

The role of computation in Complex Regulatory Networks

Pau Fernàndez Ricard V. Solé

Summary

A model for studying complex regulatory networks

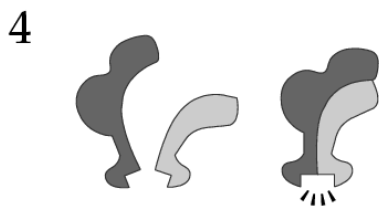
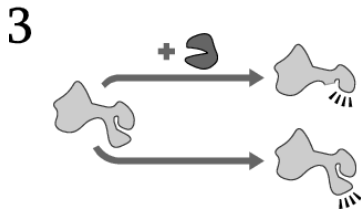
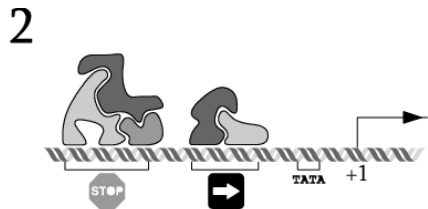
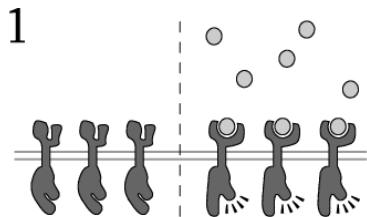
- Information plays an important role in biological systems (transmission, storage and manipulation)
- They are able to perform non-linear computations
- Unpredictability of the outcomes

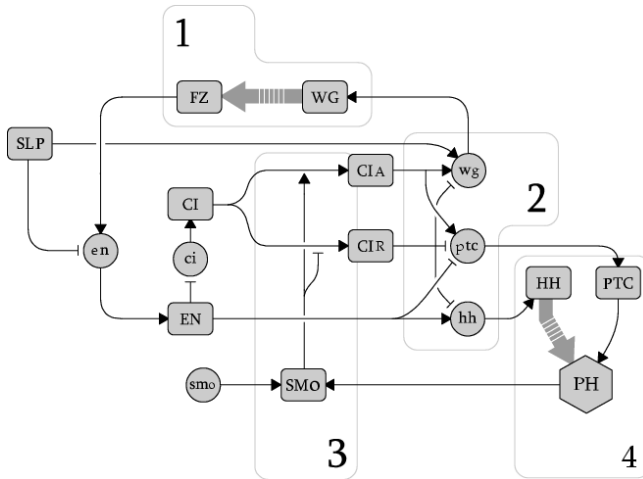
Useful to understand:

- 1 basic properties of general systems
- 2 how these properties could be achieved by an evolutionary point of view

Regulatory Networks

- Concern biological systems (such as cells)
- Units of regulations: genes, proteins and molecules
- Values: either concentration (proteins) or levels of activation (genes)
- Interactions: gene transcription, post-transcription modifications, signaling, protein complexes





Regulatory networks can achieve enormous complexity in complex organism. System can be described by a set of equations, but the number of parameters is huge.

- topology of the network is more important than the exact mechanism at each node
- knowing whether an interaction is activating or inhibiting is enough to reproduce the main dynamics

Regulatory networks can achieve enormous complexity in complex organism. System can be described by a set of equations, but the number of parameters is huge.

- topology of the network is more important than the exact mechanism at each node
- knowing whether an interaction is activating or inhibiting is enough to reproduce the main dynamics

Computational Irreducibility

Find a method (shortcut) that is able to reproduce a final state of a system by knowing the initial state without having to meticulously trace out each step from the initial to final states. In some cases, no predictive ability is possible. This impossibility to predict is called **irreducibility**.

Regulatory networks are supposed to be irreducible since:

- proteins and molecules can perform logical or computational tasks
- computer simulations seems to be the only way to deal with their complexity

Boolean Networks

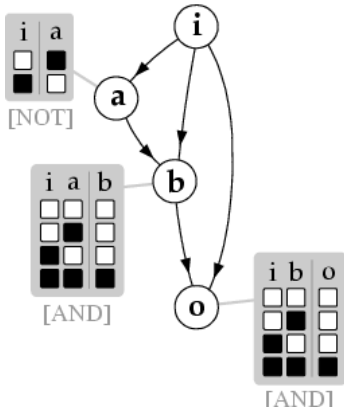
Boolean Networks can manage the complexity and are sufficient to capture the essential features because high-level behavior results from low-level unit interaction.

