

Logical modelling of regulatory networks: boolSim & Squad



Swiss Institute of
Bioinformatics

Introduction

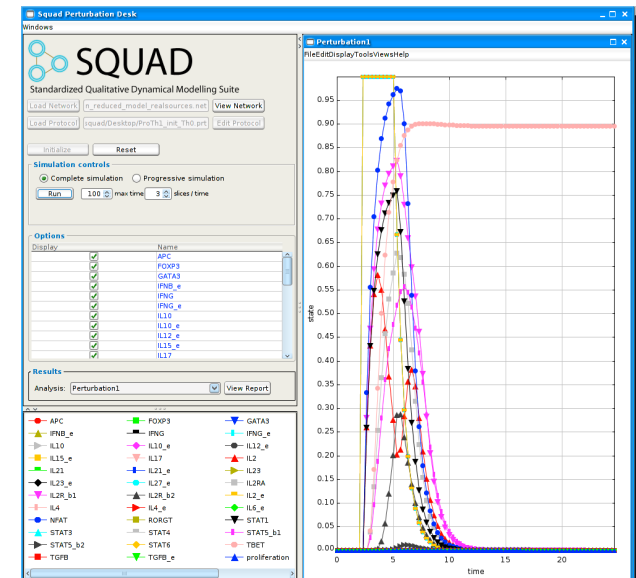
Boolsim (genYsis)¹

- Boolean networks.
- Command line tool.
- Features:
 - Perturbations.
 - Attractors (All!).
 - Reachability graphs.
 - Very efficient!

```
$ boolSim -f Th_Naldi_etal_2010_reduced.net -e nostimulation_proTh1.exp
parsing network file
geneDbOrg=36 geneDbReduced=25
processing experiment file level 1
-- reachability --
SS_1_1 ----> SS_2_7
SS_1_2 ----> SS_2_7
SS_1_3 ----> SS_2_7
SS_1_4 ----> SS_2_7
SS_1_5 ----> SS_2_5
SS_1_6 ----> SS_2_5
Results are written in the file 'results/reach.txt'
```

SQUAD²

- Standardized **QUAL**itative **DY**namical systems
- Continuous models derived from Boolean networks.
- Graphical user interface.
- Features:
 - Time evolution from user defined initial state.
 - Perturbations: time range pulse.



¹ A. Garg, A. Di Cara, I. Xenarios, L. Mendoza & G. De Micheli, Bioinformatics (2008).

² A. Di Cara, A. Garg, G. De Micheli, I. Xenarios & L. Mendoza, BMC Bioinformatics (2007).

Boolean networks: file format

Activators & inhibitors

Source -> Target



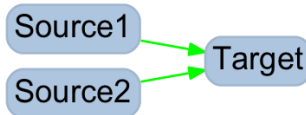
Source -| Target



Multiple activators are combined with OR:

Source1 -> Target

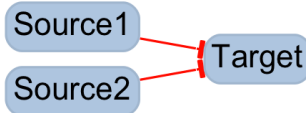
Source2 -> Target



Multiple inhibitors are combined with OR:

Source1 -| Target

Source2 -| Target



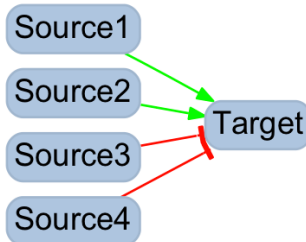
Inhibitors are “strong”:

Source1 -> Target

Source2 -> Target

Source3 -| Target

Source4 -| Target



\vee = OR

\wedge = AND

\neg = NOT

$$X'_{\text{Target}} = X_{\text{Source}}$$

$$X'_{\text{Target}} = \neg X_{\text{Source}}$$

$$X'_T = X_{S1} \vee X_{S2}$$

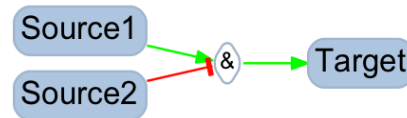
$$X'_T = \neg(X_{S1} \vee X_{S2})$$

$$X'_T = (X_{S1} \vee X_{S2}) \wedge \neg(X_{S3} \vee X_{S4})$$

Target = 1 \Leftrightarrow at least one activator = 1 and all inhibitors = 0.

