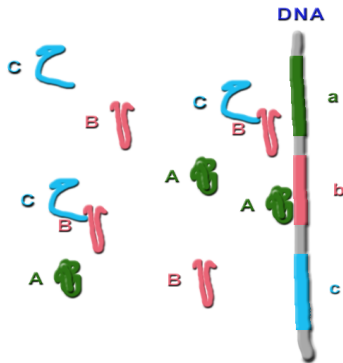
Gene Expression Dynamics

- 1 Initially – all genes expressed → all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated → more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** → **a** in-activated
- 4 “no more” **A** → **b** expressed again!
- 5 We have **B** → **a** expressed. Back at 1!



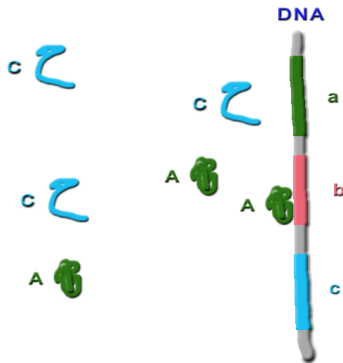
Gene Expression Dynamics

- 1 Initially – all genes expressed $\rightarrow$ all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated $\rightarrow$ more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** $\rightarrow$ **a** in-activated
- 4 “no more” **A** $\rightarrow$ **b** expressed again!
- 5 We have **B** $\rightarrow$ **a** expressed. Back at 1!



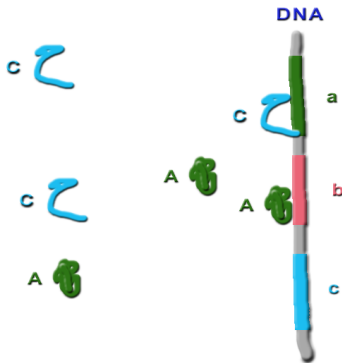
Gene Expression Dynamics

- 1 Initially – all genes expressed $\rightarrow$ all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated $\rightarrow$ more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** $\rightarrow$ **a** in-activated
- 4 “no more” **A** $\rightarrow$ **b** expressed again!
- 5 We have **B** $\rightarrow$ **a** expressed. Back at 1!



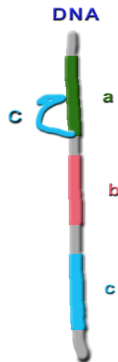
Gene Expression Dynamics

- 1 Initially – all genes expressed $\rightarrow$ all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated $\rightarrow$ more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** $\rightarrow$ **a** in-activated
- 4 “no more” **A** $\rightarrow$ **b** expressed again!
- 5 We have **B** $\rightarrow$ **a** expressed. Back at 1!



Gene Expression Dynamics

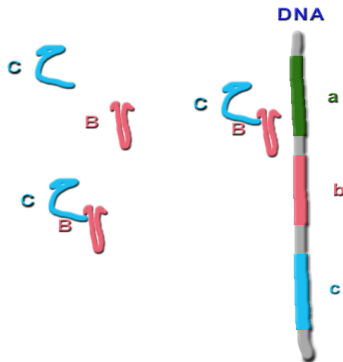
- 1 Initially – all genes expressed $\rightarrow$ all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated $\rightarrow$ more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** $\rightarrow$ **a** in-activated
- 4 “no more” **A** $\rightarrow$ **b** expressed again!
- 5 We have **B** $\rightarrow$ **a** expressed. Back at 1!



Biological Framework

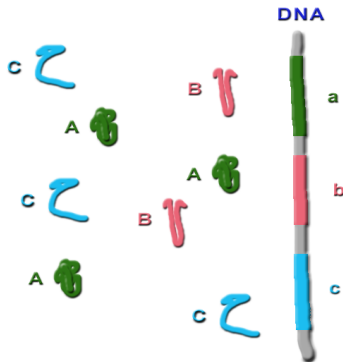
Gene Expression Dynamics

- 1 Initially – all genes expressed $\rightarrow$ all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated $\rightarrow$ more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** $\rightarrow$ **a** in-activated
- 4 “no more” **A** $\rightarrow$ **b** expressed again!
- 5 We have **B** $\rightarrow$ **a** expressed. Back at 1!