Simplifications:

- low-level unit details are neglected
- no thorough understanding of all the processes in the cells

The idea

- See the network as devices performing a definite task
- Units have a small amount of memory which defines the state
- States of units are boolean (i.e. on-off)
- Unit performs boolean functions
- The state of a unit is a function of the state of the units it interacts with (such as cellular automata)
- Synchronous discrete time steps



Boolean network recognizing the sequence "011". Each unit compute its next value using a table with all possible input-output states.

◀ Back

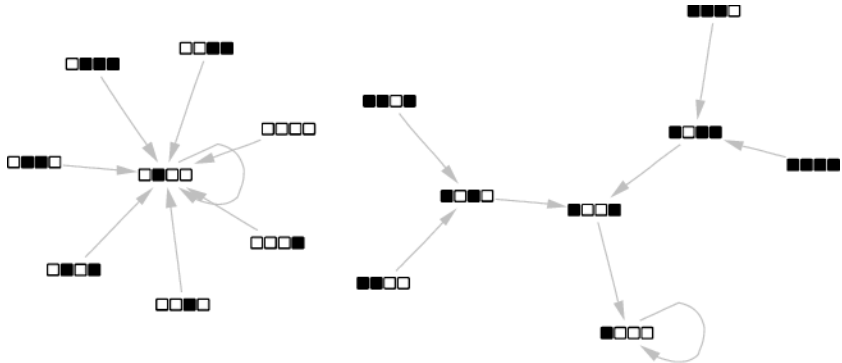
Kauffman reverse approach

Study Boolean networks wired at random with the hope of finding properties that apply to the Regulatory networks.

- Network composed of N genes g_i
- The average connectivity is K
- Interaction by means of boolean functions f_i

$$g_i^{t+1} = f_i(g_{j_1}^t, \dots, g_{j_K}^t)$$

- The K input nodes are chosen at random
- Functions are chosen so that outputs have a 1 with probability p and 0 with probability $1 - p$
- System has a global state $S \equiv (g_1, g_2, \dots, g_N)$



Different subsets of states that end in the same loop are called **basins of attraction**. Correlation between the cell types and the basins.

Propagation of errors

A change in g_i will arrive at all his $\langle K \rangle$ neighbors.
Since propagation probability is $P = 2p(1 - p)$, the average number of changed neighbors is

$$N_{ch} = P\langle K \rangle = 2p(1 - p)\langle K \rangle$$

Depending on N_{ch} there may be three phases:

- if $N_{ch} < 1$ all errors will die out, ordered phase
- if $N_{ch} > 1$ all errors will propagate, chaotic phase
- if $N_{ch} = 1$ errors will neither die out nor propagate, critical phase

Propagation of errors

A change in g_i will arrive at all his $\langle K \rangle$ neighbors.
Since propagation probability is $P = 2p(1 - p)$, the average number of changed neighbors is

$$N_{ch} = P\langle K \rangle = 2p(1 - p)\langle K \rangle$$

Depending on N_{ch} there may be three phases:

- if $N_{ch} < 1$ all errors will die out, **ordered phase**
- if $N_{ch} > 1$ all errors will propagate, chaotic phase
- if $N_{ch} = 1$ errors will neither die out nor propagate, critical phase

Propagation of errors

A change in g_i will arrive at all his $\langle K \rangle$ neighbors.
Since propagation probability is $P = 2p(1 - p)$, the average number of changed neighbors is

$$N_{ch} = P\langle K \rangle = 2p(1 - p)\langle K \rangle$$

Depending on N_{ch} there may be three phases:

- if $N_{ch} < 1$ all errors will die out, ordered phase
- if $N_{ch} > 1$ all errors will propagate, **chaotich phase**
- if $N_{ch} = 1$ errors will neither die out nor propagate, critical phase

