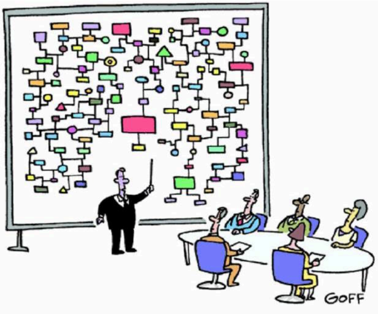
Logical modelling of cellular regulatory networks

Claudine Chaouiya
claudine.chaouiya@univ-amu.fr

L3 SV Bioinformatique: Réseaux et régulation
COURS 4



And that's why we need a computer

February 2019

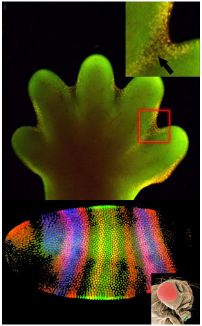
Motivation

Assessing the mechanisms driving cellular responses

Complex cellular processes (proliferation, differentiation, apoptosis,...) are controlled by heterogeneous, complex interaction networks

Frog development

Mouse forelimb development



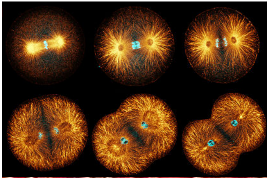
Drosophila Development

Motivation

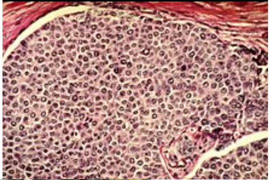
Assessing the mechanisms driving cellular responses

Complex cellular processes (proliferation, differentiation, apoptosis,...) are controlled by heterogeneous, complex interaction networks

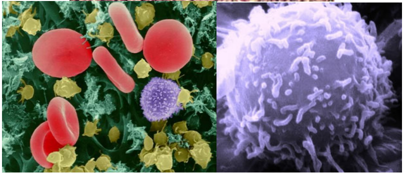
Cell division



Breast cancer cells

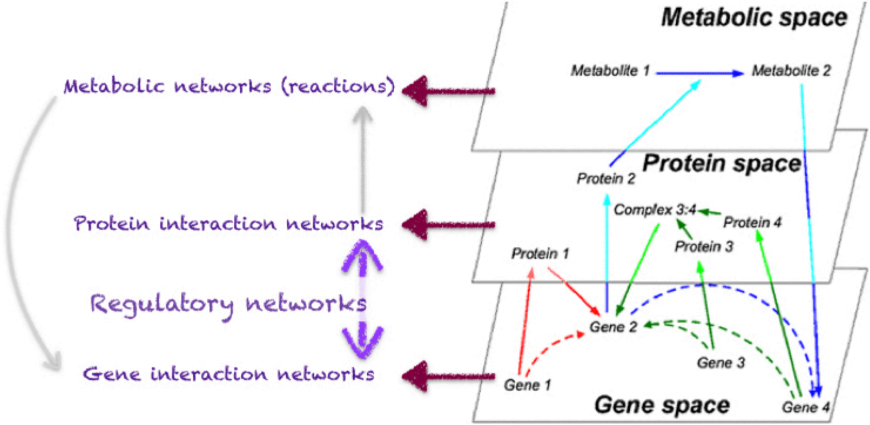


Immune system (blood cells)



Motivation

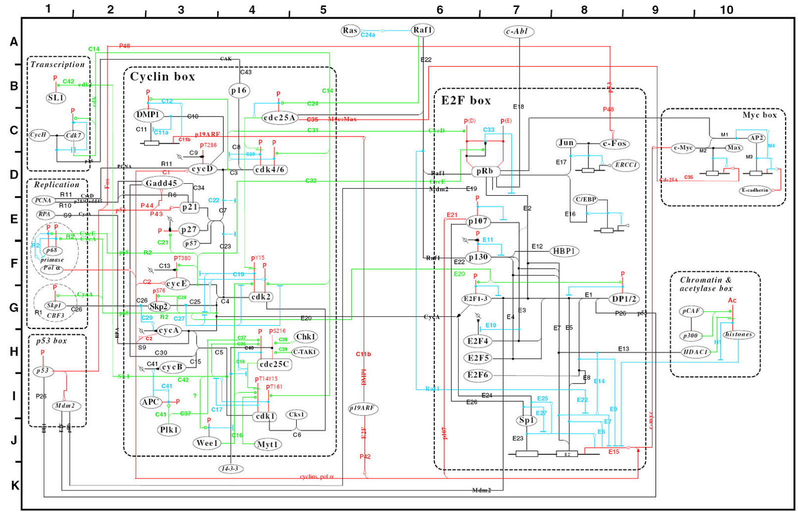
Assessing the mechanisms driving cellular responses



Different types of networks: here, we focus on regulatory, *i.e.* influence networks

Motivation

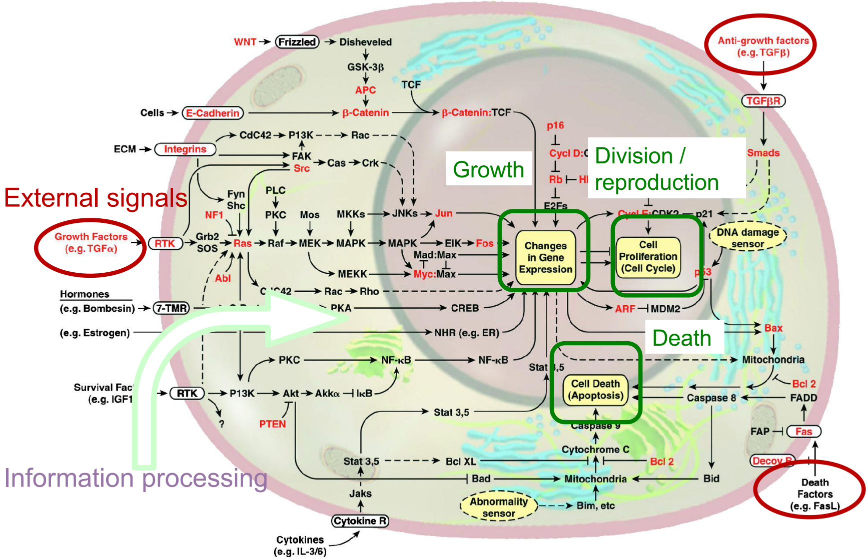
Assessing the mechanisms driving cellular responses



Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems. K. Kohn (1999) Mol Biol Cell

Motivation

Assessing the mechanisms driving cellular responses



Cancer Hallmarks. Hanahan & Weinberg (2000) Cell, 100:57-70

Motivation

Beyond the network topology... dynamical models

Aims

- Assess the behaviour driven by the network
- Understand the role of individual components and interactions
- Suggest missing components and interactions
- Predict behaviours upon perturbations

Motivation

Beyond the network topology... dynamical models

Aims

- Assess the behaviour driven by the network
- Understand the role of individual components and interactions
- Suggest missing components and interactions
- Predict behaviours upon perturbations

Advantages of mathematical and computer tools

- Precise and unambiguous description of the network & relations
- *In silico* experiments are cheap and easy!

Motivation

Beyond the network topology... dynamical models

Aims

- Assess the behaviour driven by the network
- Understand the role of individual components and interactions
- Suggest missing components and interactions
- Predict behaviours upon perturbations

Advantages of mathematical and computer tools

- Precise and unambiguous description of the network & relations
- *In silico* experiments are cheap and easy!

Static *vs* dynamical models of biological networks

- Static models → topology of the networks (nodes and edges)
- Dynamical models → dynamics of the variables associated with the network nodes (nodes, edges, functions)

Systems Biology ⟶ *Use of mathematics to study how genes and proteins interact to produce the complex behaviors of a living cell (J. Tyson)*

Motivation

Beyond the network topology... dynamical models

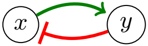
A variety of mathematical / computational formalisms Choice depends on the data at hand, the size of the network, the question to be assessed, etc.

Motivation

Beyond the network topology... dynamical models

A variety of mathematical / computational formalisms Choice depends on the data at hand, the size of the network, the question to be assessed, etc.

Example of a small regulatory circuit

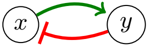


Motivation

Beyond the network topology... dynamical models

A variety of mathematical / computational formalisms Choice depends on the data at hand, the size of the network, the question to be assessed, etc.