Gene Expression Dynamics

- 1 Initially – all genes expressed $\rightarrow$ all proteins are produced
- 2 When we have **A** in the cell **b** is in-activated $\rightarrow$ more **B** production and **B** concentration drops
- 3 After some time there are much more **C** than **B** $\rightarrow$ **a** in-activated
- 4 “no more” **A** $\rightarrow$ **b** expressed again!
- 5 We have **B** $\rightarrow$ **a** expressed. **Back at 1!**



The Kauffman Model

- Stuart Kauffman presented this model in 1967
- Main attention in 1993 when publishing “The Origins of Order: Self-organization and Selection in Evolution” (700 pages).

The Kauffman Model

- Stuart Kauffman presented this model in 1967
- Main attention in 1993 when publishing “The Origins of Order: Self-organization and Selection in Evolution” (700 pages).

The model is based on:

Boolean Rules

$$x_i(t+1) = f_i(x_1(t), x_2(t), \dots, x_N(t)) \in (0, 1)$$

The Kauffman Model

- Stuart Kauffman presented this model in 1967
- Main attention in 1993 when publishing “The Origins of Order: Self-organization and Selection in Evolution” (700 pages).

The model is based on:

Boolean Rules

$$x_i(t+1) = f_i(x_1(t), x_2(t), \dots, x_N(t)) \in (0, 1)$$

“The Equation of Life”

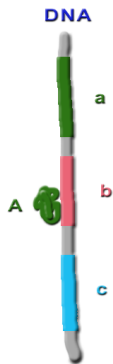
The Kauffman Model

Gene Expression

- According to the Kauffman model, a gene is either “on” or “off”
- The gene expression at time t is written as $\vec{x}(t)$
- The state to the right correspond to

$$\vec{x} = (1, 0, 1)$$

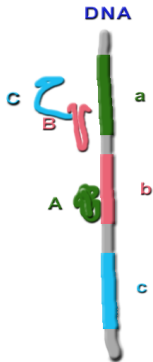
$$(x_1 = a = \text{on})$$



The Kauffman Model

Gene Regulation

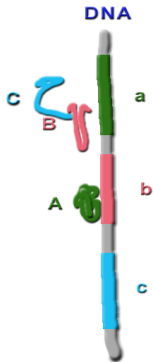
- The **boolean rules** are the central part of the model
- Each gene x_i has a boolean rule f_i
- Iterative process:
$$x_i = f_i(x_1, x_2, \dots, x_N)$$
- The rules in the previous example:
$$f_1 = (\text{NOT } x_3) \text{ OR } x_2$$
$$f_2 = \text{NOT } x_1$$
$$f_3 = \text{true}$$
- The **connectivities** are
 $K_1 = 2, K_2 = 1, K_3 = 0$



The Kauffman Model

Gene Regulation

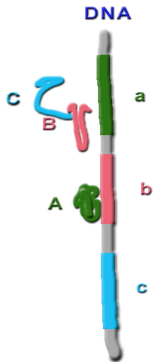
- The **boolean rules** are the central part of the model
- Each gene x_i has a boolean rule f_i
- Iterative process:
$$x_i = f_i(x_1, x_2, \dots, x_N)$$
- The rules in the previous example:
$$f_1 = (\text{NOT } x_3) \text{ OR } x_2$$
$$f_2 = \text{NOT } x_1$$
$$f_3 = \text{true}$$
- The **connectivities** are
 $K_1 = 2, K_2 = 1, K_3 = 0$



The Kauffman Model

Gene Regulation

- The **boolean rules** are the central part of the model
- Each gene x_i has a boolean rule f_i
- Iterative process:
$$x_i = f_i(x_1, x_2, \dots, x_N)$$
- The rules in the previous example:
$$f_1 = (\text{NOT } x_3) \text{ OR } x_2$$
$$f_2 = \text{NOT } x_1$$
$$f_3 = \text{true}$$
- The **connectivities** are
$$K_1 = 2, K_2 = 1, K_3 = 0$$