Source can be a Boolean expression
(boolSim notation: &=AND, ^=NOT)

Source1 & ^ Source2 -> Target

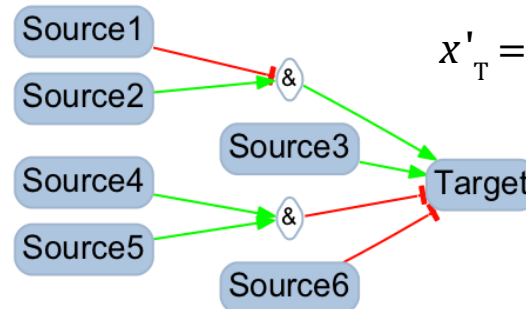


$$x'_T = x_{S1} \wedge \neg x_{S2}$$

∨ = OR
^ = AND
¬ = NOT

Example: multiple activators and inhibitors

^ Source1 & Source2 -> Target
Source3 -> Target
Source4 & Source5 - | Target
Source6 - | Target



$$x'_T = (\neg x_{S1} \wedge x_{S2} \vee x_{S3}) \wedge \neg (x_{S4} \wedge x_{S5} \vee x_{S6})$$

Notes:

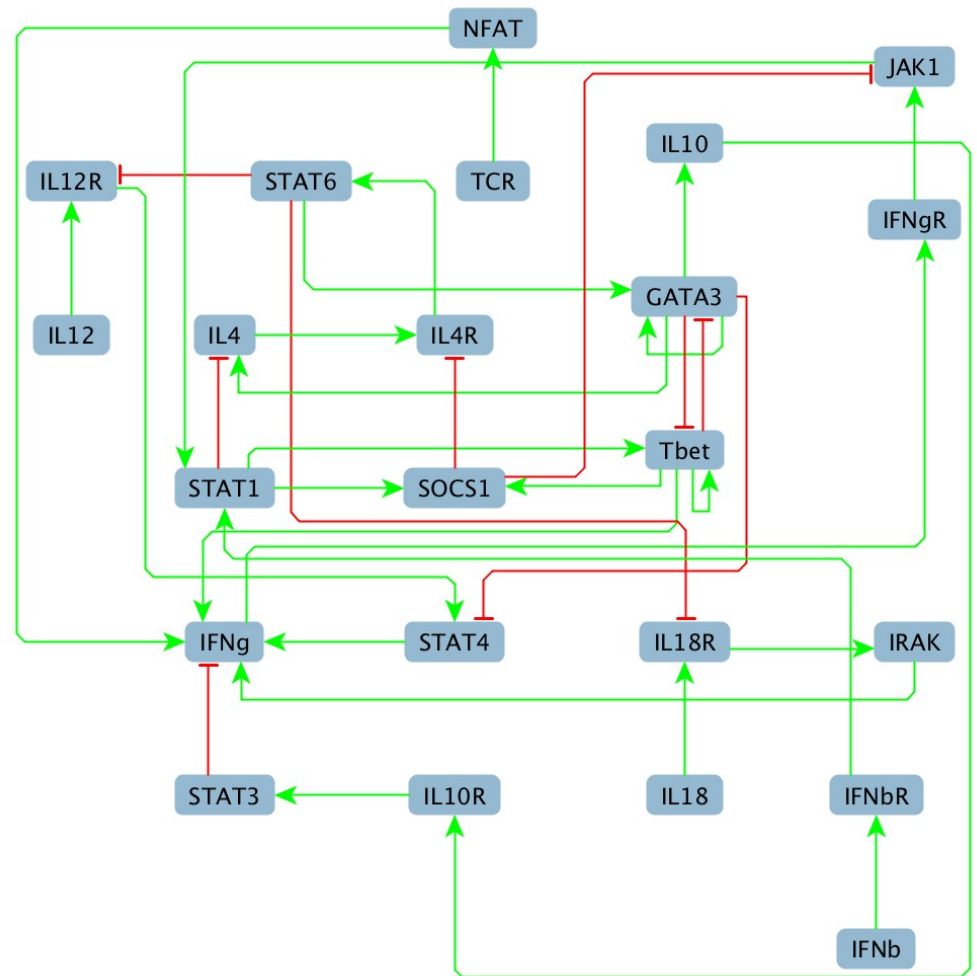
- Default node state = 0
⇒ State of nodes without input edge is always 0.
- Space must be used only to separate the interaction type (-> or - |) from source and target. Not allowed in node names, around & or around ^.

