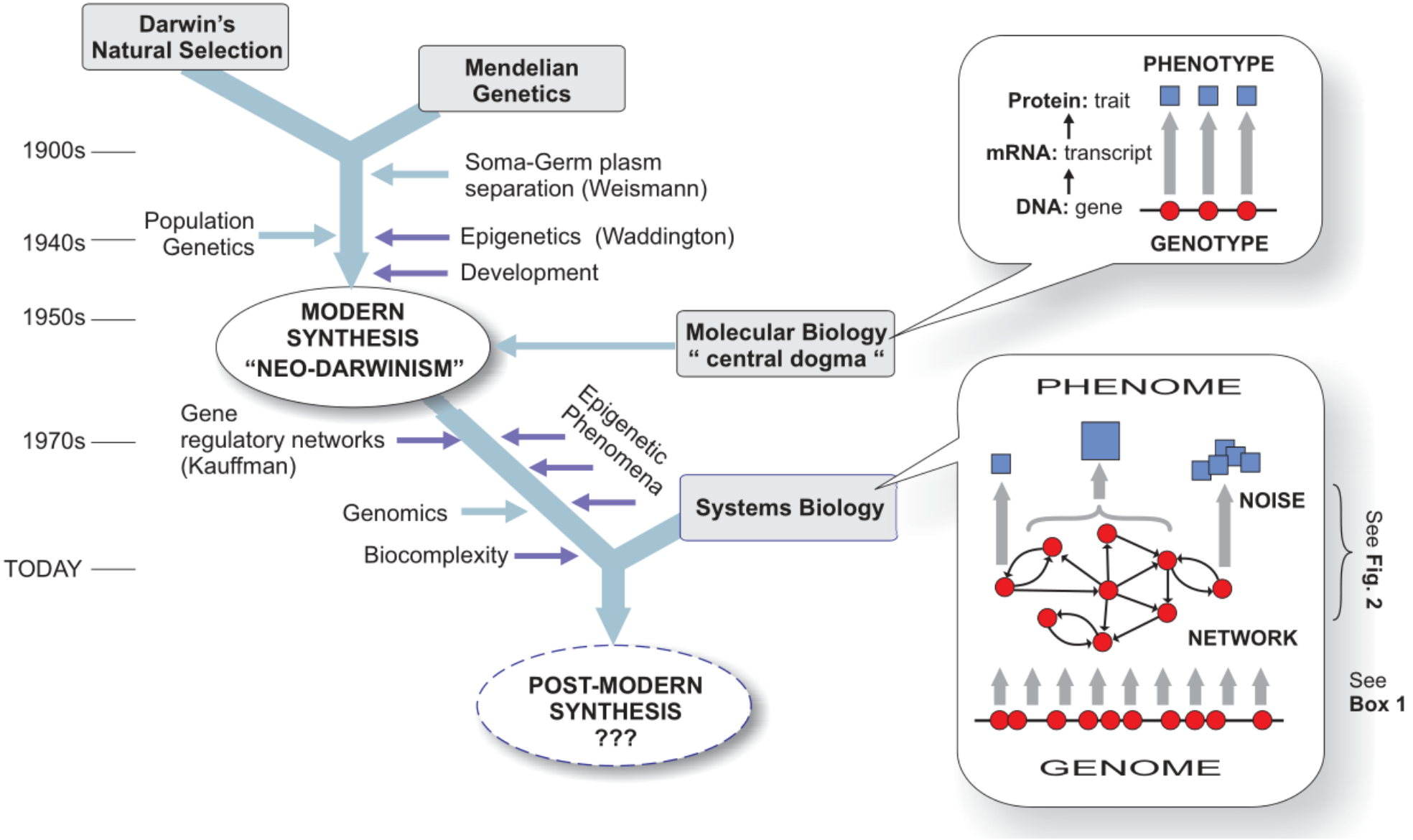
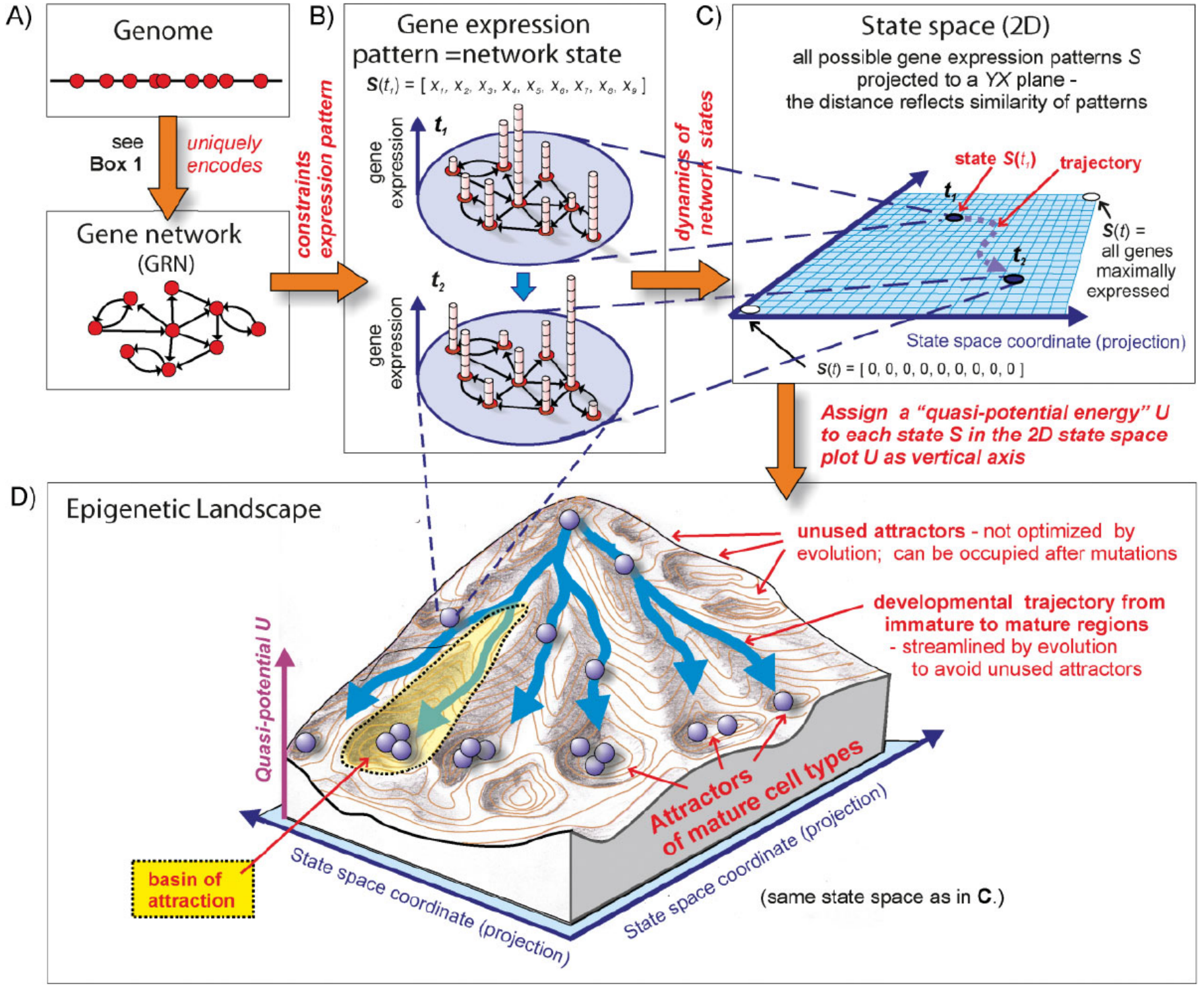
Redes de Regulación Génica

Redes de Regulación Génica

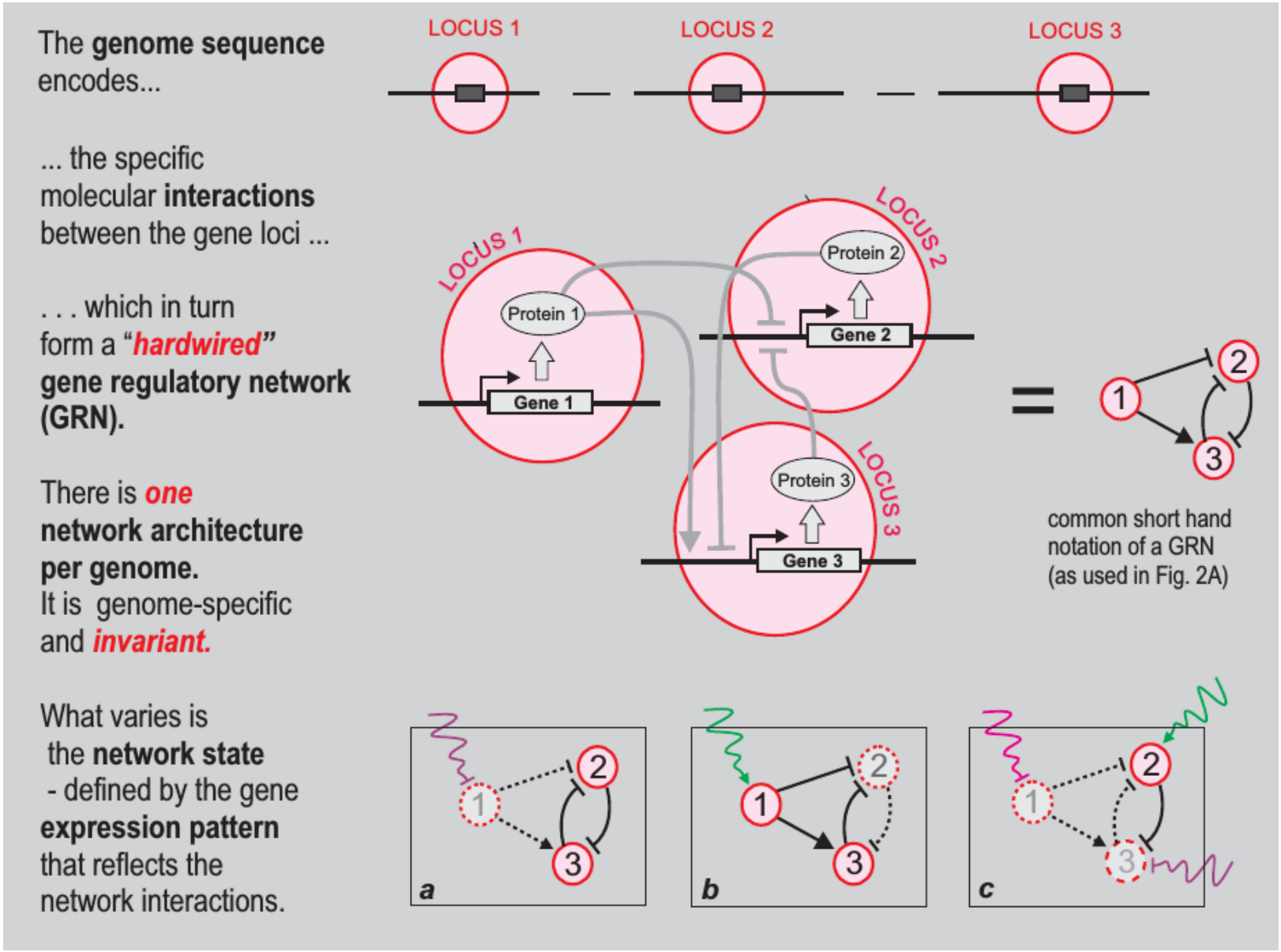


Huang, S. (2012). The molecular and mathematical basis of Waddington's epigenetic landscape: A framework for post-Darwinian biology?. Bioessays, 34(2), 149-157.



Huang, S. (2012). The molecular and mathematical basis of Waddington's epigenetic landscape: A framework for post-Darwinian biology?. Bioessays, 34(2), 149-157.

Redes de Regulación Génica



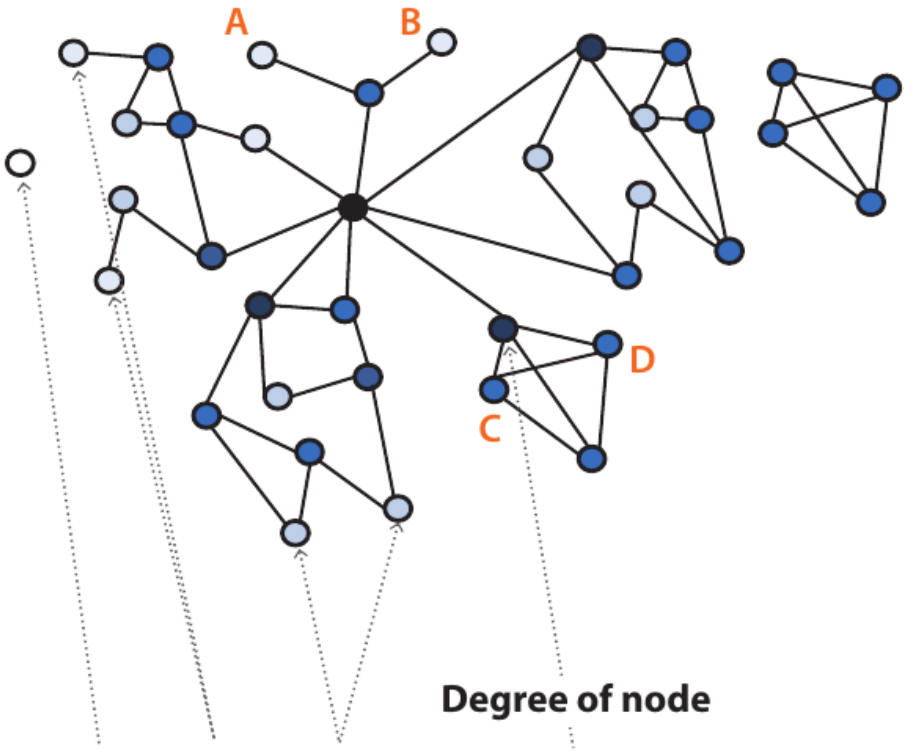
Huang, S. (2012). The molecular and mathematical basis of Waddington's epigenetic landscape: A framework for post-Darwinian biology?. Bioessays, 34(2), 149-157.