Propagation of errors

A change in g_i will arrive at all his $\langle K \rangle$ neighbors.
Since propagation probability is $P = 2p(1 - p)$, the average number of changed neighbors is

$$N_{ch} = P\langle K \rangle = 2p(1 - p)\langle K \rangle$$

Depending on N_{ch} there may be three phases:

- if $N_{ch} < 1$ all errors will die out, ordered phase
- if $N_{ch} > 1$ all errors will propagate, chaotic phase
- if $N_{ch} = 1$ errors will neither die out nor propagate, **critical phase**

The connectivity degree $\langle K \rangle$ and p alone determine the overall behavior of the system. The **critical connectivity** is

$$K_c = \frac{1}{2p(1-p)}$$

Desired properties of Boolean networks that stay in the critical phase:

- the low connectivity, i.e. the network is sparse just like real networks are ($p = 0.5 \Rightarrow K_c = 2$)
- irreducibility of systems in this phase is the main characteristic of system capable of computation
- the emergence of order "for free"

The connectivity degree $\langle K \rangle$ and p alone determine the overall behavior of the system. The **critical connectivity** is

$$K_c = \frac{1}{2p(1-p)}$$

Desired properties of Boolean networks that stay in the critical phase:

- the **low connectivity**, i.e. the network is sparse just like real networks are ($p = 0.5 \Rightarrow K_c = 2$)
- irreducibility of systems in this phase is the main characteristic of system capable of computation
- the emergence of order "for free"

The connectivity degree $\langle K \rangle$ and p alone determine the overall behavior of the system. The **critical connectivity** is

$$K_c = \frac{1}{2p(1-p)}$$

Desired properties of Boolean networks that stay in the critical phase:

- the low connectivity, i.e. the network is sparse just like real networks are ($p = 0.5 \Rightarrow K_c = 2$)
- **irreducibility** of systems in this phase is the main characteristic of system capable of computation
- the emergence of order "for free"

The connectivity degree $\langle K \rangle$ and p alone determine the overall behavior of the system. The **critical connectivity** is

$$K_c = \frac{1}{2p(1-p)}$$

Desired properties of Boolean networks that stay in the critical phase:

- the low connectivity, i.e. the network is sparse just like real networks are ($p = 0.5 \Rightarrow K_c = 2$)
- irreducibility of systems in this phase is the main characteristic of system capable of computation
- the emergence of **order "for free"**

Functional aspects

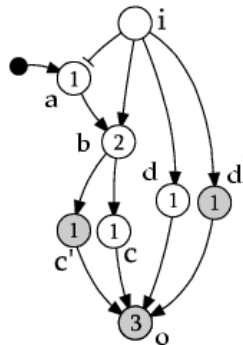
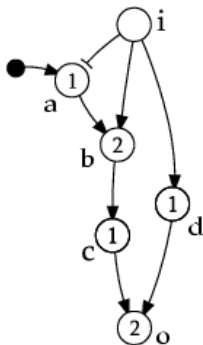
Boolean logic is unable to tolerate the failure of single components if the circuit represents the minimal implementation.

However, natural systems have a great deal of robustness against:

- thermal noise introduces stochasticity
- mutations: cells accumulate changes in the genome through time

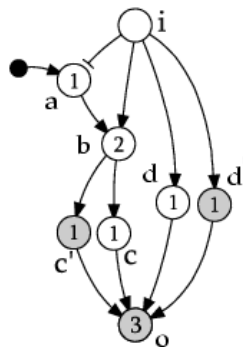
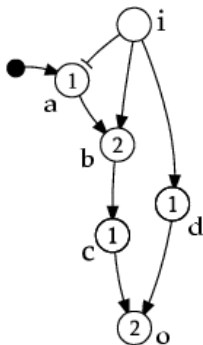
▶ Example

Redundancy



Simple form of robustness would make organisms far less able to innovate, making adaptation very awkward.