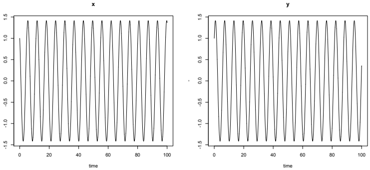
Example of a small regulatory circuit



System of ordinary differential equations

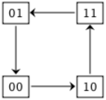
ODE

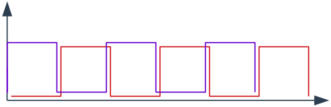
$$\begin{cases} \frac{dx(t)}{dt} = -y(t) \\ \frac{dy(t)}{dt} = x(t) \end{cases}$$



System of logical equations

$$\begin{cases} f_x(x,y) = not(y) \\ f_y(x,y) = x \end{cases}$$



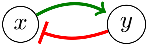


Motivation

Beyond the network topology... dynamical models

A variety of mathematical / computational formalisms Choice depends on the data at hand, the size of the network, the question to be assessed, etc.

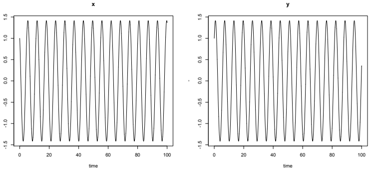
Example of a small regulatory circuit



System of ordinary differential equations

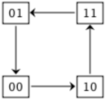
ODE

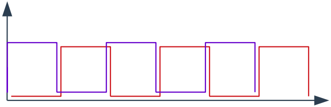
$$\begin{cases} \frac{dx(t)}{dt} = -y(t) \\ \frac{dy(t)}{dt} = x(t) \end{cases}$$



System of logical equations

$$\begin{cases} f_x(x,y) = not(y) \\ f_y(x,y) = x \end{cases}$$



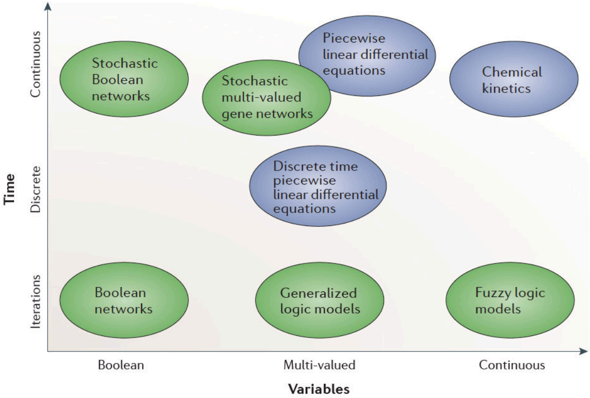


Essentially, all models are wrong, but some are useful (George Box)

Motivation

Beyond the network topology... dynamical models

A variety of mathematical / computational formalisms Choice depends on the data at hand, the size of the network, the question to be assessed, etc.



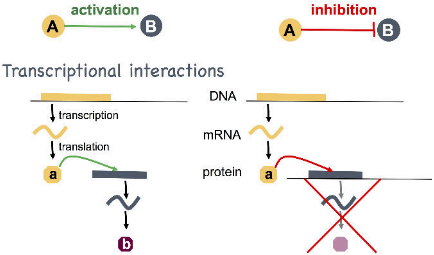
N Le Novère, Nat Rev Genet 2015

Motivation

Logical modelling

- Lack of precise, quantitative data (concentrations, kinetics)
- Mostly qualitative observations
- Ever larger networks
- Non-linear regulatory effects

Influence networks controlling cell fates

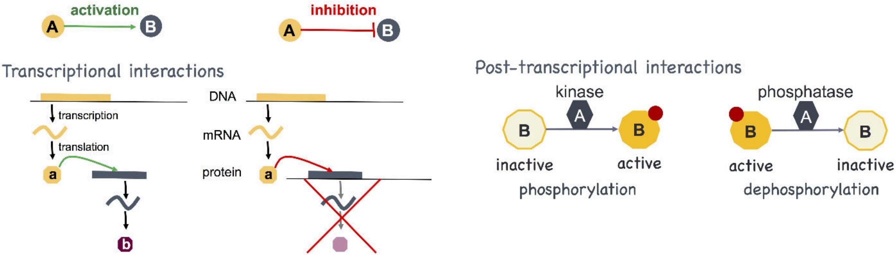


Motivation

Logical modelling

- Lack of precise, quantitative data (concentrations, kinetics)
- Mostly qualitative observations
- Ever larger networks
- Non-linear regulatory effects

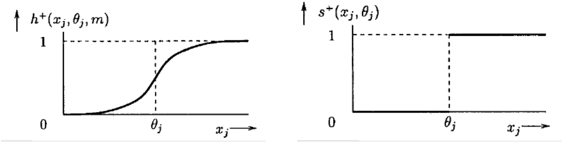
Influence networks controlling cell fates



Motivation

Logical modelling

- Lack of precise, quantitative data (concentrations, kinetics)
- Mostly qualitative observations
- Ever larger networks
- Non-linear regulatory effects



⟶ Boolean networks: each regulatory component associated to a Boolean variable representing its levels of activity, of concentration, etc.

⟶ Extension to multi-valued variables

Basics of the logical modelling framework

50 years of logical modelling

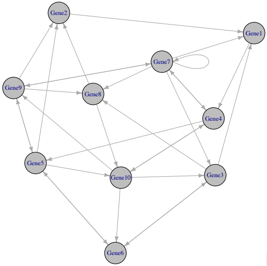
S. Kauffman (1969) Journal of Theor. Biol. 22 (3): 437-67

- (Random) Boolean networks to investigate generic self-organizing properties of gene networks
- N-K networks Random connections, N nodes with degree K, Random regulatory (Boolean) functions
- Cell types ⟶ attractors in gene networks
- Cell differentiation ⟶ transitions between attractors Focus on asymptotic behaviours

Basics of the logical modelling framework

50 years of logical modelling

S. Kauffman (1969) Journal of Theor. Biol. 22 (3): 437-67



N=10, K=3

Transition function F:

Gene1 = (!Gene2 & !Gene3) | (Gene2 & !Gene7 & Gene3)
Gene2 = (!Gene8 & Gene5) | (Gene5 & Gene9)
Gene3 = (Gene6 & !Gene10 & !Gene7) | (Gene6 & Gene10 & Gene7)
Gene4 = (Gene10 & !Gene1) | (Gene7 & Gene1)
Gene5 = (Gene6 & !Gene9) | (Gene4 & !Gene9) | (Gene4 & !Gene6)
Gene6 = (!Gene10 & !Gene3) | (!Gene10 & Gene5) | (Gene10 & !Gene5 & Gene3)
Gene7 = (Gene9 & !Gene4) | (Gene9 & Gene7)
Gene8 = (!Gene7 & !Gene9) | (!Gene3) | (Gene7 & Gene9)
Gene9 = (!Gene10 & !Gene5 & Gene7) | (Gene10 & Gene5 & Gene7)
Gene10 = (!Gene4 & !Gene5 & !Gene8) | (Gene5 & Gene8)
| (Gene4 & Gene8) | (Gene4 & Gene5)

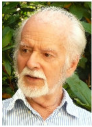
Synchronous update: $x^{t+1} = F(x_t)$

Basics of the logical modelling framework

50 years of logical modelling


S. Kauffman (1969) Journal of Theor. Biol. 22 (3): 437-67

- (Random) Boolean networks to investigate generic self-organizing properties of gene networks
- N-K networks Random connections, N nodes with degree K, Random regulatory (Boolean) functions
- Cell types ⟶ attractors in gene networks
- Cell differentiation ⟶ transitions between attractors Focus on asymptotic behaviours

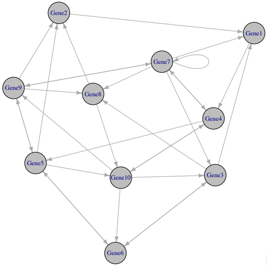

R. Thomas (1973) Journal of Theor. Biol. 42: 563?85.

- Boolean networks to investigate the dynamics of gene networks
- Regulated switch (Lysis *vs* lysogeny) of the bacteriophage λ
- Asynchronous update
- Extension to multi-valued variables

Basics of the logical modelling framework

50 years of logical modelling

S. Kauffman (1969) Journal of Theor. Biol. 22 (3): 437-67

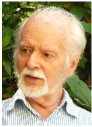


N=10, K=3

Transition function F:

Gene1 = (!Gene2 & !Gene3) | (Gene2 & !Gene7 & Gene3)
Gene2 = (!Gene8 & Gene5) | (Gene5 & Gene9)
Gene3 = (Gene6 & !Gene10 & !Gene7) | (Gene6 & Gene10 & Gene7)
Gene4 = (Gene10 & !Gene1) | (Gene7 & Gene1)
Gene5 = (Gene6 & !Gene9) | (Gene4 & !Gene9) | (Gene4 & !Gene6)
Gene6 = (!Gene10 & !Gene3) | (!Gene10 & Gene5) | (Gene10 & !Gene5 & Gene3)
Gene7 = (Gene9 & !Gene4) | (Gene9 & Gene7)
Gene8 = (!Gene7 & !Gene9) | (!Gene3) | (Gene7 & Gene9)
Gene9 = (!Gene10 & !Gene5 & Gene7) | (Gene10 & Gene5 & Gene7)
Gene10 = (!Gene4 & !Gene5 & !Gene8) | (Gene5 & Gene8)
| (Gene4 & Gene8) | (Gene4 & Gene5)

Synchronous update: $x^{t+1} = F(x_t)$

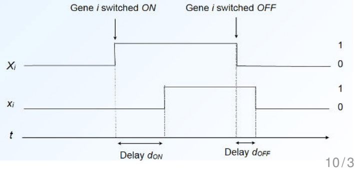


R. Thomas (1973) Journal of Theor. Biol. 42: 563?85.

- Boolean networks to investigate the dynamics of gene networks
- Regulated switch (Lysis *vs* lysogeny) of the bacteriophage λ
- Asynchronous update
- Extension to multi-valued variables

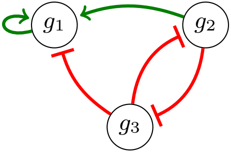
$x_i \longrightarrow$ current level of product of gene i
$X_i \longrightarrow$ whether gene i is currently transcribed

$$X^t = F(x^t)$$



Basics of the logical modelling framework

The Boolean case

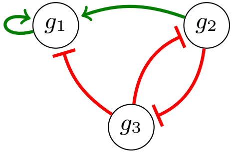


Regulatory graph

- Components (genes, proteins, phenotypes), each g_i associated with a Boolean variable $x_i \in \{0, 1\}$
- Regulatory interactions (+, -, ±)

Basics of the logical modelling framework

The Boolean case



Regulatory graph

- Components (genes, proteins, phenotypes), each g_i associated with a Boolean variable $x_i \in \{0, 1\}$
- Regulatory interactions (+, -, ±)

Regulatory functions

- One Boolean function f_i for each component g_i, defining its evolution
- The ensemble of the regulatory functions defines a **transition function** over the state space

$$f : S = \Pi_{i=1,..n}\{0, 1\} \longrightarrow S,$$
$$f(x) = (f_1(x), f_2(x), \ldots f_n(x))$$

e.g. , g_1 is activated in the presence of g_1 or g_2 and the absence of g_3

Basics of the logical modelling framework

The Boolean case

Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$

g_1 or g_2 and not g_3

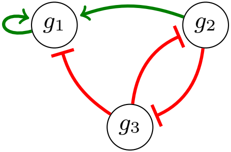
$x_2 = !x_3$

not g_3

$x_3 = !x_2$

not g_2

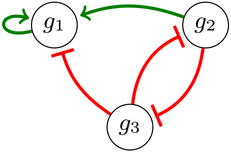
x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$



Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$
g_1 or g_2 and not g_3
$x_2 = !x_3$
not g_3
$x_3 = !x_2$
not g_2

x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	0	1	1



Discrete dynamics - State Transition Graph (STG)

- Nodes are states: $x \in S = \Pi_{i=1,...n}\{0,...max_i\}$ (e.g. expression patterns)
- Directed edges are transitions: updates defined by the regulatory functions f_i

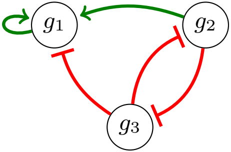
Basics of the logical modelling framework

The Boolean case

Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$
g_1 or g_2 and not g_3
$x_2 = !x_3$
not g_3
$x_3 = !x_2$
not g_2

x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	0	1	1
0	0	1	0	0	1



Discrete dynamics - State Transition Graph (STG)

- Nodes are states: $x \in S = \Pi_{i=1,...n}\{0,...max_i\}$ (e.g. expression patterns)
- Directed edges are transitions: updates defined by the regulatory functions f_i

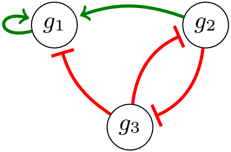
Basics of the logical modelling framework

The Boolean case

Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$
g_1 or g_2 and not g_3
$x_2 =!x_3$
not g_3
$x_3 =!x_2$
not g_2

x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	0



Discrete dynamics - State Transition Graph (STG)

- Nodes are states: $x \in S = \Pi_{i=1,...n}\{0,...max_i\}$ (e.g. expression patterns)
- Directed edges are transitions: updates defined by the regulatory functions f_i

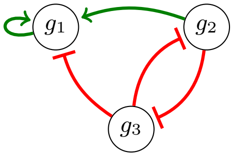
Basics of the logical modelling framework

The Boolean case

Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$
g_1 or g_2 and not g_3
$x_2 = !x_3$
not g_3
$x_3 = !x_2$
not g_2

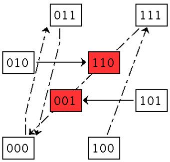
x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	0



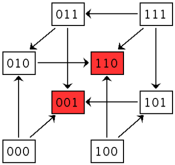
Discrete dynamics - State Transition Graph (STG)

- Nodes are states: $x \in S = \Pi_{i=1,...n}\{0,...max_i\}$ (e.g. expression patterns)
- Directed edges are transitions: updates defined by the regulatory functions f_i

Synchronous update



Asynchronous update



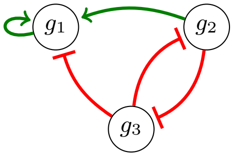
Basics of the logical modelling framework

The Boolean case

Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$
g_1 or g_2 and not g_3
$x_2 =!x_3$
not g_3
$x_3 =!x_2$
not g_2

x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	0



Regulatory functions define the regulatory graph

$\bar{x}^i$ the state differing from x on the sole i^{th} component by ±1

$$(g_i, g_j) \iff \exists x \in S\ s.t.\ f_j(x) = 1 - f_j(\bar{x}^i)$$

∃ a pair of states differing on x_i for which f_j also differs

Basics of the logical modelling framework

The Boolean case

Regulatory functions

$x_1 = (x_1|x_2)\&!(x_3)$

g_1 or g_2 and not g_3

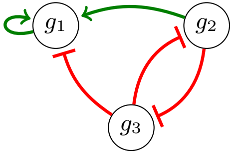
$x_2 = !x_3$

not g_3

$x_3 = !x_2$

not g_2

x_1	x_2	x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	0



Regulatory functions define the regulatory graph

$\bar{x}^i$ the state differing from x on the sole i^{th} component by ±1

$$(g_i, g_j) \iff \exists x \in S\ s.t.\ f_j(x) = 1 - f_j(\bar{x}^i)$$

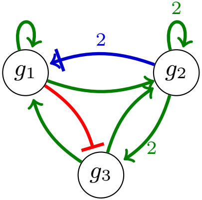
$\exists$ a pair of states differing on x_i for which f_j also differs

Exercice

- what interaction(s) defined by f over states 000 and 010?
- give a pair of states showing that f defines an inhibition from g_3 to g_2

Basics of the logical modelling framework

Extension to multi-valued variables

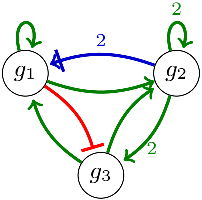


Regulatory graph

- Components (genes, proteins, phenotypes), each g_i associated with a discrete variable $x_i \in \{0, \dots max_i\}$
- Regulatory interactions (+, -, ±) with associated thresholds

Basics of the logical modelling framework

Extension to multi-valued variables



Regulatory graph

- Components (genes, proteins, phenotypes), each g_i associated with a discrete variable $x_i \in \{0, \dots max_i\}$
- Regulatory interactions (+, -, ±) with associated thresholds

Regulatory functions

- One function f_i for each component g_i, defining its evolution
- The ensemble of the regulatory functions defines a **transition** function over the state space

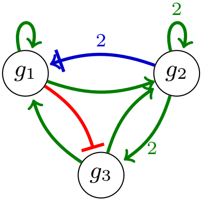
$$f : S = \Pi_{i=1,..n}\{0, \dots max_i\} \longrightarrow S,\ f(x) = (f_1(x), f_2(x), \dots f_n(x))$$

e.g. , g_3 is activated in absence of its repressor and presence of its activator:

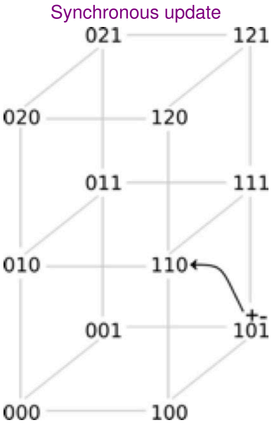
$$f_3(x) = !(x_1)\&(x_2 : 2)$$

Basics of the logical modelling framework

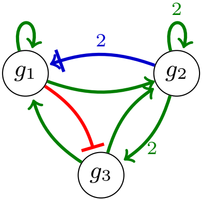
Extension to multi-valued variables



Discrete dynamics - State Transition Graph (STG)

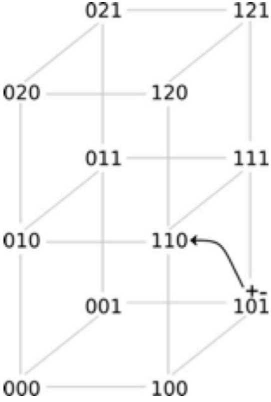


x	$f(x)$
000	000
001	010
010	100
020	021
011	110
021	121
100	110
101	110
120	020
111	121
121	120

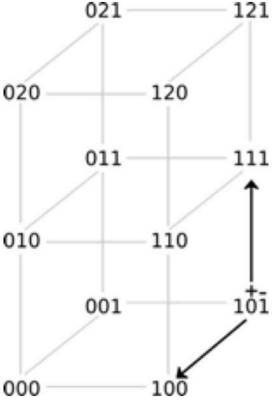


Discrete dynamics - State Transition Graph (STG)

Synchronous update



Asynchronous update



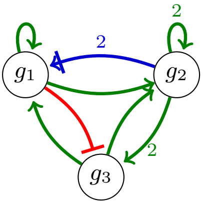
x	$f(x)$
000	000
001	010
010	100
020	021
011	110
021	121
100	110
101	110
120	020
111	121
121	120

Basics of the logical modelling framework

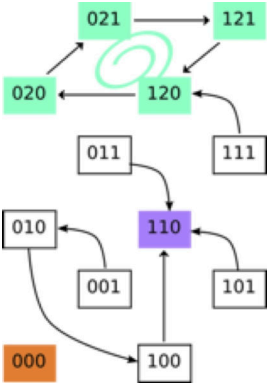
Attractors

Defined by **terminal** Strongly Connected Components (SCCs) in the STG
(maximal set of states for which every state is reachable from every other state)

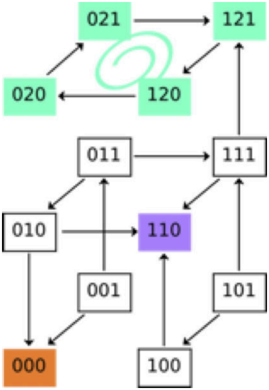
- Stable state (SCC reduced to a single state)
- Complex (or cyclic) attractor (complex SCC, i.e. with several states)



Synchronous update



Asynchronous update

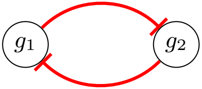


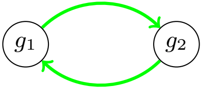
x	$f(x)$
000	000
001	010
010	100
020	021
011	110
021	121
100	110
101	110
120	020
111	121
121	120

Basics of the logical modelling framework

Exercice

- Draw the synchronous & asynchronous STGs for the two component cross-inhibition circuit
- Draw the synchronous & asynchronous STGs for the two component cross-activation circuit
- In general attractors are different in the synchronous and asynchronous STGs. Is this the case for stable states?
- Give two situations for which an interaction between a gene g_1 and a gene g_2 is dual (positive and negative)



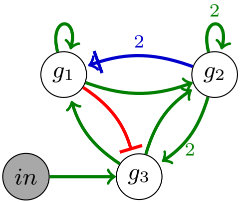


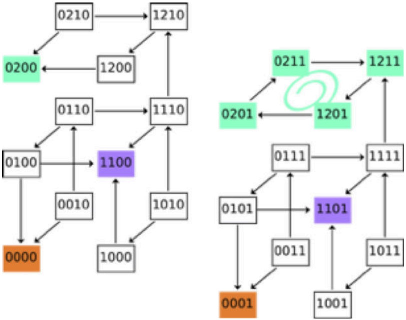
Basics of the logical modelling framework

Input components

- Receptors receiving external signals
- No regulators, hence no logical rules (considered as being constant)

→ disconnected STGs, one for each value of the input





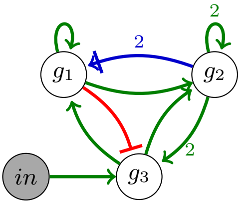
x	$f(x)$	x	$f(x)$
0000	0000	0001	0001
0010	0100	0011	0111
0100	1000	0101	1101
0200	0200	0201	0211
0110	1100	0111	1111
0210	1210	0211	1211
1000	1100	1001	1101
1010	1100	1011	1101
1200	0200	1201	0201
1110	1210	1111	1211
1210	1200	1211	1201

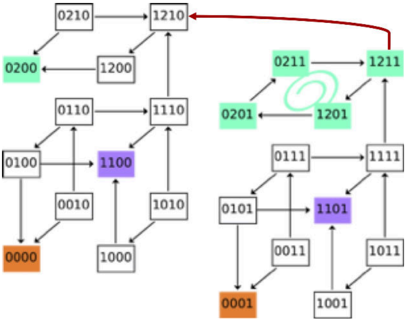
Basics of the logical modelling framework

Input components

- Receptors receiving external signals
- No regulators, hence no logical rules (considered as being constant)

→ disconnected STGs, one for each value of the input

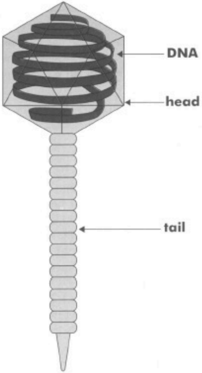


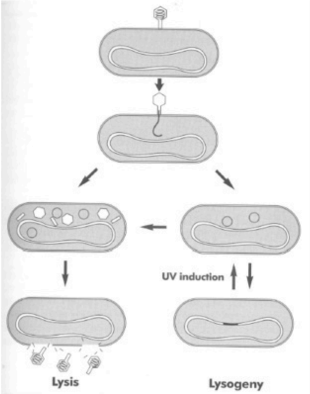


Switching the input value amounts to switching STG

x	$f(x)$	x	$f(x)$
0000	0000	0001	0001
0010	0100	0011	0111
0100	1000	0101	1101
0200	0200	0201	0211
0110	1100	0111	1111
0210	1210	0211	1211
1000	1100	1001	1101
1010	1100	1011	1101
1200	0200	1201	0201
1110	1210	1111	1211
1210	1200	1211	1201

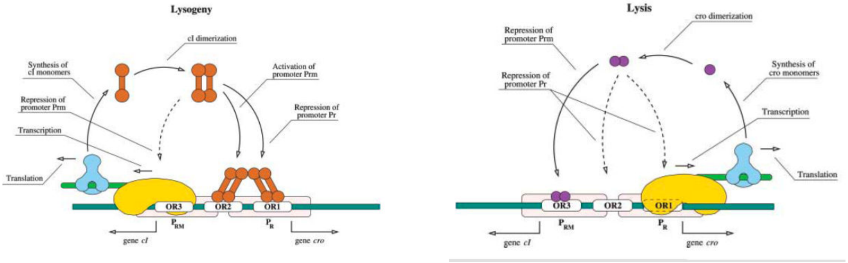
First illustration: the phage λ

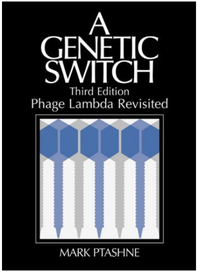




First illustration: the phage λ

Santillan & Mackey (2004) Biophysical J. 86:75-84

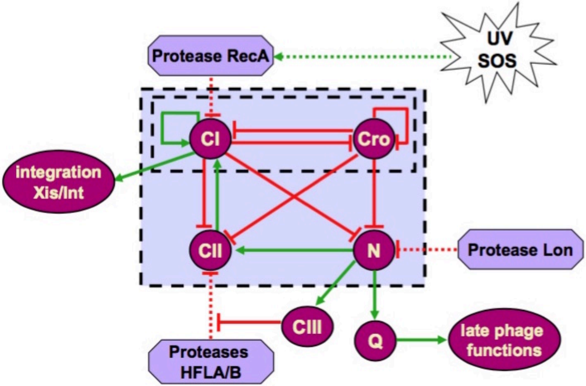




Mark Ptashne, A Genetic Switch, Third Edition, Phage Lambda Revisited. CSHL Press, 2004

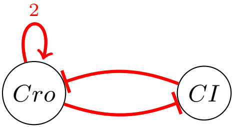
First illustration: the phage λ

Thieffry, D., Thomas, R. (1995) Dynamical behaviour of biological regulatory networks II. Immunity control in bacteriophage lambda. Bull. Math. Biol. 57: 277- 295.