Redes de Regulación Génica

Grafos → Topología

Autómatas → Dinámica

Teoría de Grafos

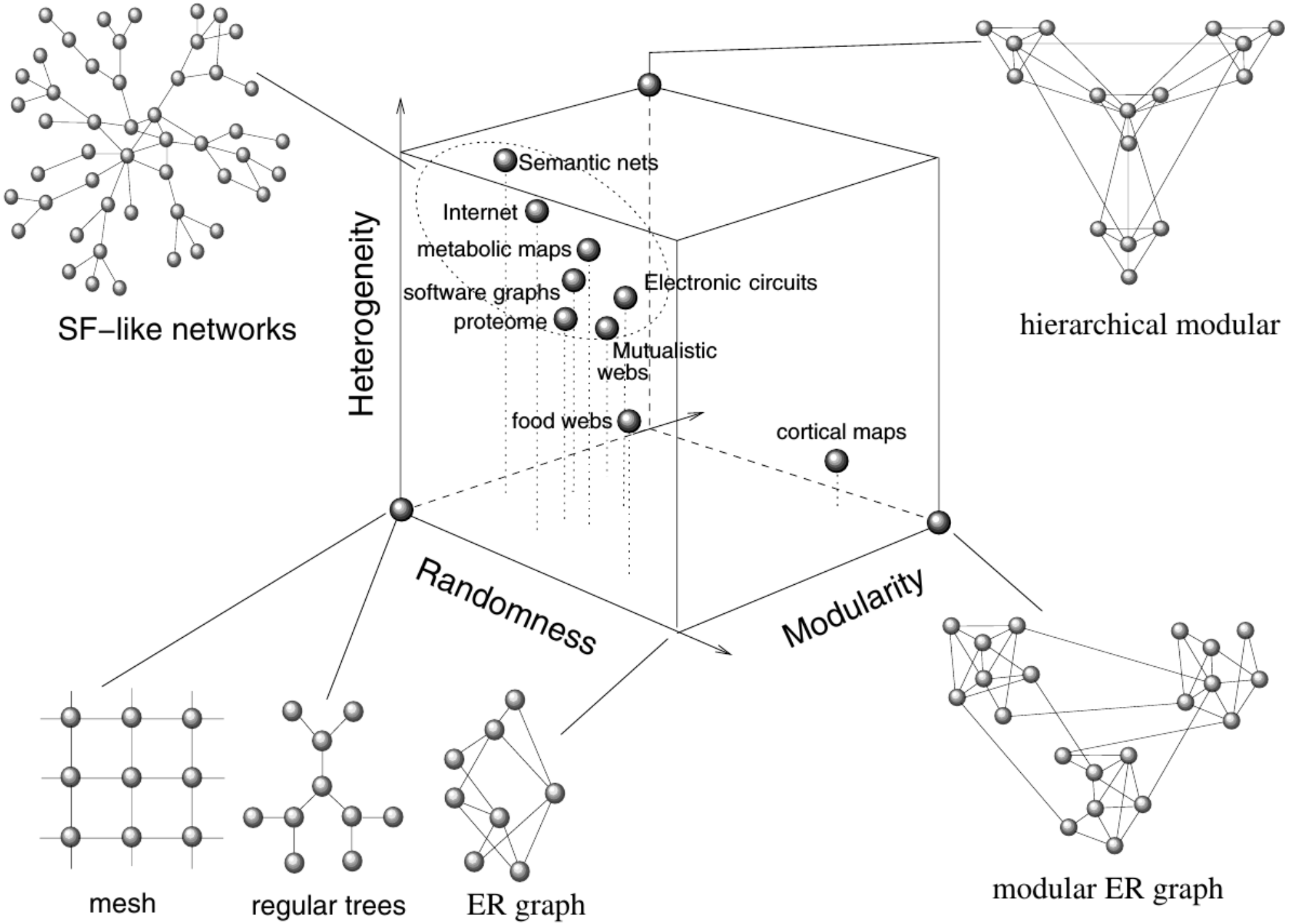


Degree of node	0	1	2	3	4	5	6	7	8
Number of nodes	1	4	10	17	4	0	0	0	1
Frequency (%)	3	11	27	46	11	0	0	0	3

Figure 3

Network descriptors. A small, three component network is shown along with its degree distribution. Node shading indicates degree, from lightest (degree 0) to darkest (degree 8). Dotted arrows indicate typical nodes with particular degrees. The node with degree 8 links together much of the network. Nodes A and B are not clustered; nodes C and D are.

Shaw, M. W. (2014). Networks and plant disease management: Concepts and applications. Annual review of phytopathology, 52, 477-493.



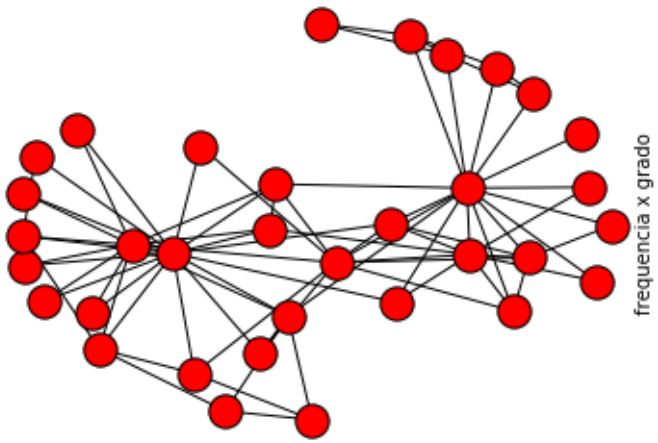
Solé, R. V., & Valverde, S. (2004). Information theory of complex networks: on evolution and architectural constraints. In Complex networks (pp. 189-207). Springer Berlin Heidelberg.

Teoría de Grafos

```
import networkx as nx
import matplotlib.pyplot as plt
```

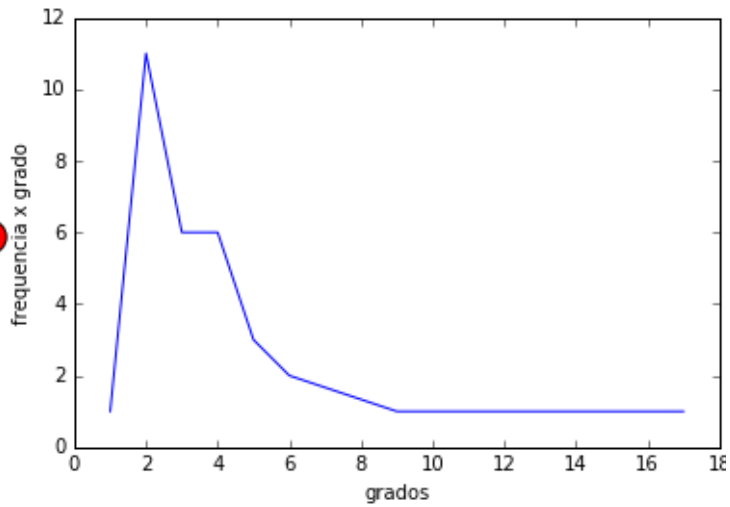
```
kG = nx.karate_club_graph()
```

```
nx.draw(kG)
#plt.show()
```



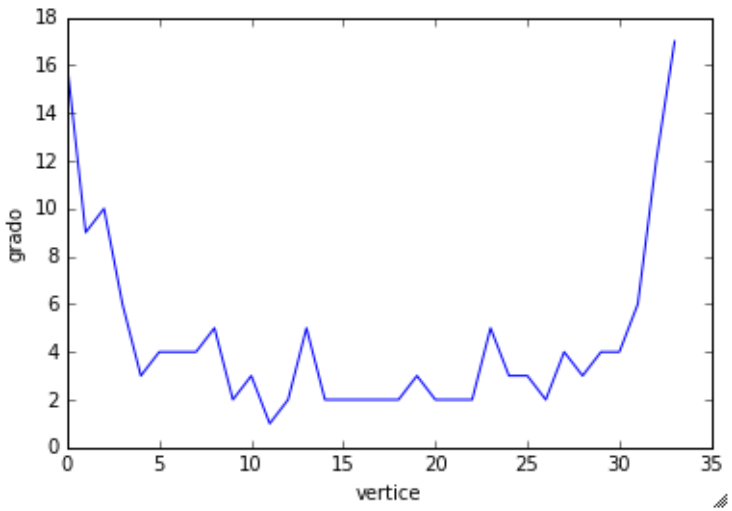
```
#DISTRIBUCION DE GRADOS por PLOTEO
kG_grados = nx.degree(kG)
kGg = sorted(set(kG_grados.values()))
kG_freq= [kG_grados.values().count(g) for g in kGg]
plt.plot(kGg, kG_freq)
plt.xlabel('grados')
plt.ylabel('frequencia x grado')
```

```
<matplotlib.text.Text at 0x7f3b314d2210>
```




```
#VISUALIZACION DE { eje X: elementos VERTICE
#plt.plot(sorted(D.values(), reverse=True))
plt.plot(D.values())
plt.xlabel('vertice')
plt.ylabel('grado')
```

```
<matplotlib.text.Text at 0x7f3b30d205d0>
```



Teoría de Grafos

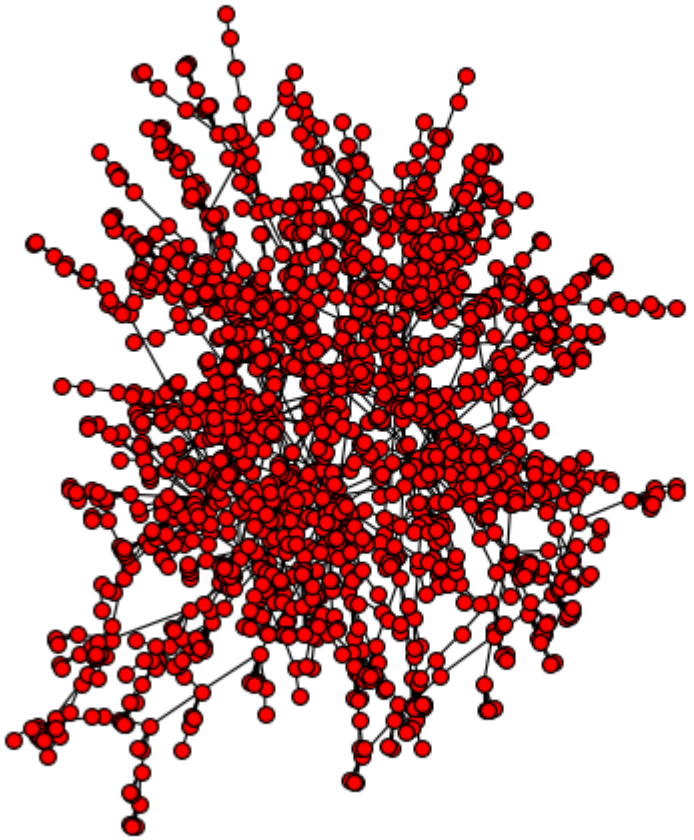
Open Save

```
# Copies and Copyright-Notice
#
#      RegulonDB is free for academic/noncommercial use
#
#      User is not entitled to change or erase data sets of the RegulonDB
#      database or to eliminate copyright notices from RegulonDB. Furthermore,
#      User is not entitled to expand RegulonDB or to integrate RegulonDB partly
#      or as a whole into other databank systems, without prior written consent
#      from CCG-UNAM.
#
#      Please check the license at http://regulondb.ccg.unam.mx/menu/download/full_version/terms_and_conditions.jsp
#
# Citation
#
#      Salgado, H. et al. (2013). "RegulonDB v8.0: Omics data sets, evolutionary conservation,
#      regulatory phrases, cross-validated gold standards and more".
#      Nucleic Acids Research. 2013 Jan 1;41(D1):D203-D213. Epub 2012 Nov 29.
#
#
# Contact
#
#      Person: RegulonDB Team
#      Web Page: http://regulondb.ccg.unam.mx/menu/about_regulondb/contact_us/index.jsp
#          (regulondb@ccg.unam.mx)
#
#
# Release: 9.2 Date: 09-08-2016
# ------------------------------------------------------------------------
#
# Columns:
#  (1) Transcription Factor (TF) name
#  (2) Gene regulated by the TF (regulated gene)
#  (3) Regulatory effect of the TF on the regulated gene (+ activator, - repressor, +- dual, ? unknown)
#  (4) Evidence that supports the existence of the regulatory interaction
#
AccB    accB    -       []      null
AccB    accC    -       []      null
AcrR    acrA    -       [BCE, BPP, GEA, HIBSCS] Weak
AcrR    acrB    -       [BCE, BPP, GEA, HIBSCS] Weak
AcrR    acrR    -       [AIBSCS, BCE, BPP, GEA, HIBSCS] Weak
```