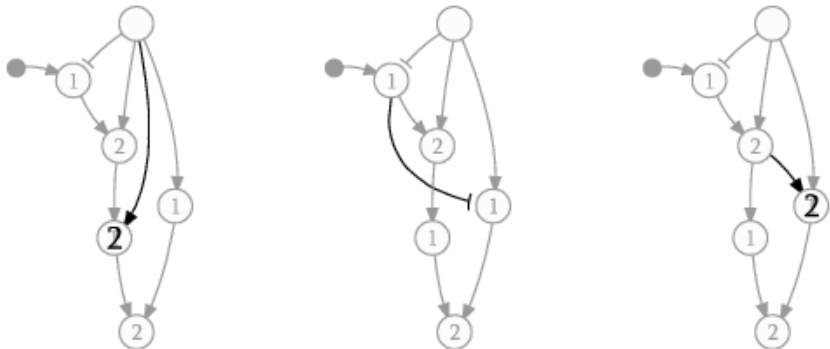
Redundancy



Simple form of robustness would make organisms far less able to innovate, making adaptation very awkward.

Degeneracy

The ability of elements that are structurally different to perform the same function (by means of neutral mutations).



Evolvability

The capacity to generate heritable, selectable, phenotypic variations.

- 1 How to reduce the lethality of mutations? Robustness
- 2 How to reduce the number of mutations needed to produce phenotypic variations?

Small number of variations in circuits can produce different shapes

Evolvability

The capacity to generate heritable, selectable, phenotypic variations.

- 1 How to reduce the lethality of mutations? **Robustness**
- 2 How to reduce the number of mutations needed to produce phenotypic variations?

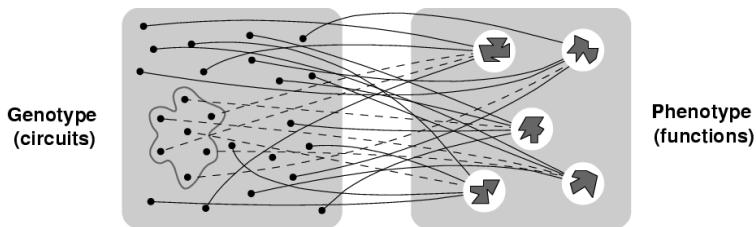
Small number of variations in circuits can produce different shapes

Evolvability

The capacity to generate heritable, selectable, phenotypic variations.

- 1 How to reduce the lethality of mutations? Robustness
- 2 How to reduce the number of mutations needed to produce phenotypic variations?

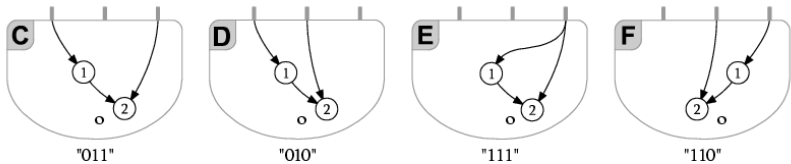
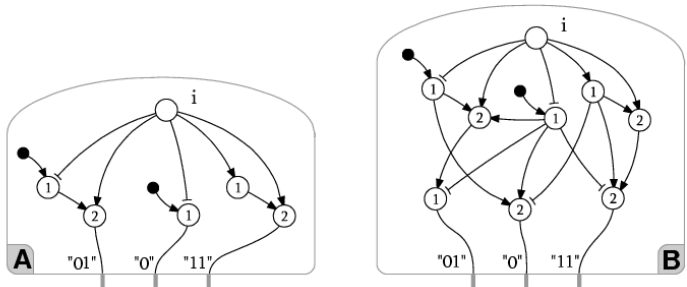
Small number of variations in circuits can produce different shapes



Modularity: fast evolvability

Groups of genes act as constant modules which can recombine against others creating new subnetworks with high-level computational features.

- Subnetworks found to be useful modules may be made more robust (degeneracy)
- Mutations will involve more likely how these modules are combined
- Topological point of view: modules are clusters of densely connected nodes
- Real networks have a higher clustering coefficient
- Real networks display a scale-free distribution: strongly connected modules assembled into less cohesive bigger modules.



Conclusions

- Very little is known about how complex regulatory networks are organized
- This model can give important answers to questions regarding global average dynamics
- The reasons for the success of this approximation might be found in the irreducibility of cellular processes