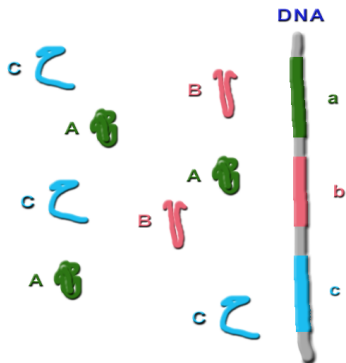


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

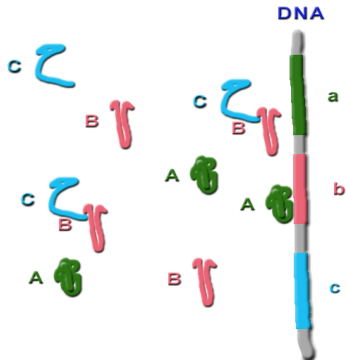


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

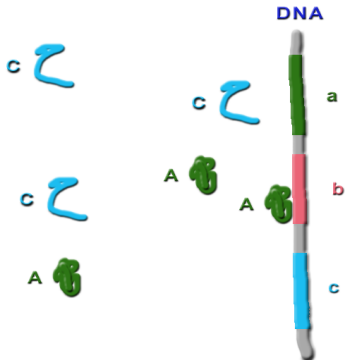


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

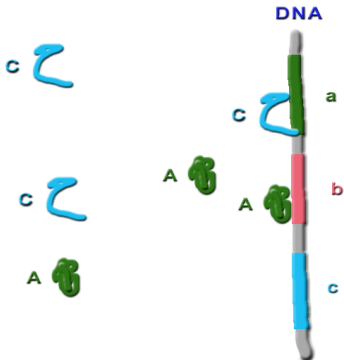


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

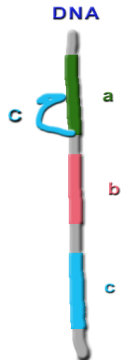


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

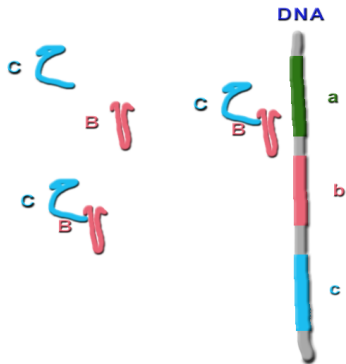


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

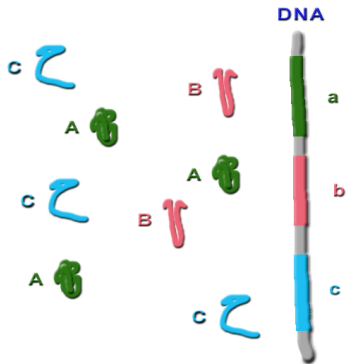


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)

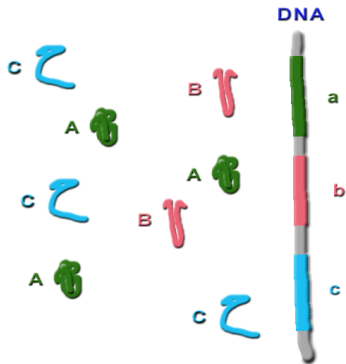


The Kauffman Model

Dynamics

- 1 $\vec{x}(t = 0) = (1, 1, 1)$
- 2 $\vec{x}(t = 1) = (1, 0, 1)$
- 3 $\vec{x}(t = 2) = (0, 0, 1)$
- 4 $\vec{x}(t = 3) = (0, 1, 1)$
- 5 $\vec{x}(t = 4) = \vec{x}(t = 0)$

Our system has **organised itself** by reaching an **attractor** (in this case a 4-loop)



The Kauffman Model

In “*The Origins of Order: Self-organization and Selection in Evolution*” Kauffman presents the following study (computer simulation):