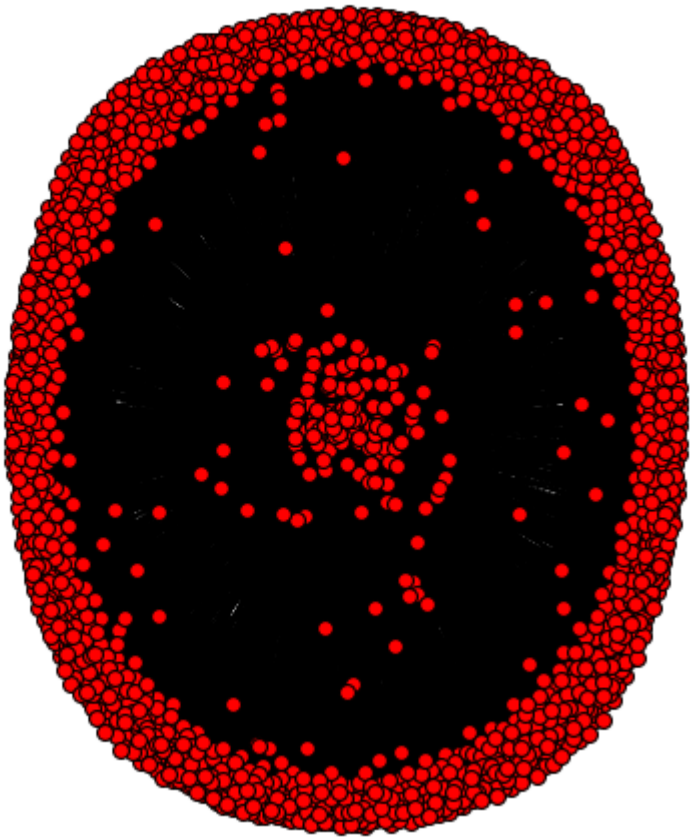
Plain Text Tab Width: 8 Ln 1, Col 1 INS

Teoría de Grafos

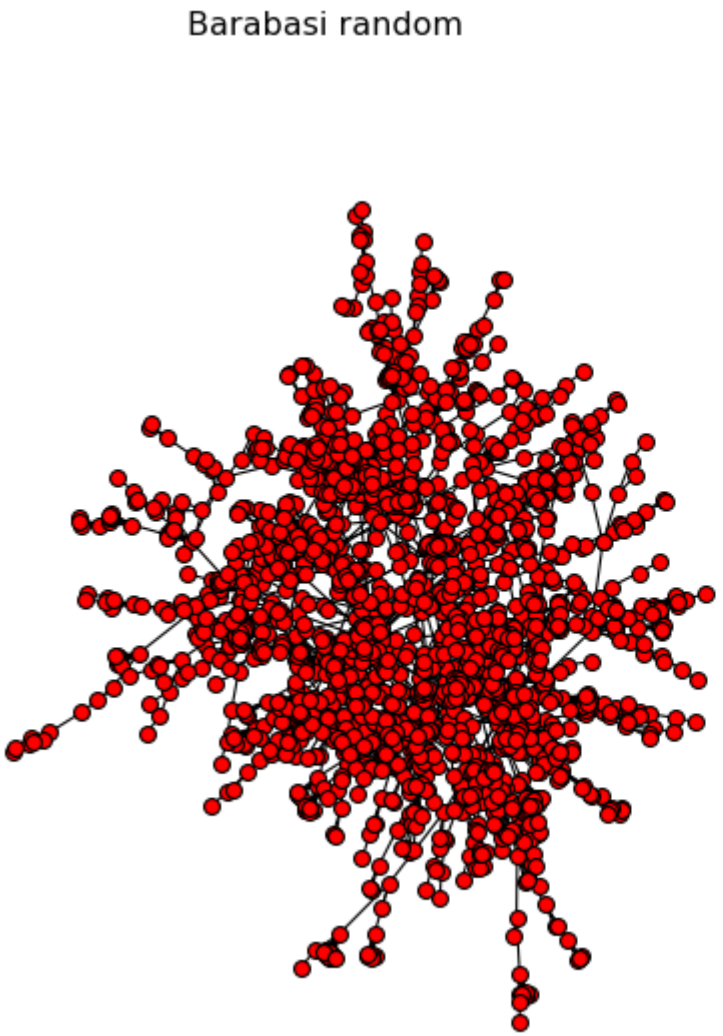
Barabasi random

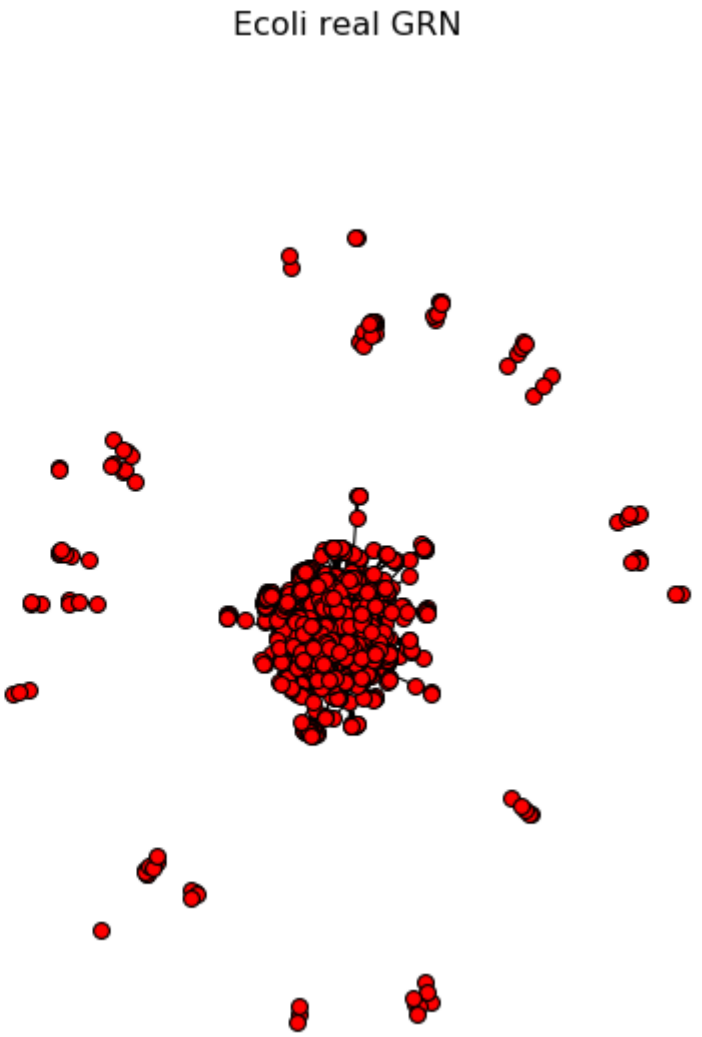


Ecoli real GRN

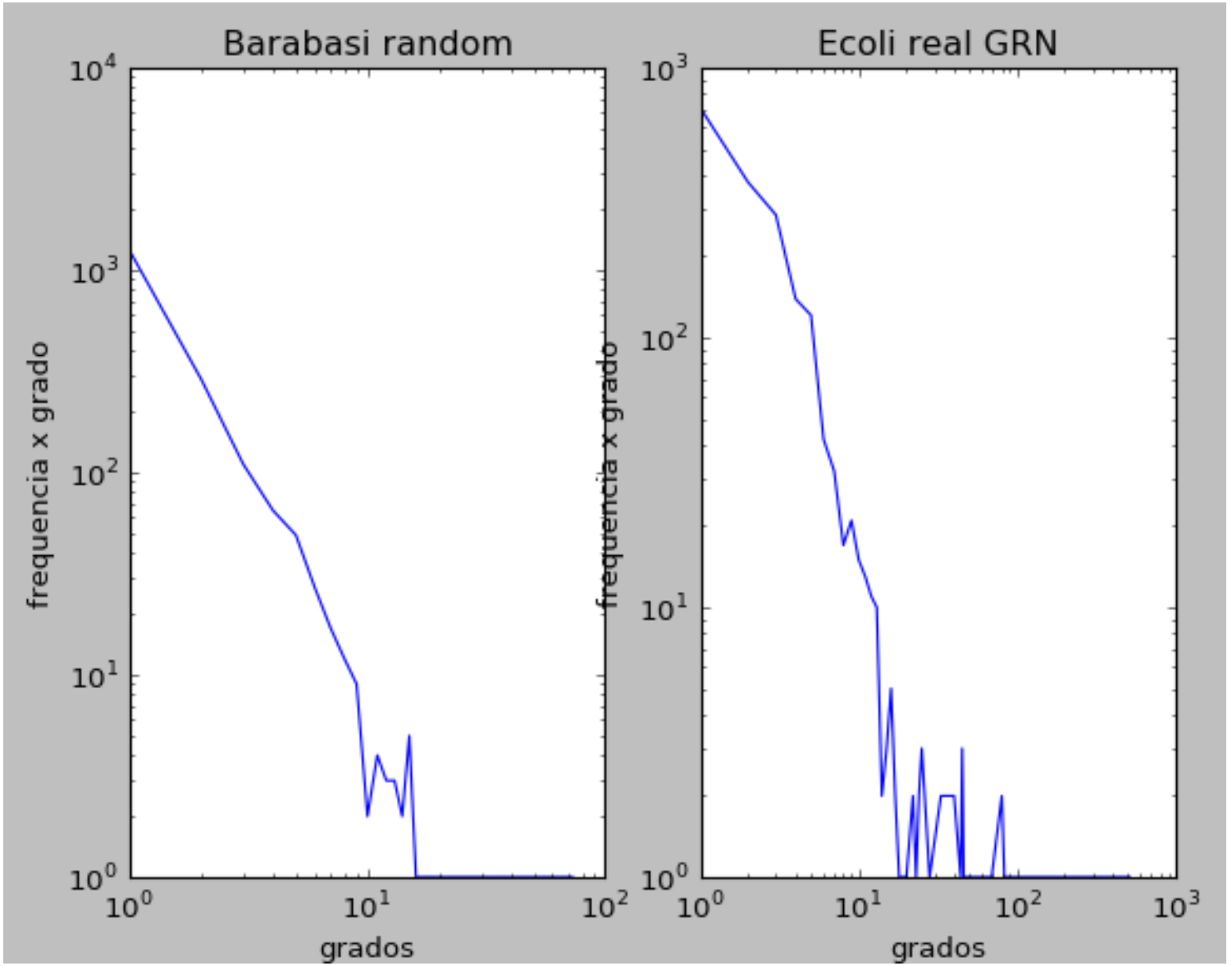


Teoría de Grafos

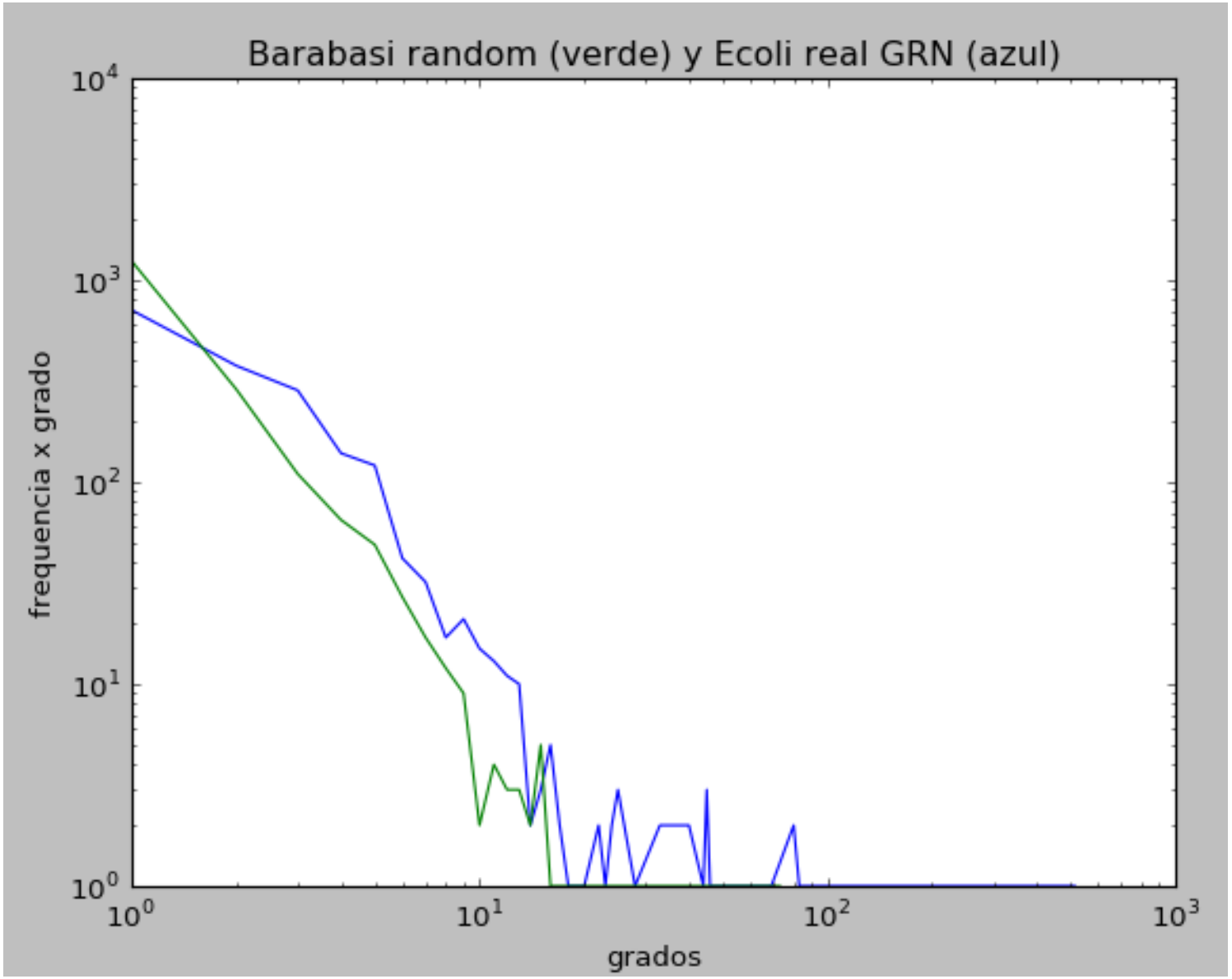




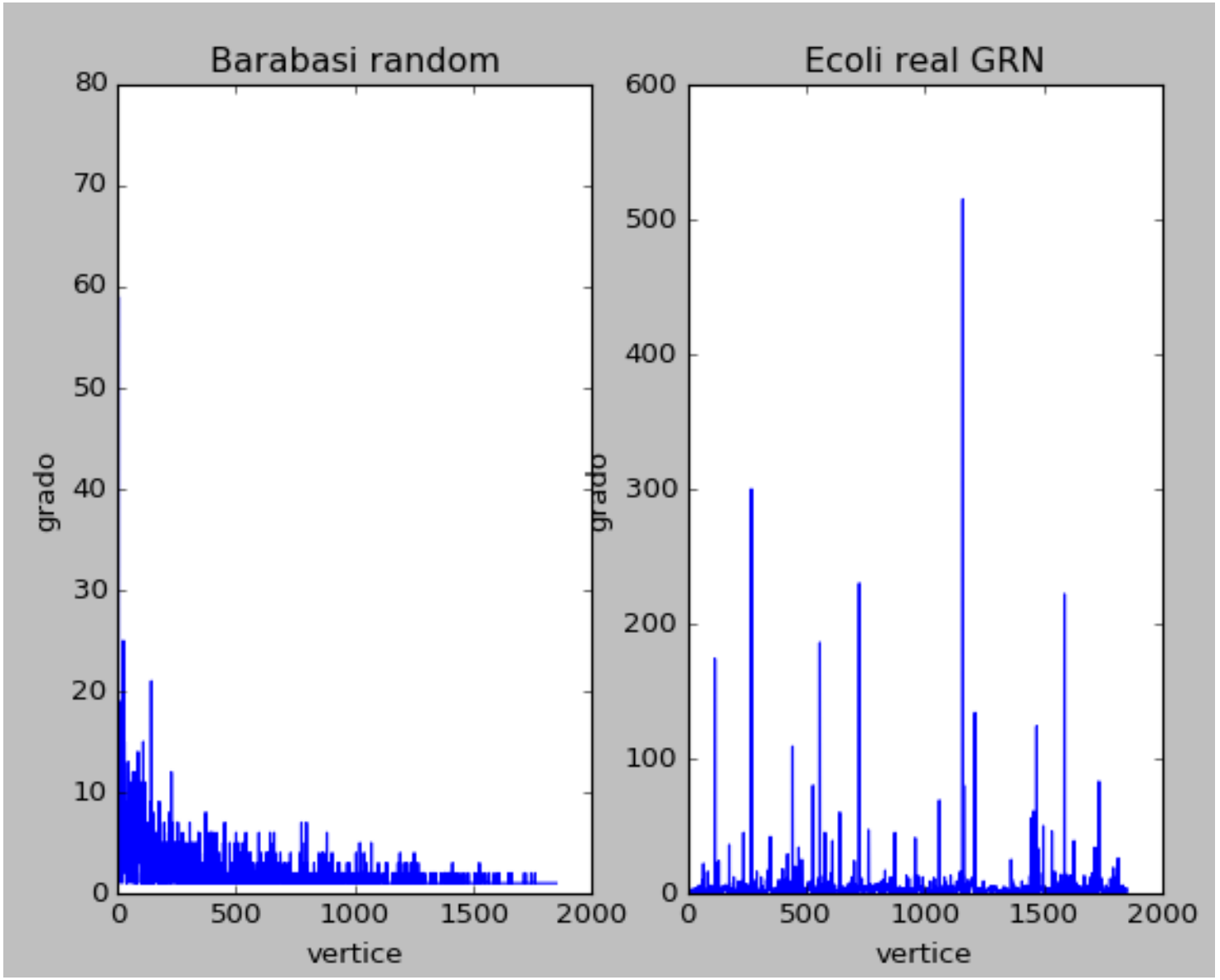
Teoría de Grafos



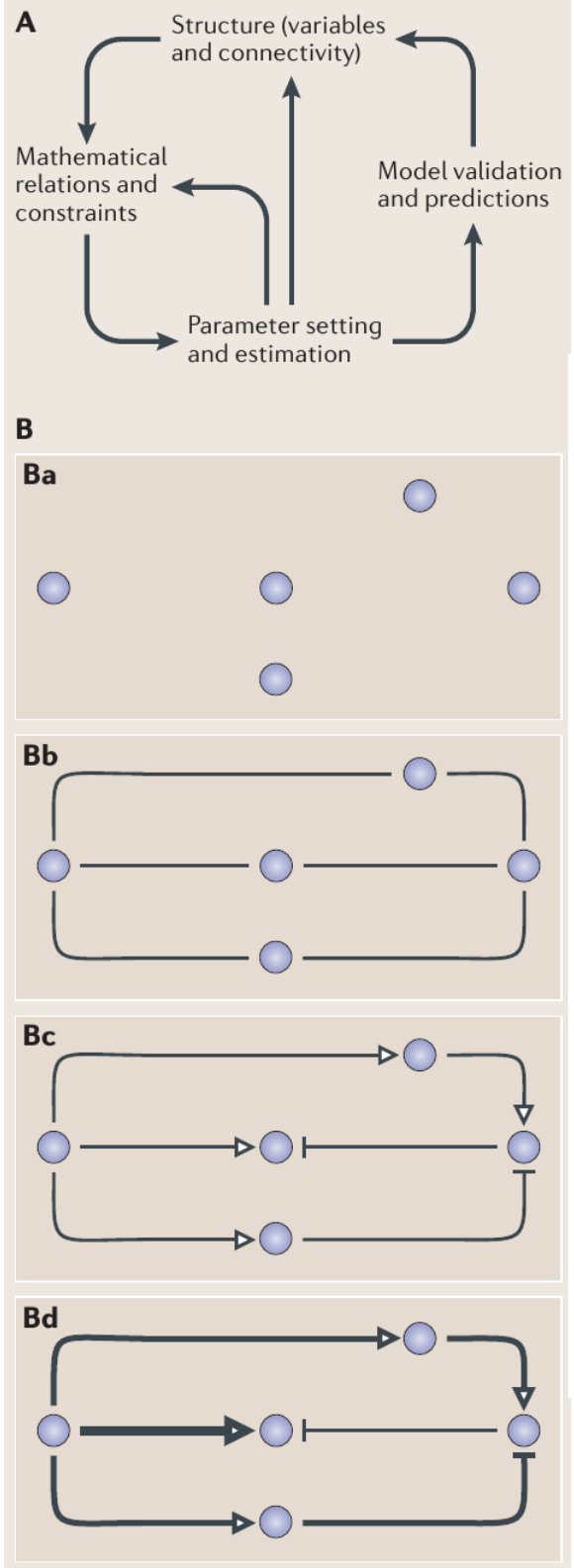
Teoría de Grafos

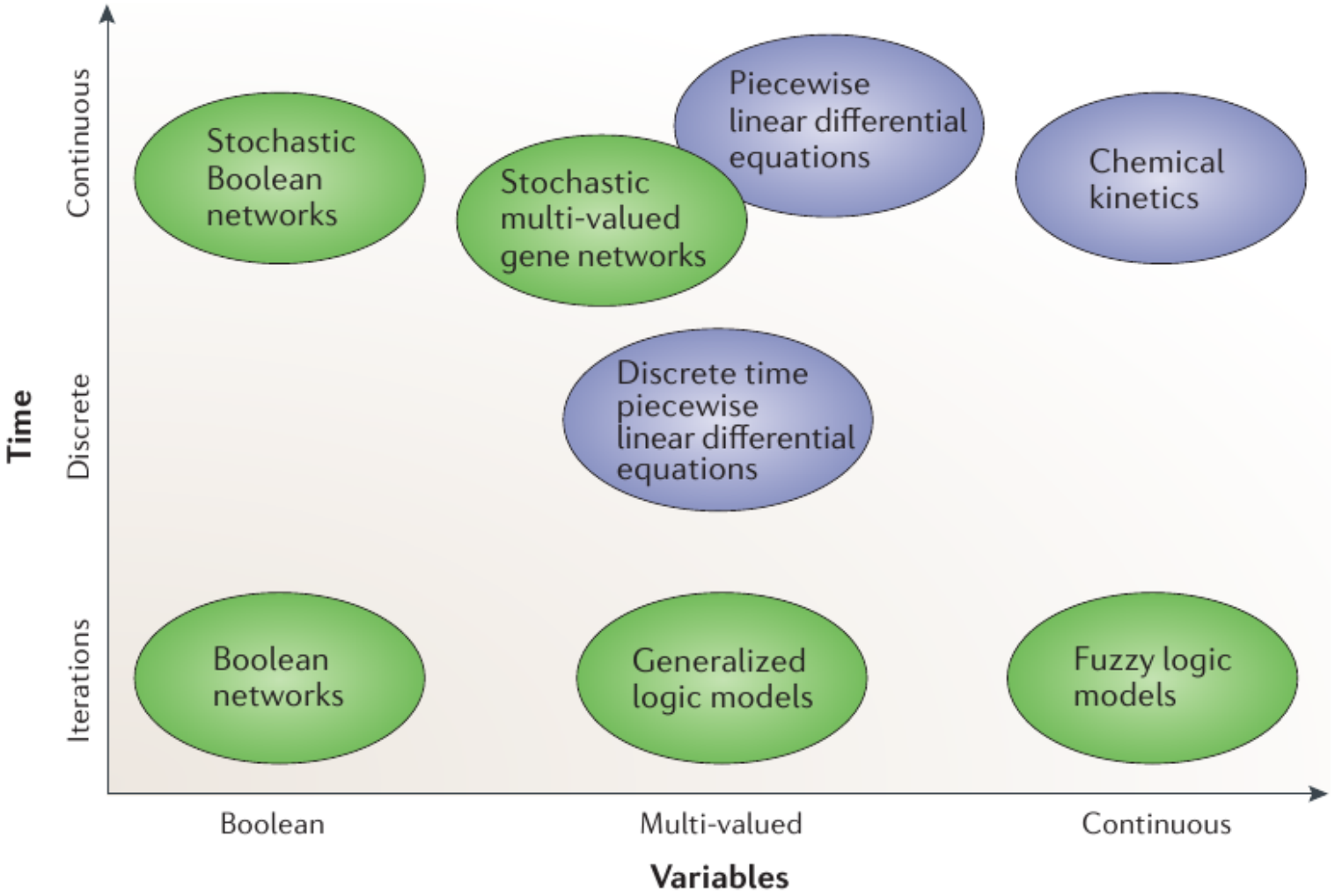


Teoría de Grafos



Dinámica → Enfoques





Le Novère, N. (2015). Quantitative and logic modelling of molecular and gene networks. Nature Reviews Genetics, 16(3), 146-158.

Redes de Regulación Génica

Table 1 | **Comparison of quantitative and logic models**

	Quantitative model	Logic model
Suitable for	Time series	Phenotypes
Time representation	Linear representation	Abstract iterations
Variables	Quantitative	Qualitative
Mechanism representation?	Yes	No
What can we do?	Compute concentrations and durations; evaluate the effect of parameter values	Compute state transitions and attractors (steady states and cyclic attractors)
Data necessary to build the model	Molecular species, genes, interactions and biochemical processes	Activities, defined phenotypes and rules linking those
Data to parameterize and validate the model	Amount of molecular species, time courses and quantitative phenotype	Perturbations of activities such as RNA interference, inhibitors, qualitative phenotypes
Advantages	Quantitative, precise; direct comparison with quantitative measurements; large existing toolkit	Easy to build; easy to compose; easy simulation of perturbations
Weaknesses	Requires quantitative knowledge of initial conditions and kinetics	Cannot provide quantitative predictions; difficult to choose between alternative behaviours

Le Novère, N. (2015). Quantitative and logic modelling of molecular and gene networks. Nature Reviews Genetics, 16(3), 146-158.

Redes de Regulación Génica

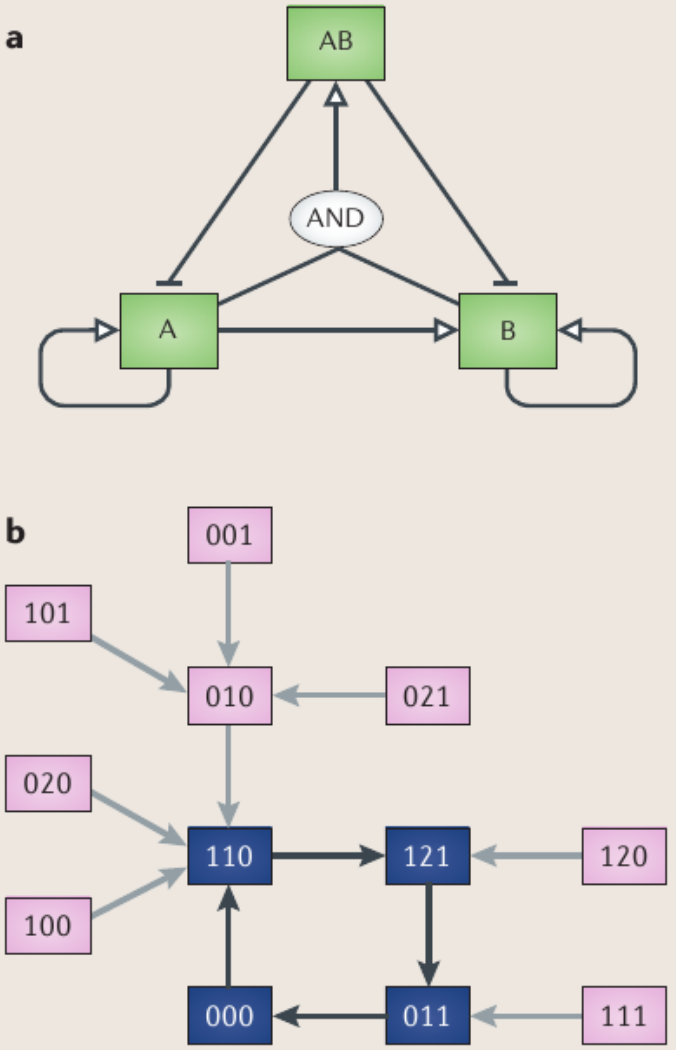
Box 3 | **Logic modelling**

Logic modelling is based on the idea that a variable can take a discrete number of states or values (two in the case of Boolean models) and that the state of a variable is decided by a logical combination of the states of other variables. The system can be updated synchronously, with the values of all variables being calculated after a transition, or asynchronously, when variables undergo transitions one at a time[114].