Example: T-helper model, Mendoza & Xenarios (2006)

```

GATA3 → GATA3
STAT6 → GATA3
Tbet →| GATA3
IFNb → IFNbR
IRAK → IFNg
NFAT → IFNg
STAT3 →| IFNg
STAT4 → IFNg
Tbet → IFNg
IFNg → IFNgR
GATA3 → IL10
IL10 → IL10R
IL12 → IL12R
STAT6 →| IL12R
IL18 → IL18R
STAT6 →| IL18R
GATA3 → IL4
STAT1 →| IL4
IL4 → IL4R
SOCS1 →| IL4R
IL18R → IRAK
IFNgR → JAK1
SOCS1 →| JAK1
TCR → NFAT
STAT1 → SOCS1
Tbet → SOCS1
IFNbR → STAT1
JAK1 → STAT1
IL10R → STAT3
GATA3 →| STAT4
IL12R → STAT4
IL4R → STAT6
GATA3 →| Tbet
STAT1 → Tbet
STAT4 → Tbet
IL18R → Tbet
IFNbR → Tbet

```



boolSim: attractors

Using boolSim to find attractors

```
$ boolSim -f Th_Mendoza_Xenarios_2006.net -p 3 -o results/attractors
```

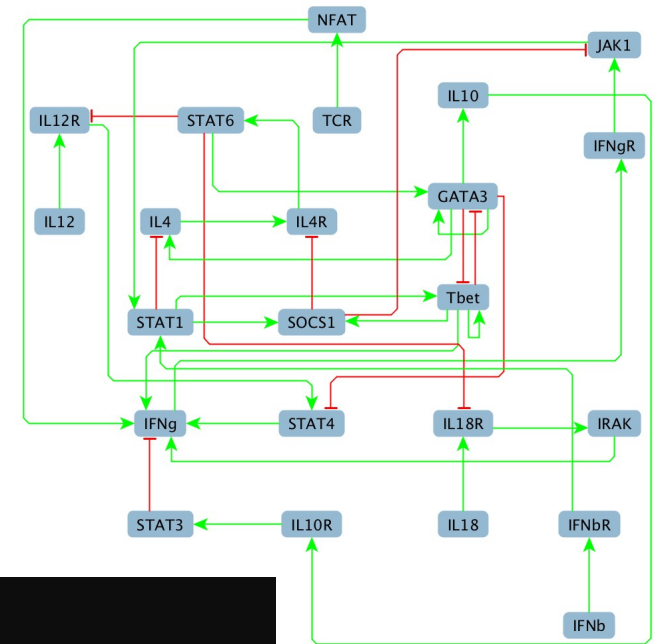
Arguments:

- -f network file
- -p network dynamics
 - p 1 → synchronous
 - p 2 → asynchronous (slow, do not use)
 - p 3 → asynchronous (fast)
- -o output

Output:

```
parsing network file
geneDbOrg=23 geneDbReduced=13
Attractor 1 ::: number of states = 1
Attractor 2 ::: number of states = 1
Attractor 3 ::: number of states = 1
```

```
States of the attractor 1 are written in the file results/attractors_1
States of the attractor 2 are written in the file results/attractors_2
States of the attractor 3 are written in the file results/attractors_3
```



T-helper model
Mendoza & Xenarios (2006)

Attractors are saved in separate files:

```
$ cat results/attractors_1
Gene Name/State No.   S_1
GATA3                  1
STAT6                  1
Tbet                   0
IFNbR                  0
IFNb                   0
IFNg                   0
IRAK                   0
NFAT                   0
STAT3                  1
STAT4                  0
IFNgR                  0
IL10                   1
IL10R                  1
IL12R                  0
IL12                   0
IL18R                  0
IL18                   0
IL4                    1
STAT1                  0
IL4R                   1
SOCS1                  0
JAK1                   0
TCR                    0
```