First illustration: the phage λ

A two node model

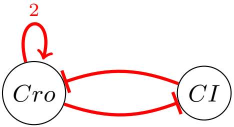


CI	Cro	F_{CI}	F_{Cro}
0	0	1	2
0	1	0	2
0	2	0	1
1	0	1	0
1	1	0	0
1	2	0	0

Exercice: Give the logical expressions for the functions of CI and Cro

First illustration: the phage λ

A two node model

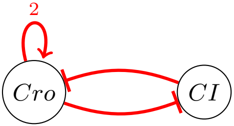


CI	Cro	F_{CI}	F_{Cro}
0	0	1	2
0	1	0	2
0	2	0	1
1	0	1	0
1	1	0	0
1	2	0	0

Exercice: Give the logical expressions for the functions of CI and Cro $\begin{cases} F^1_{CI}(CI, Cro) = !Cro \\ F^1_{Cro}(CI, Cro) = !CI \& Cro{:}2 \\ F^2_{Cro}(CI, Cro) = !CI \& !Cro{:}2 \end{cases}$

First illustration: the phage λ

A two node model



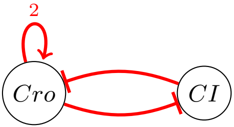
CI	Cro	F_{CI}	F_{Cro}
0	0	1	2
0	1	0	2
0	2	0	1
1	0	1	0
1	1	0	0
1	2	0	0

Exercice: Give the logical expressions for the functions of CI and Cro $\begin{cases} F^1_{CI}(CI, Cro) = !Cro \\ F^1_{Cro}(CI, Cro) = !CI \& Cro{:}2 \\ F^2_{Cro}(CI, Cro) = !CI \& !Cro{:}2 \end{cases}$

Exercice: Draw the asynchronous STG, what are the attractors?

First illustration: the phage λ

A two node model

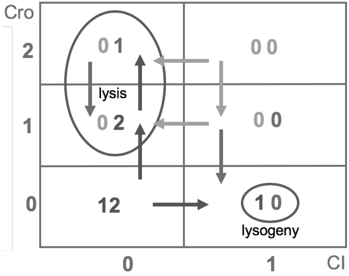


CI	Cro	F_{CI}	F_{Cro}
0	0	1	2
0	1	0	2
0	2	0	1
1	0	1	0
1	1	0	0
1	2	0	0

Exercice: Give the logical expressions for the functions of CI and Cro

$$\begin{cases} F^1_{CI}(CI, Cro) = !Cro \\ F^1_{Cro}(CI, Cro) = !CI \& Cro{:}2 \\ F^2_{Cro}(CI, Cro) = !CI \& !Cro{:}2 \end{cases}$$

Exercice: Draw the asynchronous STG, what are the attractors?

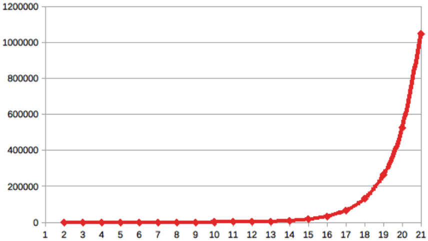


boxes correspond to model states, inside target states are indicated

Handling large networks

Combinatorial explosion of the number of states

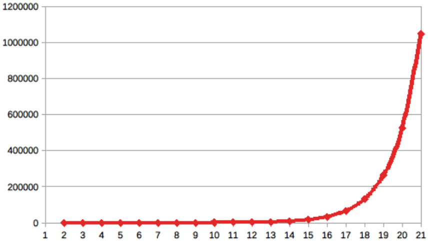
n	2^n
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536
17	131072
18	262144
19	524288
20	1048576



Handling large networks

Combinatorial explosion of the number of states

n	2^n
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536
17	131072
18	262144
19	524288
20	1048576



1sec to handle 1 state ⟶ ∼12 days to handle the state space of a 20 node model!

Handling large networks

Combinatorial explosion of the number of states

Complexity of related algorithms (*e.g.* time complexity)

Size Complexity	10	20	30	40	50	60
n	.00001s	.00002s	.00003s	.00004s	.00005s	.00006s
n^2	.0001s	.0004s	.0009s	.0016s	.0025s	.0036s
n^3	.001s	.008s	.027s	.064s	.125s	.216s
n^5	.1s	3.2s	24.3s	1.7 mn	5.2 mn	13 mn
2^n	.0001s	1.0s	17.9 mn	12.7 days	**35.7 century**	**366 century**
3^n	.059s	58 mn	**6.5 years**	**3855 century**	**$2\mathrm{x}10^8$ century**	**$1.3\mathrm{x}10^{13}$ century**

Assuming 10^6 operations per second

Handling large networks

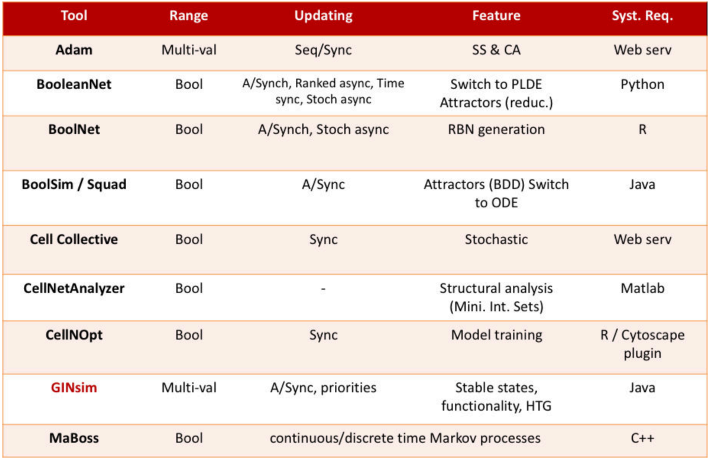
Combinatorial explosion of the number of states
Tool development

- Methods to analyse large STG: attractors, reachability conditions, etc
- Software tools to easy the definition, analysis and simulation

Handling large networks

Combinatorial explosion of the number of states
Tool development

Existing software tools (not exhaustive!))



Tool	Range	Updating	Feature	Syst. Req.
Adam	Multi-val	Seq/Sync	SS & CA	Web serv
BooleanNet	Bool	A/Synch, Ranked async, Time sync, Stoch async	Switch to PLDE Attractors (reduc.)	Python
BoolNet	Bool	A/Synch, Stoch async	RBN generation	R
BoolSim / Squad	Bool	A/Sync	Attractors (BDD) Switch to ODE	Java
Cell Collective	Bool	Sync	Stochastic	Web serv
CellNetAnalyzer	Bool	-	Structural analysis (Mini. Int. Sets)	Matlab
CellNOpt	Bool	Sync	Model training	R / Cytoscape plugin
GINsim	Multi-val	A/Sync, priorities	Stable states, functionality, HTG	Java
MaBoss	Bool	continuous/discrete time Markov processes		C++

Methods to assess crucial properties related to model attractors

Properties derived from the model structure

- Identification of stable states
- Circuit analysis

Methods to assess crucial properties related to model attractors

Properties derived from the model structure

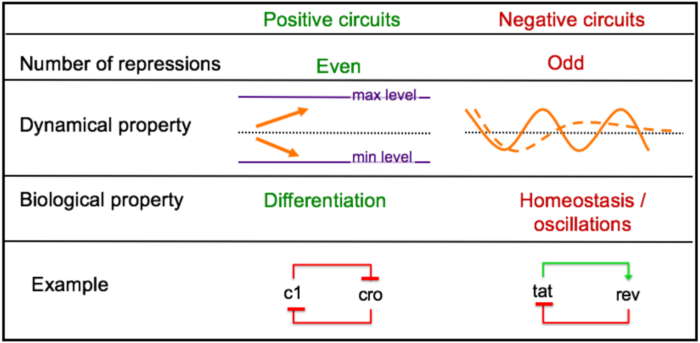
- Identification of stable states
- Circuit analysis