We can create a logic version of the system presented in BOX 2 by building a model with three nodes representing protein A, protein B and the complex AB (see the figure, part **a**). The activity of A is represented by a Boolean variable. It is inhibited by the complex AB, otherwise it is always 'on'. The activity of AB is represented by a Boolean variable and is stimulated if both A and B are active. Finally, the activity of B is represented by three values. It can be off, low or high if A is on (and stimulates its production) and AB is off. Note that in the following expressions, B is true if B = 1 or B = 2.

- A = 0 if AB
- A = 1 if not AB
- B = 0 if not A and AB
- B = 1 if (not A and not AB) or (A and AB)
- B = 2 if A and not AB
- AB = 0 if not A or not B
- AB = 1 if A and B

The model was implemented using the GINsim software[140]. The synchronous simulation of the logic rules permits the tracing of trajectories across the ensemble of states. The combination of all of these trajectories forms the state-transition graph (see the figure, part **b**). Whatever the starting state, the system will end up as a circular attractor in which all three variables oscillate.



Le Novère, N. (2015). Quantitative and logic modelling of molecular and gene networks. Nature Reviews Genetics, 16(3), 146-158.

Redes booleanas

- Formalismo usado en la descripción de GRN al nivel de la expresión protéica.
- El **nivel de expresión** es considerado como *prendido* (1) o *apagado* (0).
- El **estado** del modelo es un conjunto de valores booleanos que describen los niveles de transcripción de la GN en un punto en el tiempo.
- La *inhibición* o *activación* de genes es modelado mediante reglas booleanas, combinadas usando lógica booleana.

Tjelvar S. G. Olsson. (2015). The Biologist's Guide to Computing. http://book.biologistsguide2computing.com/en/stable/index.html

Redes booleanas

- La aplicación *iterativa* de la reglas boleanas permite evaluar la **dinámica del sistema** en el tiempo.

- A medida que progresa el tiempo, el número de estados de la red disminuye: el sistema se dirige hacia un número pequeño de **cíclos dinámicos** y **estados estables**, conocidos como *atractores*.

- Los **atractores** frecuentemente corresponden a estados específicos y diferenciados de la célula.