Th2

```
$ cat results/attractors_2
Gene Name/State No.   S_1
GATA3                  0
STAT6                  0
Tbet                   1
IFNbR                  0
IFNb                   0
IFNg                   1
IRAK                   0
NFAT                   0
STAT3                  0
STAT4                  0
IFNgR                  1
IL10                   0
IL10R                  0
IL12R                  0
IL12                   0
IL18R                  0
IL18                   0
IL4                    0
STAT1                  0
IL4R                   0
SOCS1                  1
JAK1                   0
TCR                    0
```

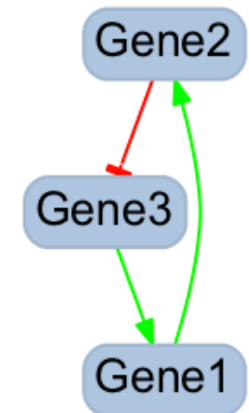
Th1

```
$ cat results/attractors_3
Gene Name/State No.   S_1
GATA3                  0
STAT6                  0
Tbet                   0
IFNbR                  0
IFNb                   0
IFNg                   0
IRAK                   0
NFAT                   0
STAT3                  0
STAT4                  0
IFNgR                  0
IL10                   0
IL10R                  0
IL12R                  0
IL12                   0
IL18R                  0
IL18                   0
IL4                    0
STAT1                  0
IL4R                   0
SOCS1                  0
JAK1                   0
TCR                    0
```

Th0

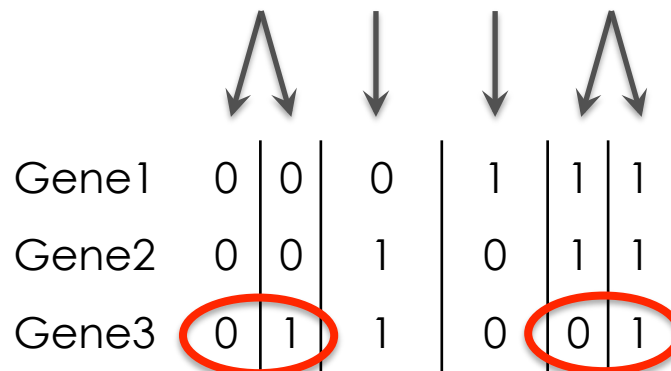
Attractors with multiple states:

```
boolSim -f ToyModel.net -p 3 -o results/attractors
parsing network file
geneDbOrg=3 geneDbReduced=3
Attractor 1 ::: number of states = 6
States of the attractor 1 are written in the file results/attractors_1
```



Compressed output:

```
$ cat results/attractors_1
Gene Name/State No.  S_1  S_2  S_3  S_4
Gene2                0    0    1    1
Gene1                0    1    0    1
Gene3                2    1    0    2
```