Properties derived from the model dynamics

- Reducing the dynamics
    - Priority classes and mixed updating policies
    - Compact representations of the dynamics
    - Model reduction
- Exploring the dynamics
    - Monte Carlo simulations
    - Model-checking techniques

Methods to assess crucial properties related to model attractors

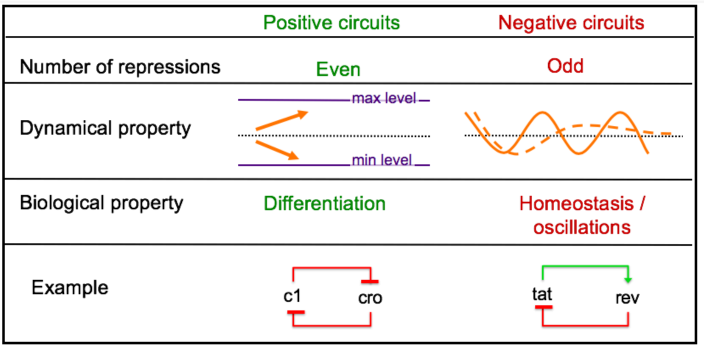
Circuit analysis



	Positive circuits	Negative circuits
Number of repressions	Even	Odd
Dynamical property	max level min level	
Biological property	Differentiation	Homeostasis / oscillations
Example	c1 cro	tat rev

Methods to assess crucial properties related to model attractors

Circuit analysis



	Positive circuits	Negative circuits
Number of repressions	Even	Odd
Dynamical property	max level min level	
Biological property	Differentiation	Homeostasis / oscillations
Example	c1 cro	tat rev

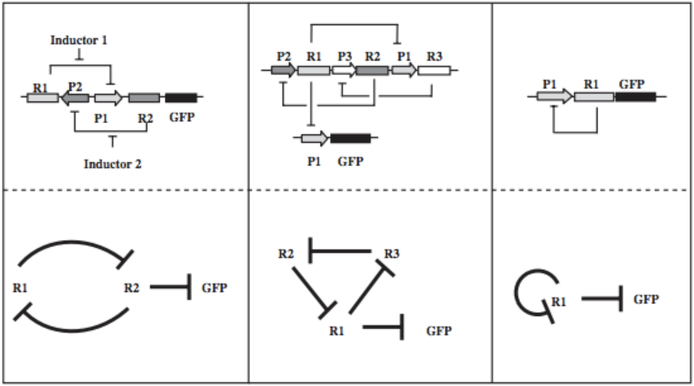
Exercice: How do you define the sign of a circuit??

Methods to assess crucial properties related to model attractors

Circuit analysis

Design of simple regulatory circuits in bacteria

- Two cross-inhibitory genes, giving rise to two alternative stable states and induction memorisation (Gardner TS, Cantor CR, Collins JJ. Construction of a genetic toggle switch in Escherichia coli. Nature 2000;403: 339-42)
- A negative circuit, leading to oscillatory gene expression for proper degradation and synthesis coefficients (Elowitz MB, Leibler S. A synthetic oscillatory network of transcriptional regulators. Nature 2000;403:335-8)
- A self-inhibitory circuit, leading to homeostatic expression of the auto-regulated gene (Becskei A, Serrano L. Engineering stability in gene networks by autoregulation. Nature 2000;405:590-3)



Methods to assess crucial properties related to model attractors

Circuit analysis

R. Thomas' rules

Thomas R (1988). Springer Series in Synergics 9: 180-93

- A positive circuit is necessary to generate multiple attractors
- A negative circuit is necessary to generate maintained oscillations

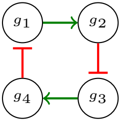
Methods to assess crucial properties related to model attractors

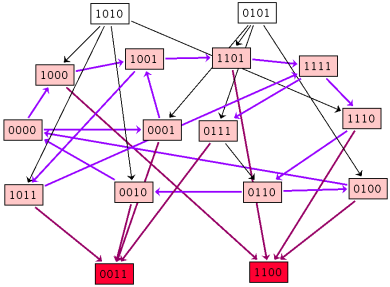
Circuit analysis

R. Thomas' rules

Thomas R (1988). Springer Series in Synergics 9: 180-93

- A positive circuit is necessary to generate multiple attractors
- A negative circuit is necessary to generate maintained oscillations





Two stable states

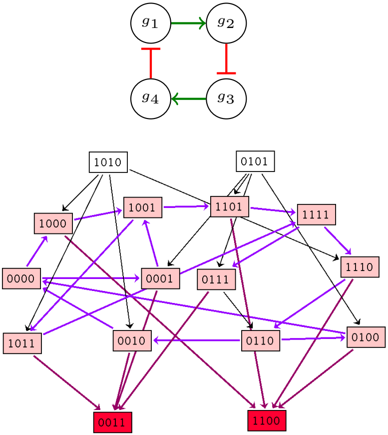
Methods to assess crucial properties related to model attractors

Circuit analysis

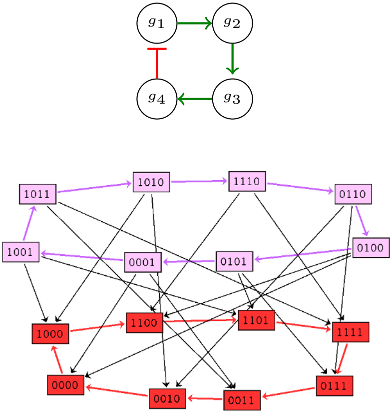
R. Thomas' rules

Thomas R (1988). Springer Series in Synergics 9: 180-93

- A positive circuit is necessary to generate multiple attractors
- A negative circuit is necessary to generate maintained oscillations



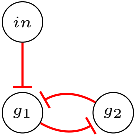
Two stable states

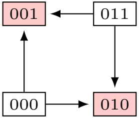


One cyclic attractor

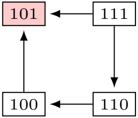
Methods to assess crucial properties related to model attractors

Circuit analysis





$in = 0$

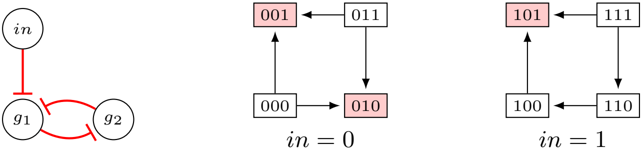


$in = 1$

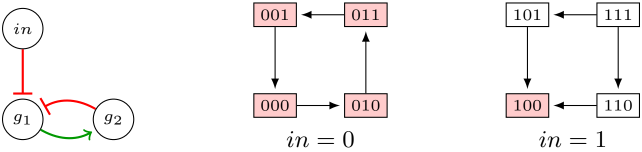
The circuit is **functional** in the absence of in

Methods to assess crucial properties related to model attractors

Circuit analysis



The circuit is **functional** in the absence of in



Functionality context

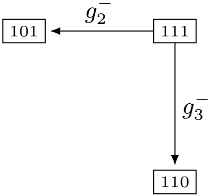
Values of external regulators for which the circuit is functional

$\longrightarrow$ region of the state space in which the circuit generates the expercted behaviour

Methods to assess crucial properties related to model attractors

Priority classes and mixed updating policies

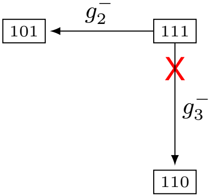
Include information about the delays when available



Methods to assess crucial properties related to model attractors

Priority classes and mixed updating policies

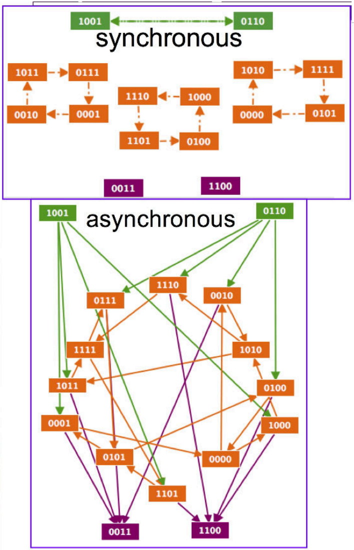
Include information about the delays when available



If the de-activation of g_2 is faster than that of g_3, transition towards 101 will occur

Methods to assess crucial properties related to model attractors

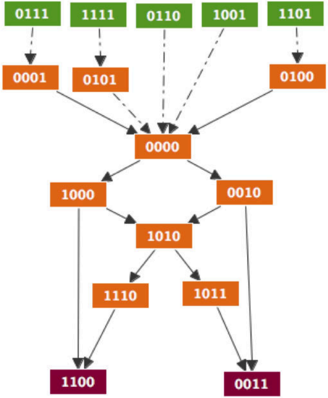
Priority classes and mixed updating policies