Tjelvar S. G. Olsson. (2015). The Biologist's Guide to Computing. http://book.biologistsguide2computing.com/en/stable/index.html

Autómatas finitos

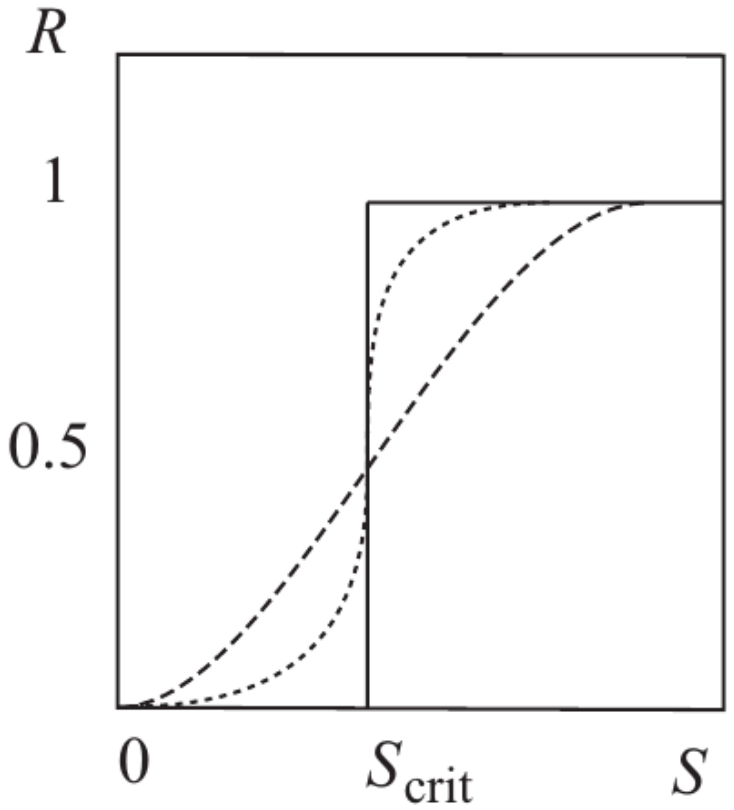
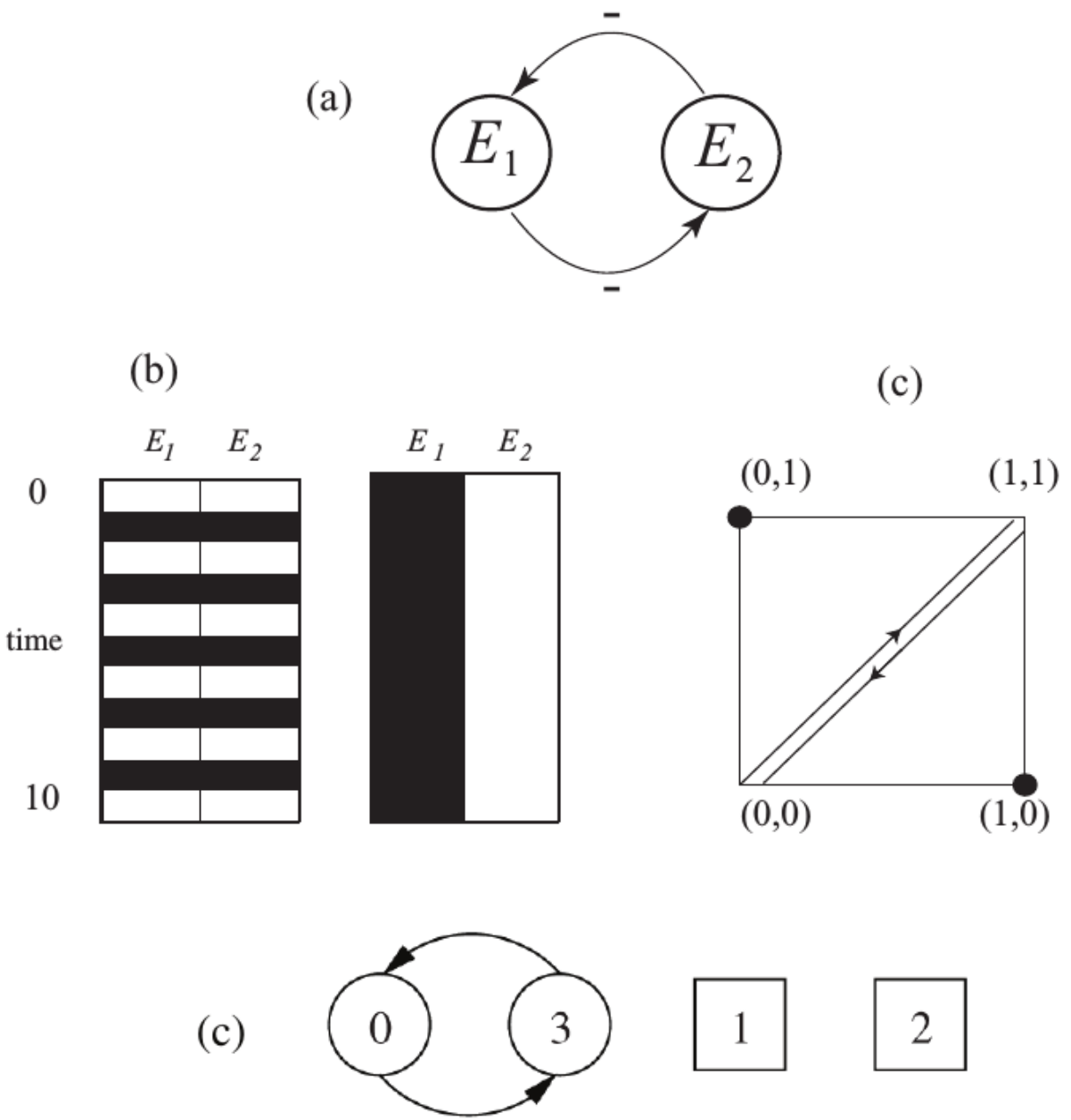


Figure 13.1. *The Boolean on-off (solid line) as an approximation for continuous responses (dashed lines). Here we show schematically how the Boolean response R to some signal S compares with typical smooth responses that we have modeled using Hill functions.*

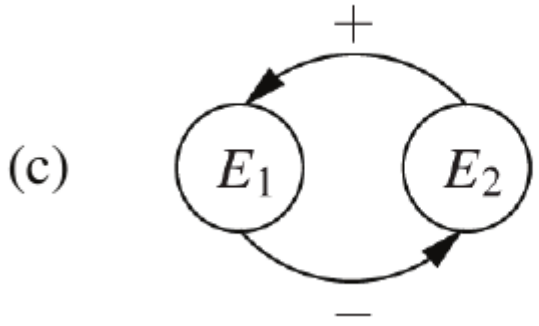
Segel, L. A., & Edelstein-Keshet, L. (2013). A Primer in Mathematical Models in Biology (Vol. 129). SIAM.

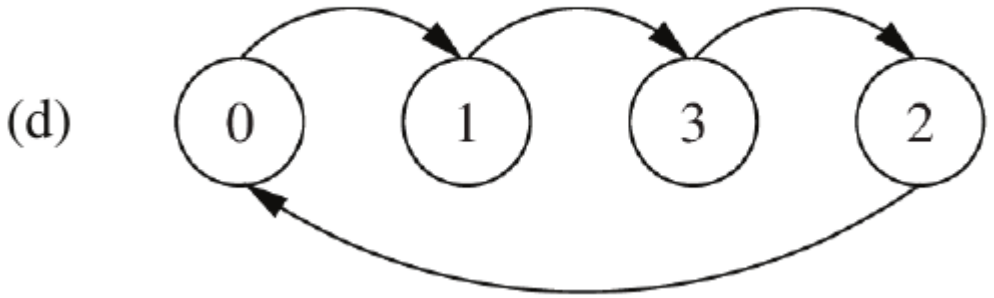
Autómatas finitos

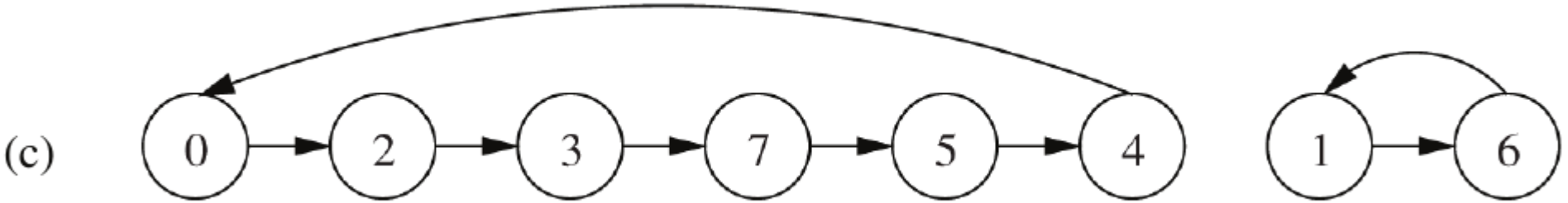


Segel, L. A., & Edelstein-Keshet, L. (2013). A Primer in Mathematical Models in Biology (Vol. 129). SIAM.

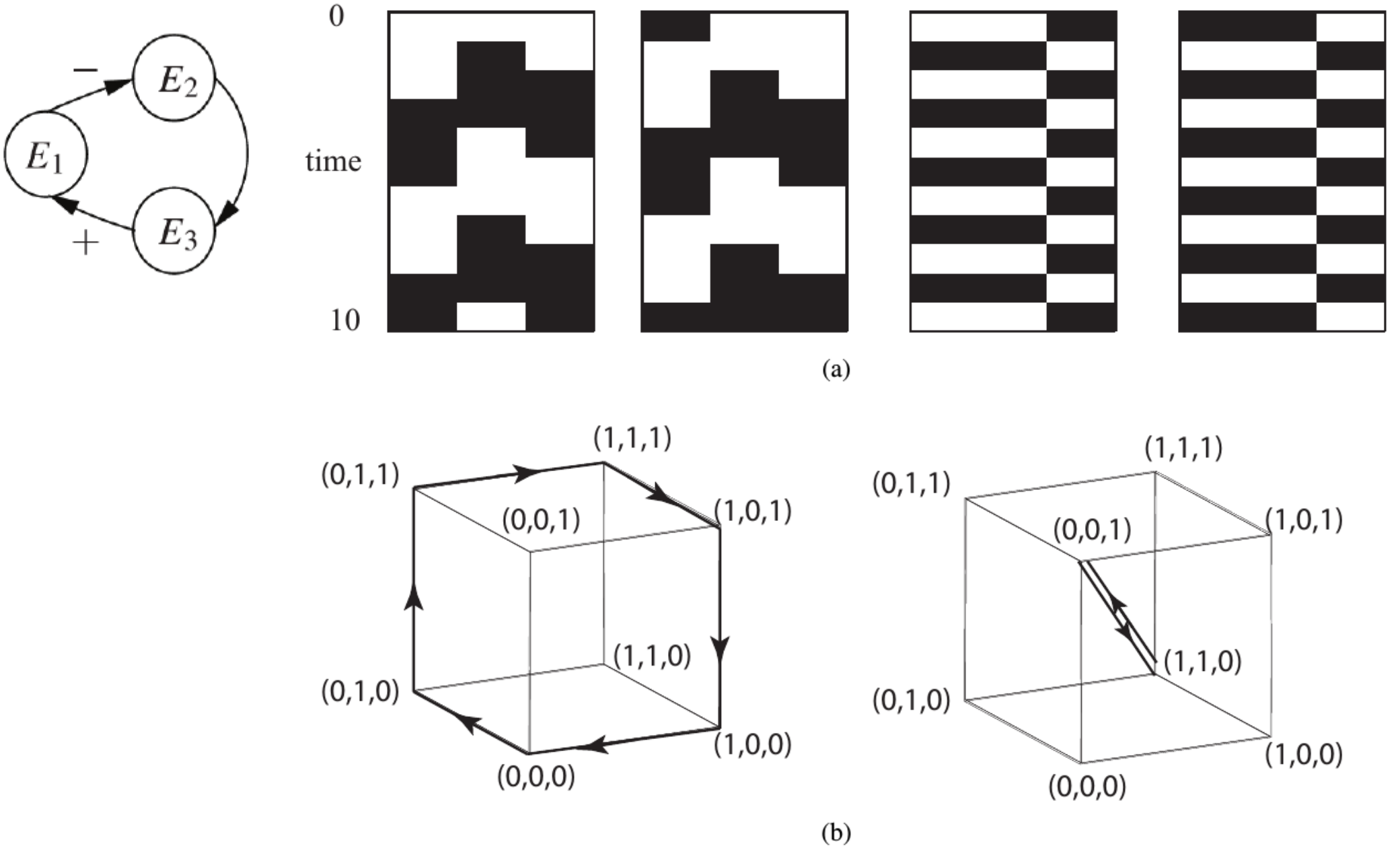
Autómatas finitos







Segel, L. A., & Edelstein-Keshet, L. (2013). A Primer in Mathematical Models in Biology (Vol. 129). SIAM.

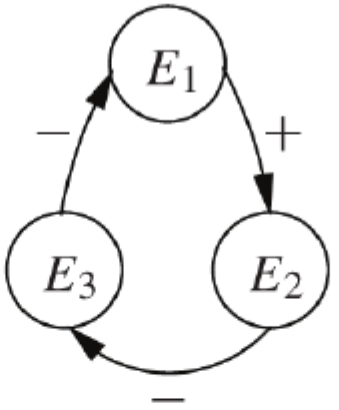


Segel, L. A., & Edelstein-Keshet, L. (2013). A Primer in Mathematical Models in Biology (Vol. 129). SIAM.

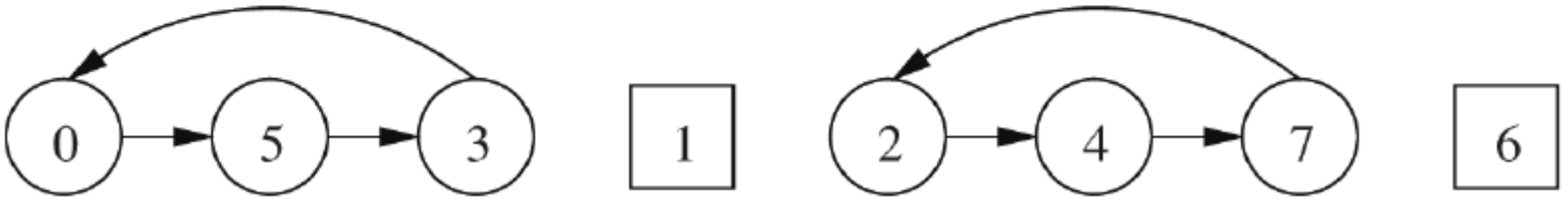
Figure 13.6. (a) *Dynamics of the three-element loop network shown in Fig.* 13.5 *with parallel updating over* 10 *time steps starting with a few different initial conditions. Time* $= 0$ *at the top of each panel. The state of the system is shown as a row* E_1, E_2, E_3 *at each time step, starting from the initial conditions* $(0,0,0),(1,0,0),(0,0,1),(1,1,0)$. *Panels*

Autómatas finitos

(a)



(b)



Segel, L. A., & Edelstein-Keshet, L. (2013). A Primer in Mathematical Models in Biology (Vol. 129). SIAM.