The Kauffman Model

In “*The Origins of Order: Self-organization and Selection in Evolution*” Kauffman presents the following study (computer simulation):

- He looks at different systems of N genes

The Kauffman Model

In “*The Origins of Order: Self-organization and Selection in Evolution*” Kauffman presents the following study (computer simulation):

- He looks at different systems of N genes
- The N boolean rules are assigned **randomly**

The Kauffman Model

In “*The Origins of Order: Self-organization and Selection in Evolution*” Kauffman presents the following study (computer simulation):

- He looks at different systems of N genes
- The N boolean rules are assigned **randomly**
- Each rule has K inputs, where K is fixed

The Kauffman Model

In “*The Origins of Order: Self-organization and Selection in Evolution*” Kauffman presents the following study (computer simulation):

- He looks at different systems of N genes
- The N boolean rules are assigned **randomly**
- Each rule has K inputs, where K is fixed
- The dynamics of each network is simulated for **each of the 2^N possible $\vec{x}$ starting states** $\rightarrow$ have to use small N .

The Kauffman Model

In “*The Origins of Order: Self-organization and Selection in Evolution*” Kauffman presents the following study (computer simulation):

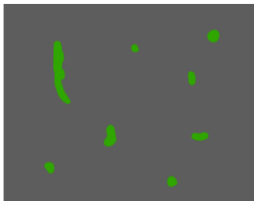
- He looks at different systems of N genes
- The N boolean rules are assigned **randomly**
- Each rule has K inputs, where K is fixed
- The dynamics of each network is simulated for **each of the 2^N possible $\vec{x}$ starting states** $\rightarrow$ have to use small N .

The possible number of gene expression states in human cells:
 $2^{25,000}$

The Kauffman Model

Results form the simulation

- active region of phase space
- frozen region of phase space



Order
 $K=1$



Edge of Chaos
 $K=2$

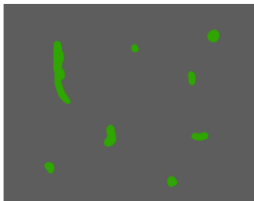


Chaos
 $K>2$

The Kauffman Model

Results from the simulation

- active region of phase space
- frozen region of phase space



Order
 $K=1$



Edge of Chaos
 $K=2$



Chaos
 $K>2$

- For $K = 2$ interesting things happen...
Kauffman calls this “the edge of chaos”, the area between stability and chaos.

The Kauffman Model

Results from the simulation

- active region of phase space
- frozen region of phase space



Order
 $K=1$



Edge of Chaos
 $K=2$



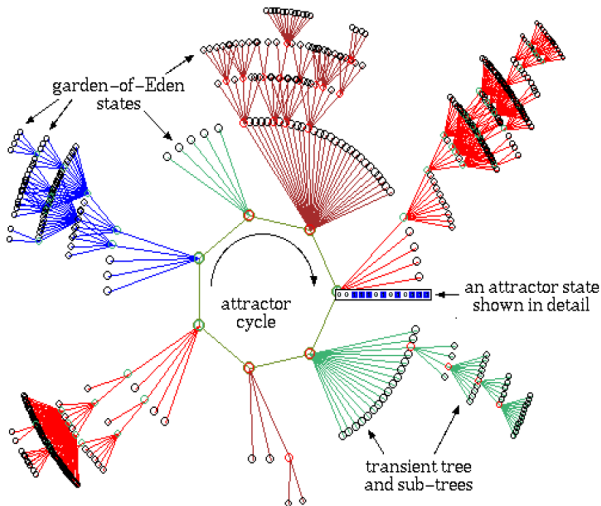
Chaos
 $K>2$

- For $K = 2$ interesting things happen...
Kauffman calls this “the edge of chaos”, the area between stability and chaos.