Priority classes (an example)

Degradations → **high** priority, synchronous

Syntheses → **low** priority, asynchronous

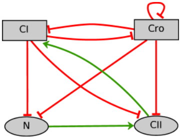


Fauré et al (2006) Bioinformatics 22(14):124-31.c

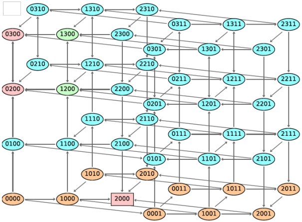
Methods to assess crucial properties related to model attractors

Compact representations of the dynamics

Berenguier (2013)Chaos 23, 025114



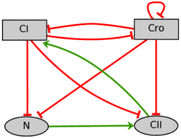
Logical functions	
$K_{CI}(s) = 2$	$\neg Cro \ \vee \ CII$
$K_{Cro}(s) = 3$	$\neg CI{:}2 \ \wedge \ \neg Cro{:}3$
$K_{Cro}(s) = 2$	$\neg CI{:}2 \ \wedge \ Cro{:}3$
$K_{CII}(s) = 1$	$\neg CI{:}2 \ \wedge \ \neg Cro{:}3 \ \wedge \ N$
$K_N(s) = 1$	$\neg CI \ \wedge \ \neg Cro{:}2$


State Transition Graph

Methods to assess crucial properties related to model attractors

Compact representations of the dynamics

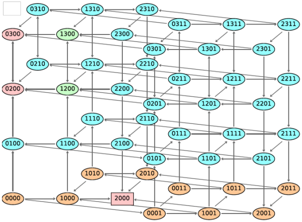
Berenguier (2013)Chaos 23, 025114



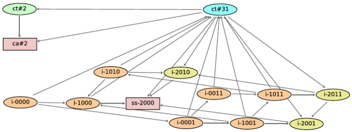
Logical functions

$K_{CI}(s) = 2$	$\neg Cro \;\vee\; CII$
$K_{Cro}(s) = 3$	$\neg CI{:}2 \;\wedge\; \neg Cro{:}3$
$K_{Cro}(s) = 2$	$\neg CI{:}2 \;\wedge\; Cro{:}3$
$K_{CII}(s) = 1$	$\neg CI{:}2 \;\wedge\; \neg Cro{:}3 \;\wedge\; N$
$K_N(s) = 1$	$\neg CI \;\wedge\; \neg Cro{:}2$

State Transition Graph



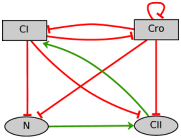
SCC graph



Methods to assess crucial properties related to model attractors

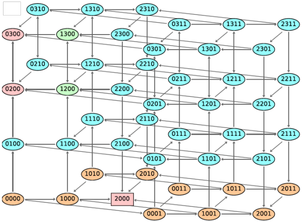
Compact representations of the dynamics

Berenguier (2013)Chaos 23, 025114

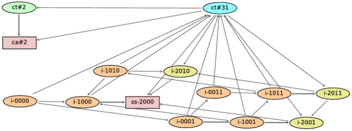


Logical functions	
$K_{CI}(s) = 2$	$\neg Cro \ \vee \ CII$
$K_{Cro}(s) = 3$	$\neg CI{:}2 \ \wedge \ \neg Cro{:}3$
$K_{Cro}(s) = 2$	$\neg CI{:}2 \ \wedge \ Cro{:}3$
$K_{CII}(s) = 1$	$\neg CI{:}2 \ \wedge \ \neg Cro{:}3 \ \wedge \ N$
$K_N(s) = 1$	$\neg CI \ \wedge \ \neg Cro{:}2$

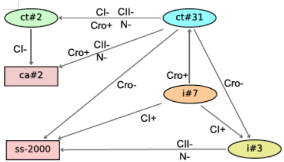
State Transition Graph



SCC graph



HTG graph



Methods to assess crucial properties related to model attractors

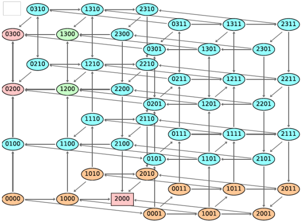
Compact representations of the dynamics

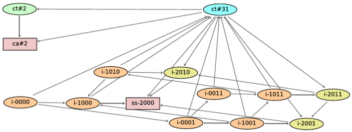
Berenguier (2013)Chaos 23, 025114

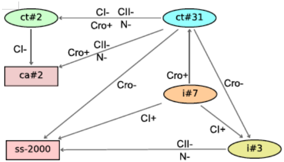
Hierarchical Transition Graph

Merges in a single node, states that are:

- irreversible (SCC reduced to a single state) and lead to the same complex SCC or attractor
- in the same complex SCC


State Transition Graph

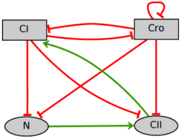

SCC graph

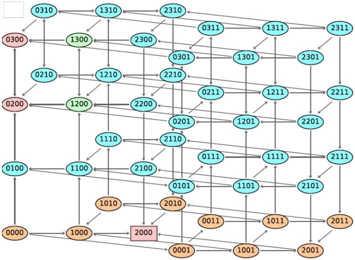

HTG graph

Methods to assess crucial properties related to model attractors

Monte Carlo simulations

Mendes *et al.*(2018) Front. Physiol., 9, pp. 1161



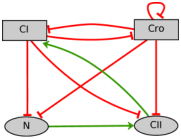


Monte Carlo simulation: repeated random sampling of the trajectories ⟶ reachability quantification (number of trajectories leading to each attractor)

Methods to assess crucial properties related to model attractors

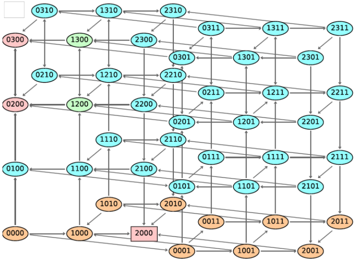
Monte Carlo simulations

Mendes *et al.*(2018) Front. Physiol., 9, pp. 1161



Monte Carlo simulation: repeated random sampling of the trajectories ⟶ reachability quantification (number of trajectories leading to each attractor)

Nodes=[CI,Cro,CII,N]
Initial conditions [0000]
Time=1.339s
Successful runs=1000
Stable states: SS1=> [2000] prob=0.356
Complex attractors: CA1=> [0200][0300] prob=0.644 size=2
Transient found: #31 states



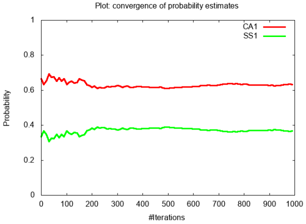


Illustration: eukaryotic cell cycle control

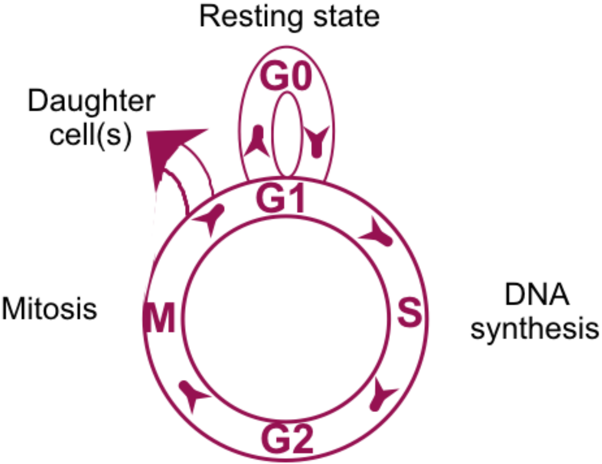
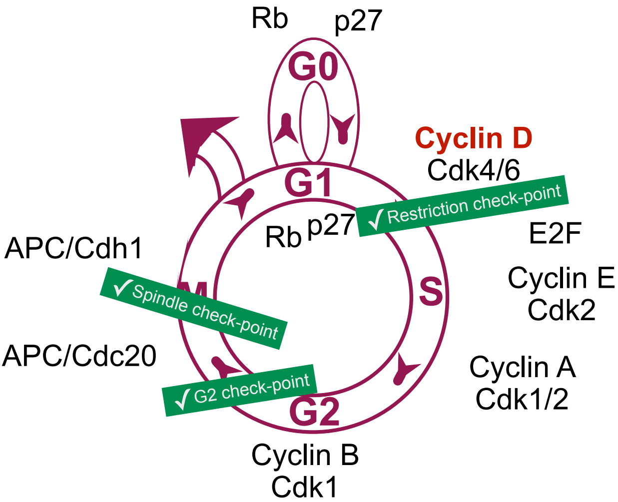