Autómatas finitos

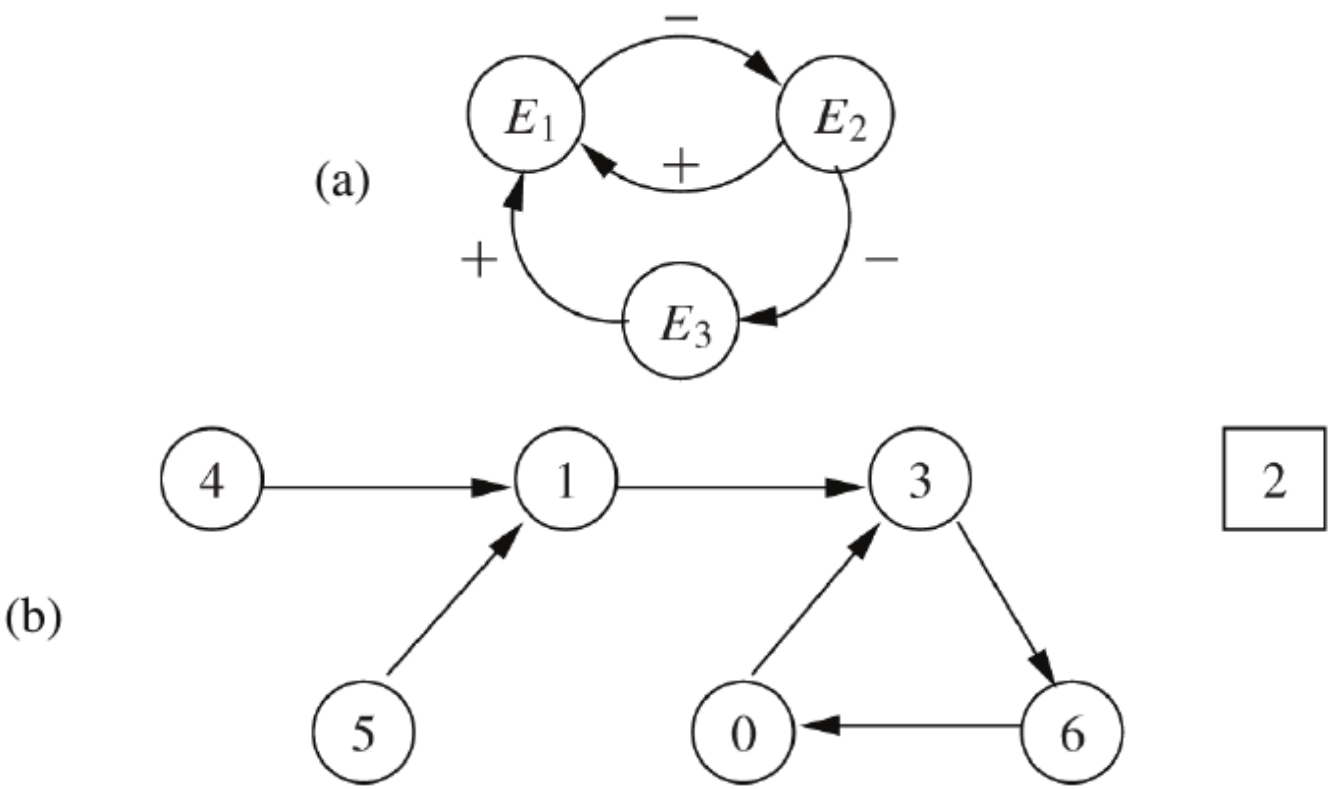


Figure 13.8. (a) *Small gene network with excitation (+) and inhibition (−). Weak excitation:* $S_1' = S_2 \cdot S_3,\ S_2' = \overline{S}_1,\ S_3' = \overline{S}_2$. *Strong excitation:* $S_1' = S_2 + S_3,\ S_2' = \overline{S}_1,\ S_3' = \overline{S}_2$. (b) *Behavior of the weak excitation network under parallel updating. See Exercise* 13.13a. *There are two attractors, a cycle (states* 0, 3, 6*), and an "isolated" steady state (state* 2*) that is not attained from any other state.*

Segel, L. A., & Edelstein-Keshet, L. (2013). A Primer in Mathematical Models in Biology (Vol. 129). SIAM.

Autómatas finitos

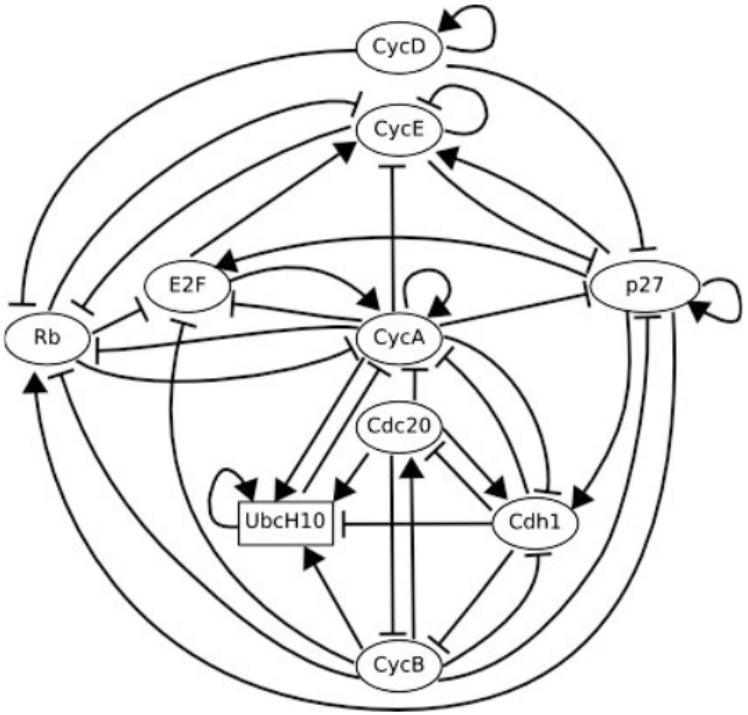
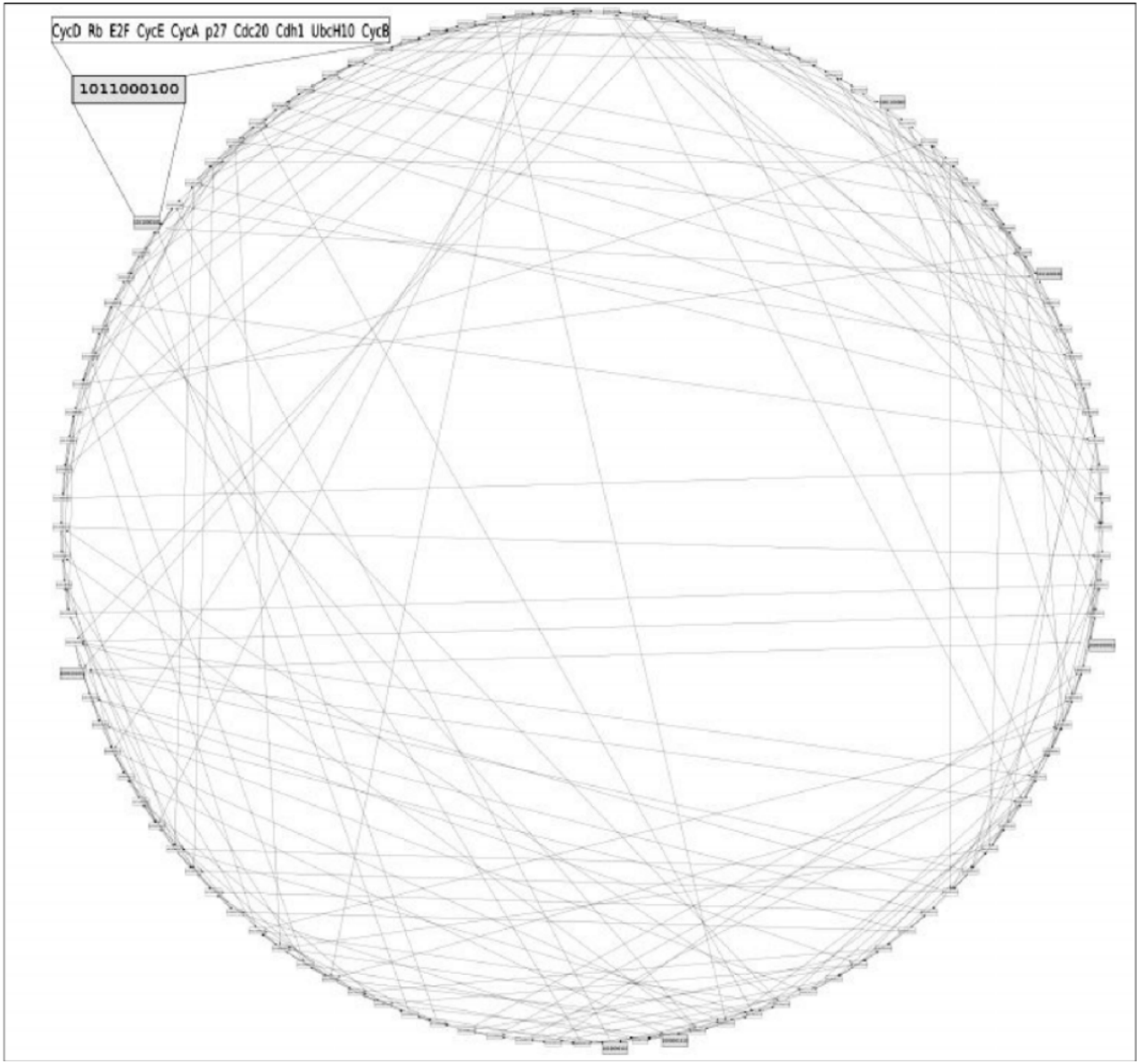


Fig. 1. Logical regulatory graph for the mammalian cell cycle network. Each node represents the activity of a key regulatory element, whereas the edges represent cross-regulations. Blunt arrows stand for inhibitory effects, normal arrows for activations.



Fauré, A., Naldi, A., Chaouiya, C., & Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics, 22(14), e124-e131.

Table 1. Logical rules underlying the definition of the logical parameters associated with the regulatory graph of Figure 1

Product	Logical rules leading to an activity of the product	Justification/References
CycD	*CycD*	CycD is an input, considered as constant.
Rb	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{CycD} \wedge \overline{CycB})$	Rb is expressed in the absence of the cyclins, which inhibit it by phosphorylation (Novak and Tyson, 2004; Taya, 1997); it can be expressed in the presence of CycE or CycA if their inhibitory activity is blocked by p27 (Coqueret, 2003).
E2F	$(\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{Rb} \wedge \overline{CycB})$	E2F is active in the absence of Rb, that blocks E2F self-transcriptional activation (Helin, 1998), and in the absence of CycA and CycB, that inhibit E2F (Novak and Tyson, 2004); CycA may be present, if its inhibitory activity is blocked by p27 (Coqueret, 2003).
CycE	$(E2F \wedge \overline{Rb})$	CycE activity requires the presence of E2f and the absence of Rb (Helin, 1998).
CycA	$(E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge Ubc)}) \vee (CycA \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge Ubc)})$	The transcription of CycA is activated by E2F in the absence of Rb, which blocks this activation (Helin, 1998), in the absence of Cdc20, as well as of the pair formed by Cdh1 and UbcH10, which both lead to the degradation of CycA (Harper *et al*., 2002; Rape and Kirschner, 2004); CycA is stable in the absence of its inhibitors Rb, Cdc20, and of the pair Cdh1 and UbcH10.
p27	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{(CycE \wedge CycA)} \wedge \overline{CycB} \wedge \overline{CycD})$	p27 is active in the absence of the cyclins; when p27 is already present, it blocks the action of CycE or CycA (but not both of them) by sequestration (Coqueret, 2003).
Cdc20	*CycB*	CycB indirectly activates Cdc20 (Harper *et al*., 2002).
Cdh1	$(\overline{CycA1} \wedge \overline{CycB}) \vee (Cdc20) \vee (p27 \wedge \overline{CycB})$	The activity of Cdh1 requires the absence of CycB and CycA, which inhibit it by phosphorylation (Harper *et al*., 2002); Cdc20 further activates Cdh1. (Novak and Tyson, 2004); p27 allows the presence of CycA, by blocking its activity.
UbcH10	$(\overline{Cdh1}) \vee (Cdh1 \wedge Ubc \wedge (Cdc20 \vee CycA \vee CycB))$	UbcH10 is active in the absence of Cdh1; this UbcH10 activity can be maintained in the presence of Cdh1 when at least one of its other targets is present (CycA, Cdc20, or CycB) (Rape and Kirschner, 2004).
CycB	$(\overline{Cdc20} \wedge \overline{Cdh1})$	CycB is active in the absence of both Cdc20 and Cdh1, which target CycB for destruction (Harper *et al*., 2002).

The names of the components of the regulatory graph of Figure 1 are listed in the first column. For each one, the second column gives the logical rules specifying its behaviour. More precisely, we have described only the situations where the component is activated (value of the corresponding Boolean variable set to 1), all other situations leading to an inactivation. This description is based on the classical logical formulation, where ''$\wedge$'' stands for ''AND'', ''$\vee$'' stands for (inclusive) ''OR'', and the negation is written by a bar over the term. As an example, considering the case of CycE, there are eight non-zero parameters attached to CycE, specifying the different combinations of incoming interactions which lead to an activation of CycE (cf. the GINML file on the *GINsim* website). These can be summarised by the logical formula ''E2F active and Rb not active, whatever the state of the other components''. Finally the last column provides some justifications for the logical rules, together with references.

Fauré, A., Naldi, A., Chaouiya, C., & Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics, 22(14), e124-e131.