In real life K can be different for each rule. Recent measurement of $\langle K \rangle$ in yeast gives $\langle K \rangle = 1.9$

The Kauffman Model

Different input states might end up in the same attractor ($K = 2$):



The Kauffman Model

For $K = 2$, Kauffman found that:

- The number of L -cycle attractors $\approx \sqrt{N}$
- Also the length L of the cycles $\approx \sqrt{N}$

The Kauffman Model

For $K = 2$, Kauffman found that:

- The number of L -cycle attractors $\approx \sqrt{N}$
- Also the length L of the cycles $\approx \sqrt{N}$

Plugging in $N = 25,000$ to the fit

- gives: **the mean number of attractors = 159**

The Kauffman Model

For $K = 2$, Kauffman found that:

- The number of L -cycle attractors $\approx \sqrt{N}$
- Also the length L of the cycles $\approx \sqrt{N}$

Plugging in $N = 25,000$ to the fit

- gives: **the mean number of attractors = 159**

Fairly good agreement with the number of cell types in our body
= 256

The Kauffman Model

For $K = 2$, Kauffman found that:

- The number of L -cycle attractors $\approx \sqrt{N}$
- Also the length L of the cycles $\approx \sqrt{N}$

Plugging in $N = 25,000$ to the fit

- gives: **the mean number of attractors = 159**

Fairly good agreement with the number of cell types in our body
= 256

Each attractor corresponds to a cell type!

The Kauffman Model

For $K = 2$, Kauffman found that:

- The number of L -cycle attractors $\approx \sqrt{N}$
- Also the length L of the cycles $\approx \sqrt{N}$

Plugging in $N = 25,000$ to the fit

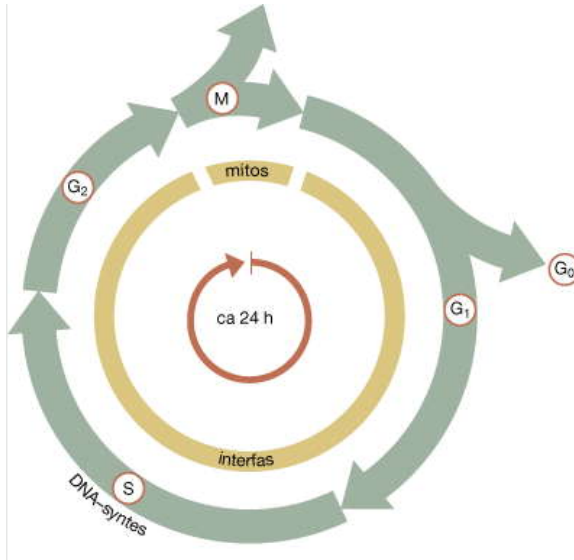
- gives: **the mean number of attractors = 159**

Fairly good agreement with the number of cell types in our body
= 256

Each attractor corresponds to a cell type!

- Also, the expected length of the cycles times the typical time to fold a protein, ~ 5 min, gives **13 hours**, which agrees well with the **cell cycle time** in human cells, which is **6-48 hours**.

The Kauffman Model



Are there flaws in the model?

- More powerful simulations show that the previous results doesn't hold for large N . The dependent does not go as N^α but is exponential.

Are there flaws in the model?

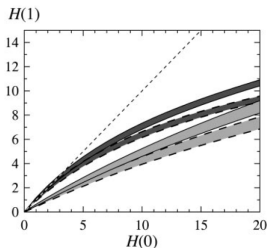
- More powerful simulations show that the previous results doesn't hold for large N . The dependent does not go as N^α but is exponential.
- However, if one excludes certain rules that are not observed in nature from the random rule assignment, one observe a potential dependence again

Are there flaws in the model?

- More powerful simulations show that the previous results doesn't hold for large N . The dependent does not go as N^α but is exponential.
- However, if one excludes certain rules that are not observed in nature from the random rule assignment, one observe a potential dependence again
- A recent paper proves that all networks that only use “canalyzing rules” are stable (independent on K)

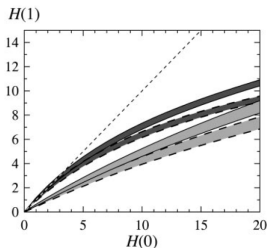
Perturbations to the network

- Real cells are affected by their surrounding — temperature changes, hormones, etc.
- Several studies published that compares perturbations in real cells vs boolean network models.