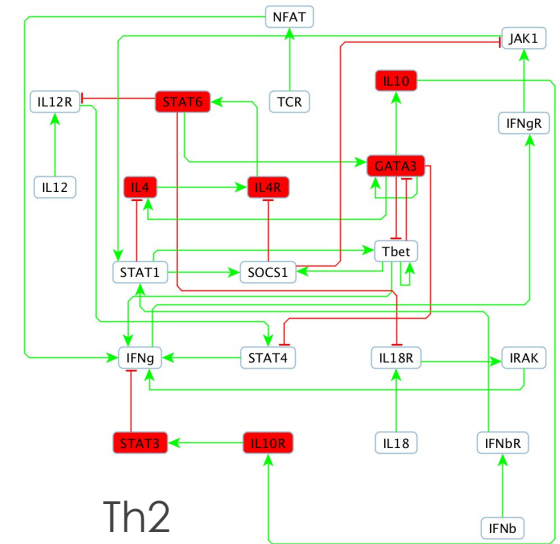
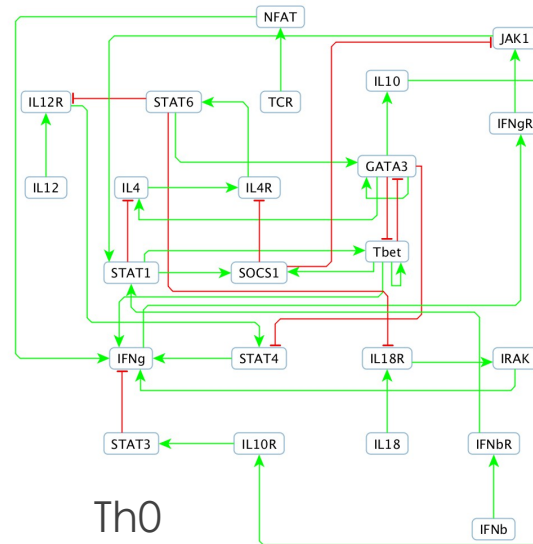
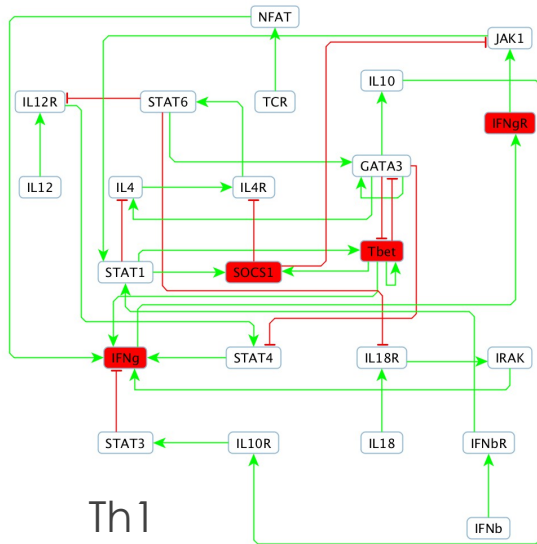


```
Gene Name/State No.  S_1
Gene2                0
Gene1                2
Gene3                2
```



boolSim: perturbation experiments

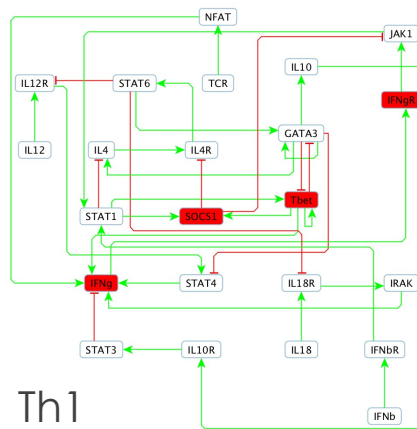
Unperturbed network has three attractors:



What happens if a node state is fixed to 0 or to 1?

E.g. IL4=1 ?

Th1 and Th2 stable. But Th0 ?



Th0

[illegible][illegible][illegible]

Perturbation experiments in boolSim:

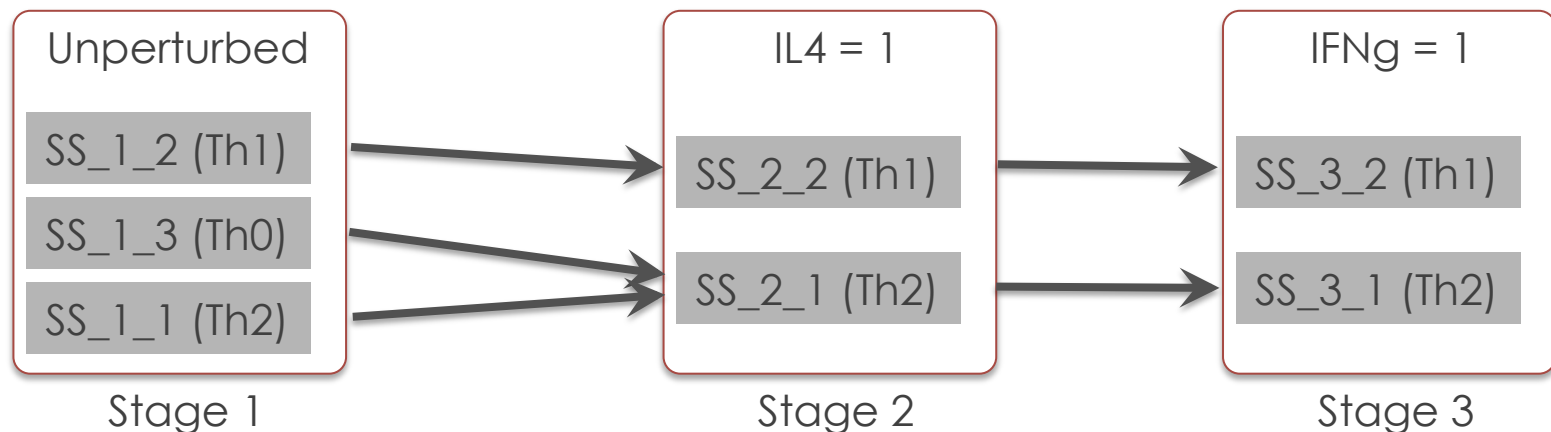
- Multiple stages.
- At each stage, a combination of nodes perturbations can be defined
 - Node state fixed to 0
 - Node state fixed to 1
 - Node state not constrained