Autómatas finitos

```
> library(BoolNet)
> data(cellcycle)
> cellcycle
Boolean network with 10 genes

Involved genes:
CycD Rb E2F CycE CycA p27 Cdc20 Cdh1 UbcH10 CycB

Transition functions:
CycD = CycD
Rb = (! CycA & ! CycB & ! CycD & ! CycE) | (p27 & ! CycB & ! CycD)
E2F = (! Rb & ! CycA & ! CycB) | (p27 & ! Rb & ! CycB)
CycE = (E2F & ! Rb)
CycA = (E2F & ! Rb & ! Cdc20 & ! (Cdh1 & UbcH10)) | (CycA & ! Rb & ! Cdc20 & ! (Cdh1 & UbcH10))
p27 = (! CycD & ! CycE & ! CycA & ! CycB) | (p27 & ! (CycE & CycA) & ! CycB &! CycD)
Cdc20 = CycB
Cdh1 = (! CycA & ! CycB) | (Cdc20) | (p27 & ! CycB)
UbcH10 = ! Cdh1 | (Cdh1 & UbcH10 & (Cdc20 | CycA | CycB))
CycB = ! Cdc20 & ! Cdh1
```

The cell cycle network is a classical Boolean network, where each transition function only depends on the previous state of the network. E.g., `CycB, ! Cdc20 & ! Cdh1` can be written formally as $CycB(t) = \neg Cdc20(t-1) \wedge \neg Cdh1(t-1)$. As already discussed before, `BoolNet` also incorporates

Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics, 26(10), 1378-1380.

Autómatas finitos

To calculate all state transitions in a synchronous network until an attractor is reached, you can call

```
> path <- getPathToAttractor(cellcycle, rep(0,10))
> path
```

	CycD	Rb	E2F	CycE	CycA	p27	Cdc20	Cdh1	UbcH10	CycB
1	0	0	0	0	0	0	0	0	0	0
2	0	1	1	0	0	1	0	1	1	1
3	0	0	0	0	0	0	1	0	1	0
4	0	1	1	0	0	1	0	1	1	0
5	0	1	0	0	0	1	0	1	0	0

A sequence can be visualized by plotting a table of state changes:

```
> plotSequence(sequence=path)
```

Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics, 26(10), 1378-1380.

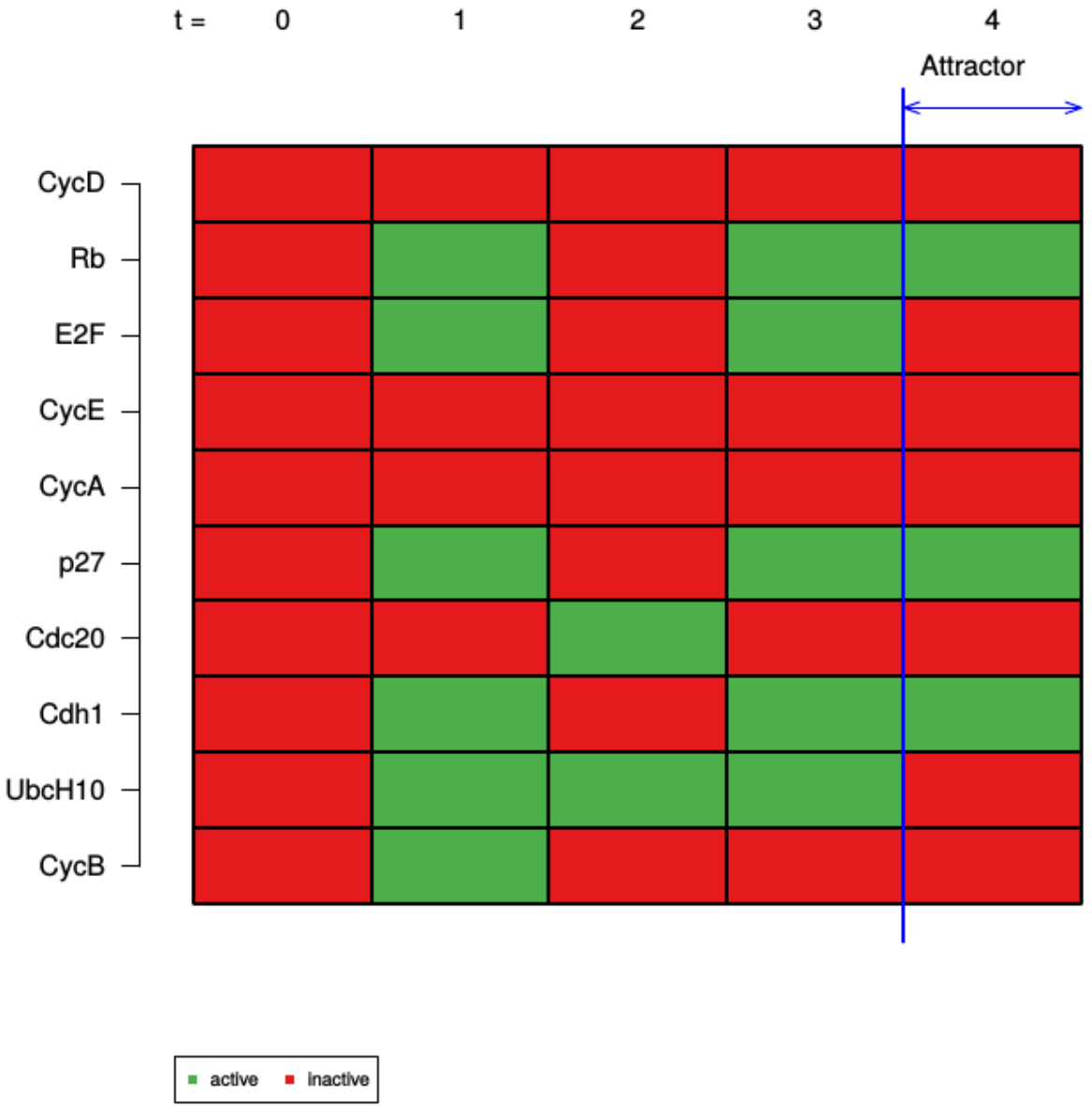
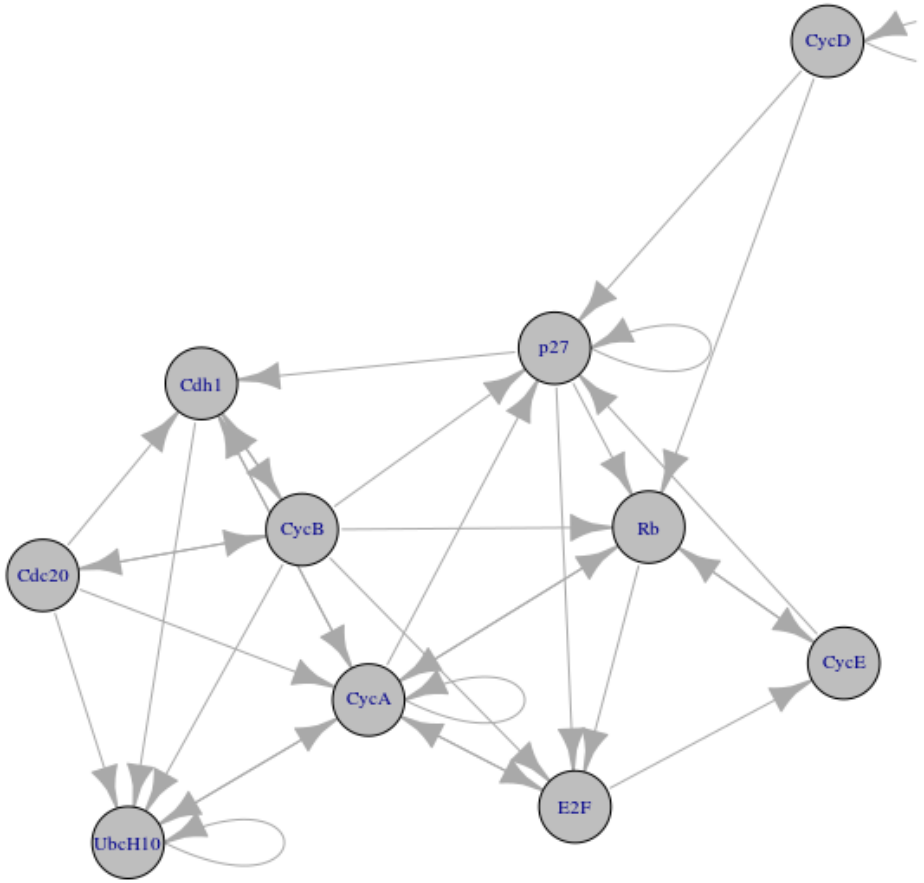


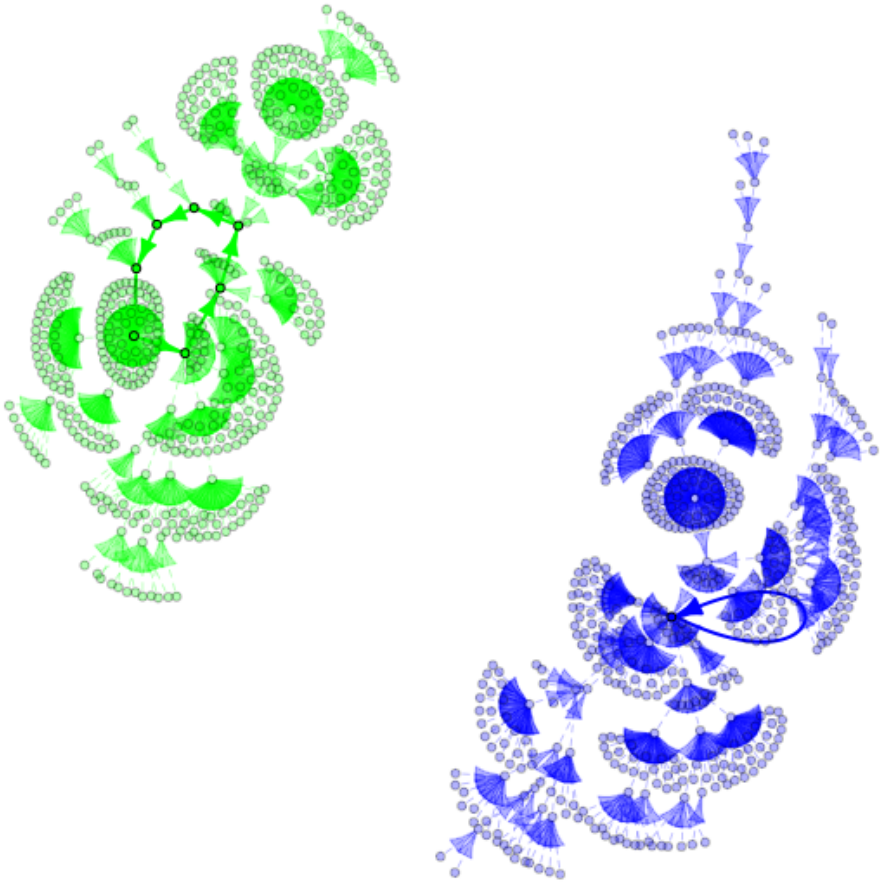
Figure 2: Visualization of a sequence of states in the mammalian cell cycle network. The columns of the table represent consecutive states of the time series. The last state is the steady-state attractor of the network.

Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics, 26(10), 1378-1380.

Autómatas finitos

```
> attr <- getAttractors(cellcycle)
> par(mfrow=c(1,2))
> plotNetworkWiring(cellcycle)
> plotStateGraph(attr)
```





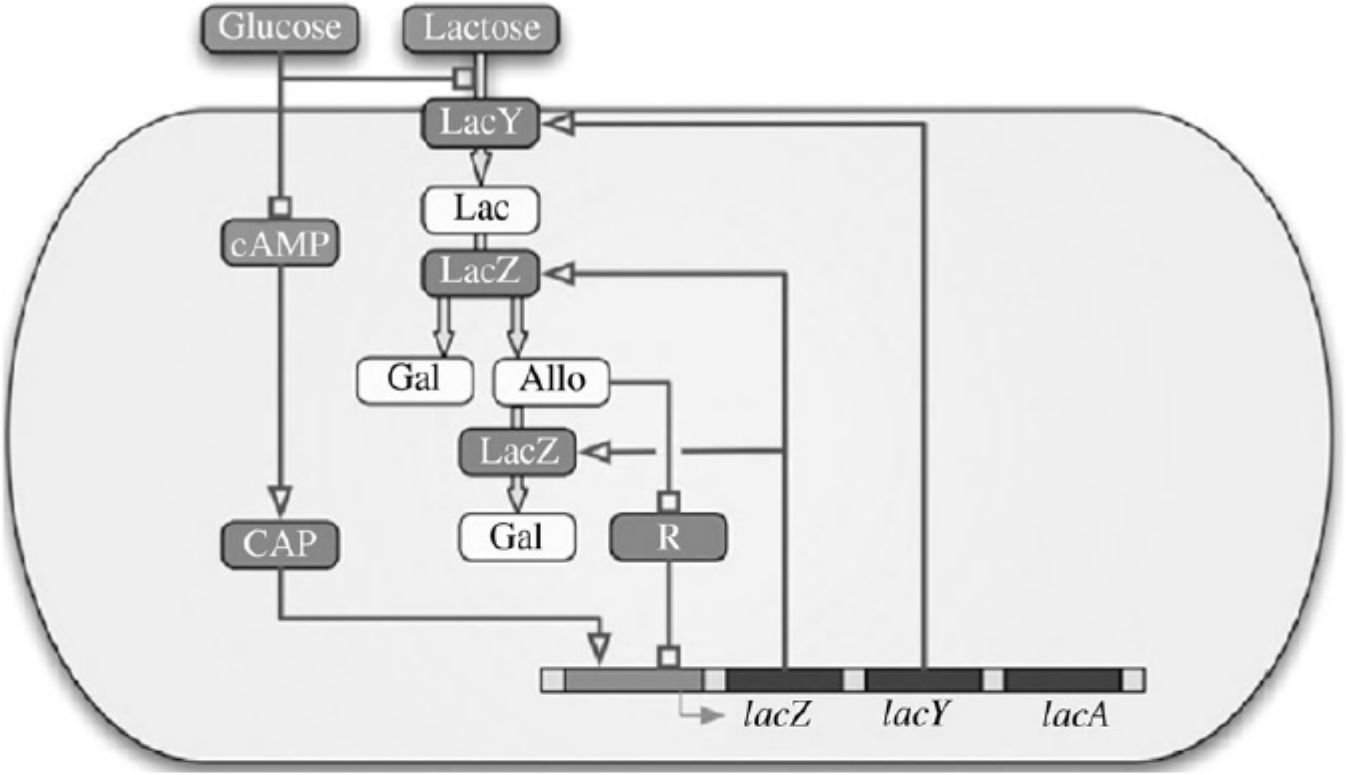
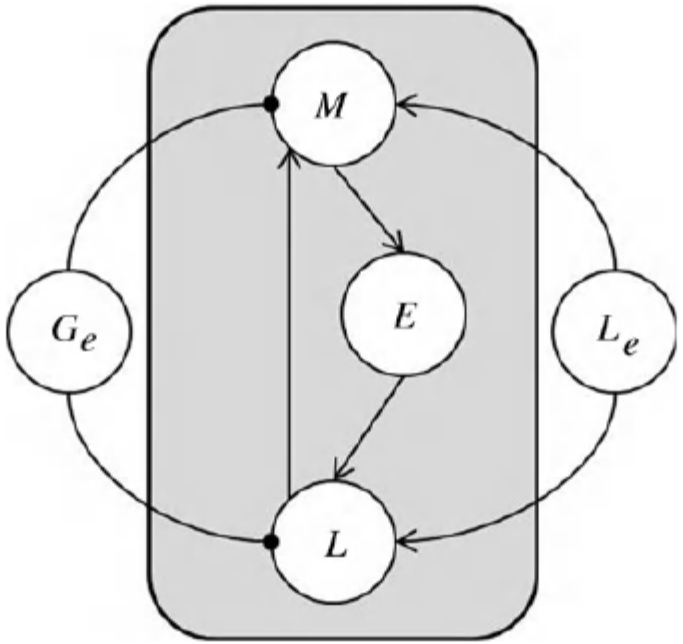