Illustration: eukaryotic cell cycle control



Rb p27

G0

Cyclin D
Cdk4/6

G1

✓Restriction check-point

Rb p27

E2F

APC/Cdh1

M

✓Spindle check-point

S

Cyclin E
Cdk2

APC/Cdc20

Cyclin A
Cdk1/2

✓G2 check-point

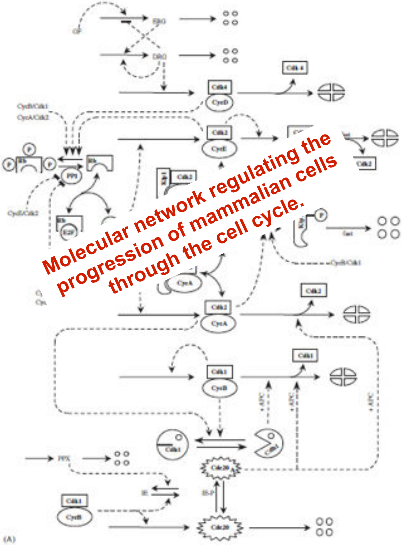
G2

Cyclin B
Cdk1

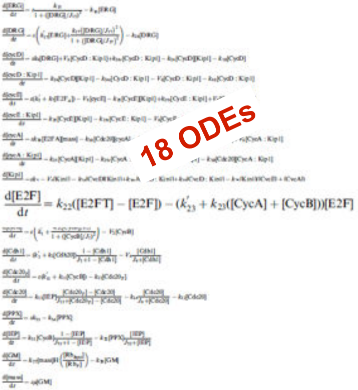
Illustration: eukaryotic cell cycle control

B. Novak, J.J. Tyson. (2004) J Theo Biol 230, 563-79

A model for restriction point control of the mammalian cell cycle.

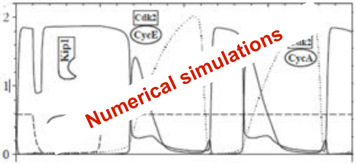


Molecular network regulating the progression of mammalian cells through the cell cycle.



$$\frac{d[E2F]}{dt} = k_{22}([E2FT]-[E2F]) - (k'_{23}+k_{23}([CycA]+[CycB]))[E2F]$$

18 ODEs

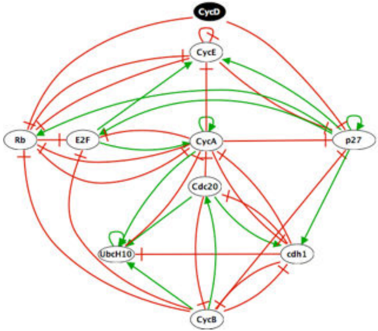


Numerical simulations

Illustration: eukaryotic cell cycle control

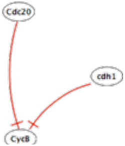
A. Fauré et al (2006) Bioinformatics, 22(14) 134-31

Regulatory graph



10 logical rules

$CycD$	$CycD$
Rb	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{CycD} \wedge \overline{CycB})$
E2F	$(\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{Rb} \wedge \overline{CycB})$
CycE	$(E2F \wedge \overline{Rb})$
CycA	$(E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge Ubc)}) \vee (CycA \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge Ubc)})$
p27	$(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{(CycE \wedge CycA)} \wedge \overline{CycB} \wedge \overline{CycD})$
Cdc20	$CycB$
Cdh1	$(\overline{CycA1} \wedge \overline{CycB}) \vee (Cdc20) \vee (p27 \wedge \overline{CycB})$
UbcH10	$(\overline{Cdh1}) \vee (Cdh1 \wedge Ubc \wedge (Cdc20 \vee CycA \vee CycB))$
CycB	$(\overline{Cdc20} \wedge \overline{Cdh1})$



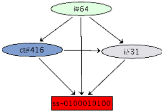
CycB is active in the absence of both Cdc20 and Cdh1

Illustration: eukaryotic cell cycle control

A. Fauré et al (2006) Bioinformatics, 22(14) 134-31

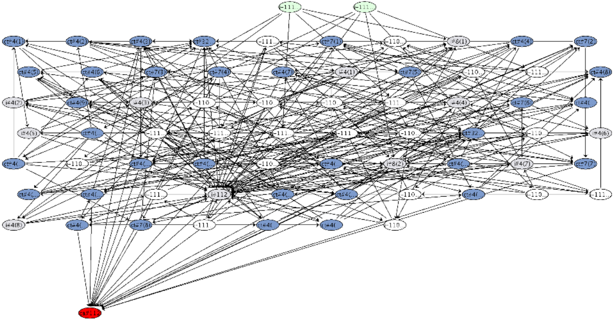
HTG, asynchronous dynamics

CycD=0



Stable state

CycD=1



Cyclical attractor

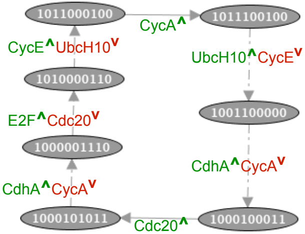
112 states

Illustration: eukaryotic cell cycle control

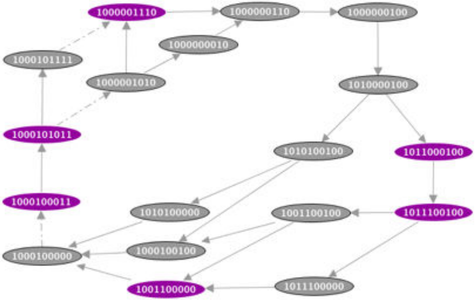
A. Fauré et al (2006) Bioinformatics, 22(14) 134-31

Cyclical attractor: does it fit the observed oscillations?

Synchronous update



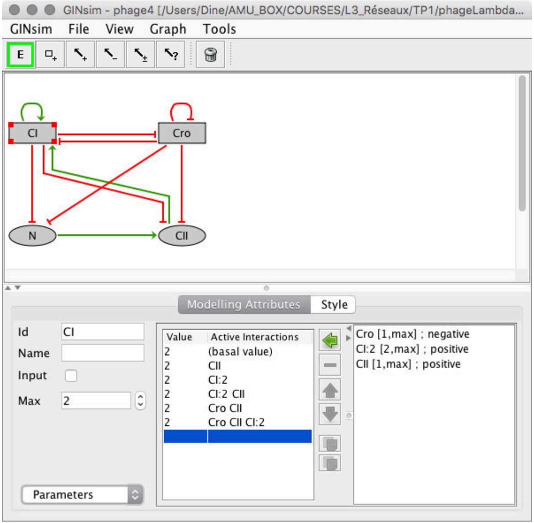
With priorities



Rank	Type	Transitions
1	asynchronous	CycD, Rb, E2F-, CycE-, p27Kip1, Cdh1
1	synchronous	CycA-, Cdc20-, UbcH10-, CycB-
2	asynchronous	E2F+, CycE+, CycA+, Cdc20+
2	synchronous	UbcH10+, CycB+

Practical session

Definition and analysis of Thieffry & Thomas' phage lambda model using the software GINsim http://ginsim.org



Id: CI

Name:

Input: ☐

Max: 2

Value	Active Interactions
2	(basal value)
2	CII
2	CI:2
2	CI:2 CII
2	Cro CII
2	Cro CII CI:2

Cro [1,max] ; negative
CI:2 [2,max] ; positive
CII [1,max] ; positive

Further reading

- W. Abou-Jaoudé *et al* (2016) Logical modeling and dynamical analysis of cellular networks, Frontiers in Genetics, Vol. 7, pp 94
  https://www.frontiersin.org/articles/10.3389/fgene.2016.00094/full
- A. Naldi *et al.* (2018) Logical modelling and analysis of cellular regulatory networks with GINsim 3.0, Front. Physiol., 9, pp. 646
  http://dx.doi.org/10.3389/fphys.2018.00646
- Thieffry D, Thomas R. (1995) Dynamical behaviour of biological regulatory networks–II. Immunity control in bacteriophage lambda. Bull Math Biol. 57(2):277-97.
  https://link.springer.com/article/10.1007%2FBF02460619?LI=true
- A. Fauré *et al.* (2006) Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle Bioinformatics, Vol, 22, Issue 14, pp:e124-31 https://doi.org/10.1093/bioinformatics/btl210