Perturbations to the network

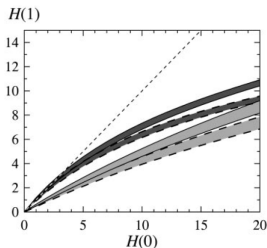
- Real cells are affected by their surrounding — temperature changes, hormones, etc.
- Several studies published that compares perturbations in real cells vs boolean network models.



- A system exactly on “the edge of chaos” will not survive perturbations. We must be on the stable side.

Perturbations to the network

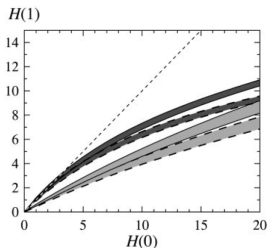
- Real cells are affected by their surrounding — temperature changes, hormones, etc.
- Several studies published that compares perturbations in real cells vs boolean network models.



- A system exactly on “the edge of chaos” will not survive perturbations. We must be on the stable side.
- Evolution of complex organisms favours “canalyzing rules”.

Perturbations to the network

- Real cells are affected by their surrounding — temperature changes, hormones, etc.
- Several studies published that compares perturbations in real cells vs boolean network models.



- A system exactly on “the edge of chaos” will not survive perturbations. We must be on the stable side.
- Evolution of complex organisms favours “canalyzing rules”.
- A recent paper proves that all networks that only use “canalyzing rules” always are stable (independent of K).

Refining the model

- A nice feature of the model is that one easily can produce measurable quantities.

Refining the model

- A nice feature of the model is that one easily can produce measurable quantities.
- Fits with data using the Kauffman's original model are usually quite poor, but are greatly improved by **refining the model**:

Refining the model

- A nice feature of the model is that one easily can produce measurable quantities.
 - Fits with data using the Kauffman's original model are usually quite poor, but are greatly improved by **refining the model**:
- 1 One can let K be different for each rule
(choose K from a suitable random distribution)

Refining the model

- A nice feature of the model is that one easily can produce measurable quantities.
 - Fits with data using the Kauffman's original model are usually quite poor, but are greatly improved by **refining the model**:
- 1 One can let K be different for each rule
(choose K from a suitable random distribution)
 - 2 Restrict the available rules

Refining the model

- A nice feature of the model is that one easily can produce measurable quantities.
- Fits with data using the Kauffman's original model are usually quite poor, but are greatly improved by **refining the model**:
 - 1 One can let K be different for each rule
(choose K from a suitable random distribution)
 - 2 Restrict the available rules
 - 3 Assign weights to each gene in the f_i function

Refining the model

- A nice feature of the model is that one easily can produce measurable quantities.
- Fits with data using the Kauffman's original model are usually quite poor, but are greatly improved by **refining the model**:
 - 1 One can let K be different for each rule (choose K from a suitable random distribution)
 - 2 Restrict the available rules
 - 3 Assign weights to each gene in the f_i function
 - 4 Assign different expression times for each gene (e.g. between 1–10 min) and iterate over small time steps.

Summary

- Biological Networks tend to organise themselves into cycles

- Biological Networks tend to organise themselves into cycles
- Based on a few simple but reasonable assumptions, one can model these systems as **Random Boolean Networks**

- Biological Networks tend to organise themselves into cycles
- Based on a few simple but reasonable assumptions, one can model these systems as **Random Boolean Networks**
- Close to “the edge of chaos” these networks agree well with nature

- Biological Networks tend to organise themselves into cycles
- Based on a few simple but reasonable assumptions, one can model these systems as **Random Boolean Networks**
- Close to “the edge of chaos” these networks agree well with nature
- Possible to make the model more realistic (and complex) which gives very good agreement with data

- Biological Networks tend to organise themselves into cycles
- Based on a few simple but reasonable assumptions, one can model these systems as **Random Boolean Networks**
- Close to “the edge of chaos” these networks agree well with nature
- Possible to make the model more realistic (and complex) which gives very good agreement with data
- Lot's of ongoing research on these networks