FIGURE 1.3

Schematic of the *lac* operon regulatory mechanism (from Santillan at al. [14]) Protein LacY is a permease that transports external lactose into the cell. Protein LacZ polymerizes into a homotetramer named β-galactosidase. This enzyme transforms internal lactose (Lac) to allolactose (Allo) or to glucose and galactose (Gal). It also converts allolactose to glucose and galactose. Allolactose can bind to the repressor (R) inhibiting it. When not bound by allolactose, R can bind to a specific site upstream of the operon structural genes and thus prevent transcription initiation. External glucose inhibits the production of cAMP that, when bound to the CRP protein (also known as CAP) to form the CAP-cAMP complex, acts as an activator of the *lac* operon. External glucose also inhibits lactose uptake by permease proteins. Reprinted from *Biophysics Journal*, Vol. 92, M. Santillan, M. C. Mackey, and E. S. Zeron, Origin of bistability in the lac operon 3830 - 3842, Copyright (2007), with permission from Elsevier.

Robeva, R., & Hodge, T. (Eds.). (2013). Mathematical concepts and methods in modern biology: using modern discrete models. Academic Press.

Autómatas finitos



$$x_M(t+1) = f_M(t+1) = \overline{G_e} \wedge (L(t) \vee L_e(t))$$
$$x_E(t+1) = f_E(t+1) = M(t)$$
$$x_L(t+1) = f_L(t+1) = \overline{G_e} \wedge ((E(t) \wedge L_e(t)) \vee (L(t) \wedge \overline{E}(t))).$$

FIGURE 1.4

Wiring diagram for the minimal model. *E* denotes the *LacZ* polypeptide, *M* – the mRNA, *L*- internal lactose. L_e and G_e denote external lactose and glucose, respectively. The nodes in the shaded rectangle represent the model variables while the outside nodes represent the model parameters. Directed links represent influences between the variables: A positive influence is indicated by an arrow; a negative influence is depicted by a circle. Self-regulating links (for nodes that may influence themselves) are not pictured.

Robeva, R., & Hodge, T. (Eds.). (2013). Mathematical concepts and methods in modern biology: using modern discrete models. Academic Press.

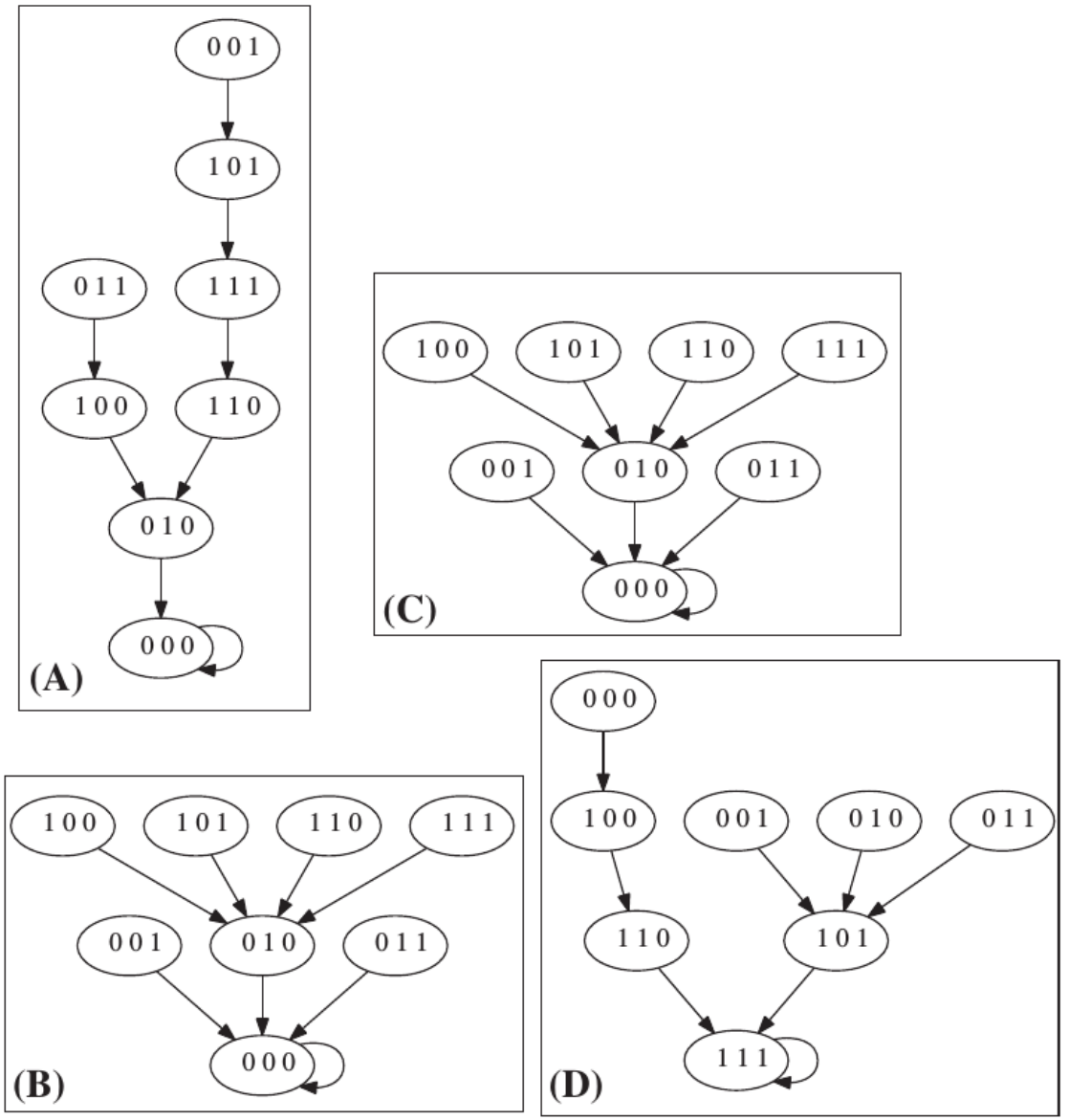


FIGURE 1.7

The state space transition diagram of triples (M, E, L) for the Boolean model of the *lac* operon defined in Eqs. (1.4) for the four possible combinations of parameter values. In each of the cases, the operon has a single fixed point and no limit cycles. Panel (A) $L_e = 0; G_e = 0$; The operon is Off. Panel (B) $L_e = 0; G_e = 1$; The operon is Off. Panel (C) $L_e = 1; G_e = 1$; The operon is Off. Panel (D) $L_e = 1; G_e = 0$; The operon is On. Graphs obtained using DVD [12].

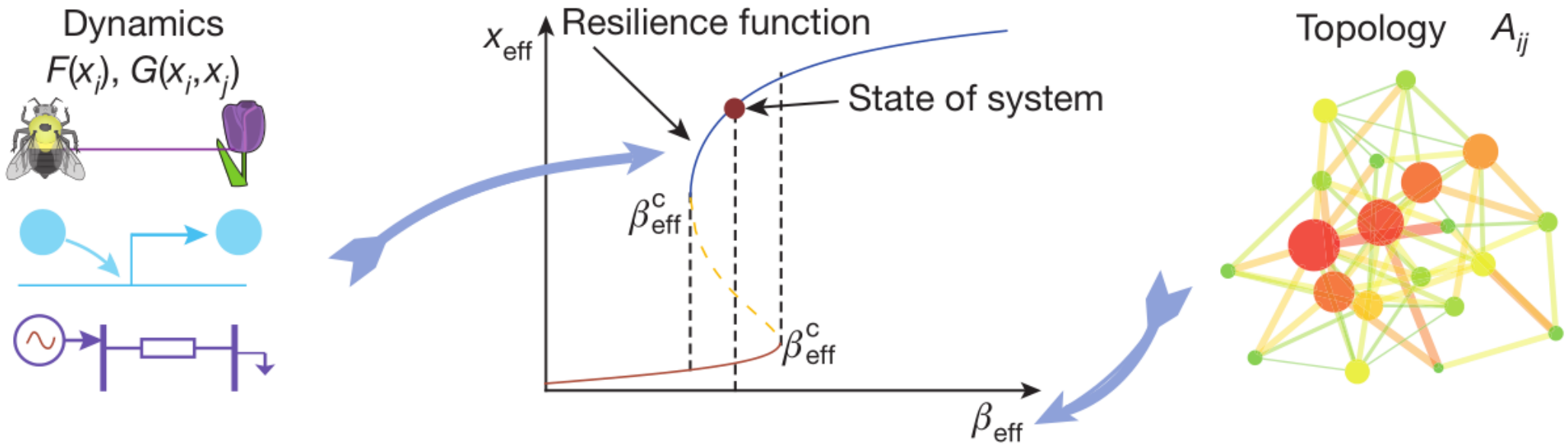
Robeva, R., & Hodge, T. (Eds.). (2013). Mathematical concepts and methods in modern biology: using modern discrete models. Academic Press.

LETTER

doi:10.1038/nature16948

Universal resilience patterns in complex networks

Jianxi Gao[1]*, Baruch Barzel[2]* & Albert-László Barabási[1,3,4,5]



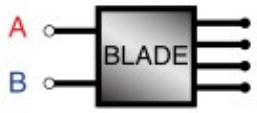
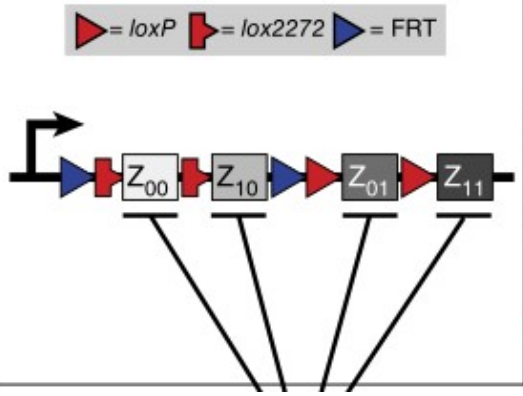
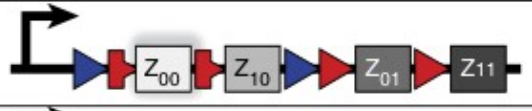
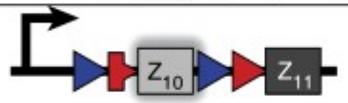
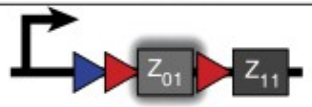
Reciliencia: Habilidad del sistema para ajustar su actividad con el fin de retener su funcionalidad básica ante errores, fallas y cambios ambientales.

nature biotechnology

Large-scale design of robust genetic circuits with multiple inputs and outputs for mammalian cells

Benjamin H Weinberg[1], N T Hang Pham[1], Leidy D Caraballo[1], Thomas Lozanoski[1], Adrien Engel[1,2], Swapnil Bhatia[3] & Wilson W Wong[1]

Engineered genetic circuits for mammalian cells often require extensive fine-tuning to perform as intended. We present a robust, general, scalable system, called 'Boolean logic and arithmetic through DNA excision' (BLADE), to engineer genetic circuits with multiple inputs and outputs in mammalian cells with minimal optimization. The reliability of BLADE arises from its reliance on recombinases under the control of a single promoter, which integrates circuit signals on a single transcriptional layer. We used BLADE to build 113 circuits in human embryonic kidney and Jurkat T cells and devised a quantitative, vector-proximity metric to evaluate their performance. Of 113 circuits analyzed, 109 functioned (96.5%) as intended without optimization. The circuits, which are available through Addgene, include a 3-input, two-output full adder; a 6-input, one-output Boolean logic look-up table; circuits with small-molecule-inducible control; and circuits that incorporate CRISPR–Cas9 to regulate endogenous genes. BLADE enables execution of sophisticated cellular computation in mammalian cells, with applications in cell and tissue engineering.

Logic gate			Truth table						
			IN		OUT				
Name	Symbol	Architecture	A Cre	B Flp	Z_{00}	Z_{10}	Z_{01}	Z_{11}	Recombined architecture
2-input BLADE template			0	0	1	0	0	0	
			1	0	0	1	0	0	
			0	1	0	0	1	0	
			1	1	0	0	0	1	

An Expanded View of Complex Traits: From Polygenic to Omnigenic

Evan A. Boyle,[1,*] Yang I. Li,[1,*] and Jonathan K. Pritchard[1,2,3,*]

A central goal of genetics is to understand the links between genetic variation and disease. Intuitively, one might expect disease-causing variants to cluster into key pathways that drive disease etiology. But for complex traits, association signals tend to be spread across most of the genome—including near many genes without an obvious connection to disease. We propose that gene regulatory networks are sufficiently interconnected such that all genes expressed in disease-relevant cells are liable to affect the functions of core disease-related genes and that most heritability can be explained by effects on genes outside core pathways. We refer to this hypothesis as an "omnigenic" model.