Note:

Nodes perturbations stay active for all following stages.

Output:

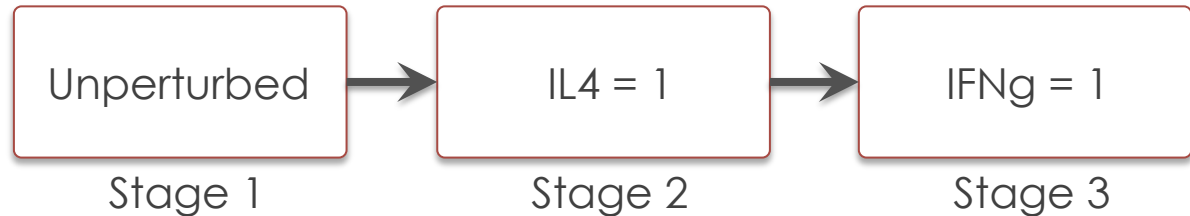
- Asynchronous attractors for each perturbation.
- Transitions between attractors in consecutive stages.



Experiment file format:

Stage 1
 Nb stages
 Nb0 Nb1 NbUnconstrained
 Nodes0
 ...
 Nodes1
 ...
 NodesUnconstrained
 ...
 Stage 2
 Nb0 Nb1 NbUnconstrained
 Nodes0
 ...
 Nodes1
 ...
 NodesUnconstrained
 ...
 Stage 3
 Nb0 Nb1 NbUnconstrained
 Nodes0
 ...
 Nodes1
 ...
 NodesUnconstrained
 ...
 ...

Example:



3

3 stages

0 0 0

Stage 1:

0 node fixed to 0, 0 node fixed to 1, 0 unconstrained node

0 1 0

Stage 2:

0 node fixed to 0, 1 node fixed to 1, 0 unconstrained node

IL4

IL4 is fixed to state 1

0 1 1

Stage 3:

0 node fixed to 0, 1 node fixed to 1, 1 unconstrained node

IFNg

IFNg is fixed to state 1

IL4

IL4 is unconstrained

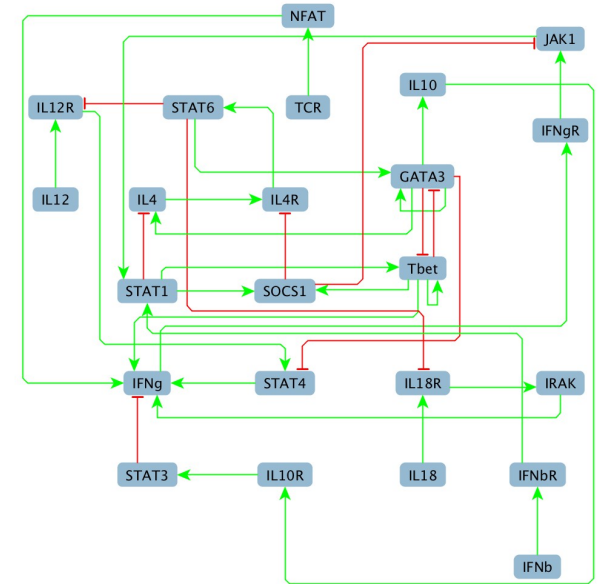
Preparation:

```
$ mkdir results/
```

```
$ cat Th_Mendoza_Xenarios_2006_experiment.exp
3
0 0 0
0 1 0
IL4
0 1 1
IFNg
IL4
```

Running boolsim (-f network -e experiments):

```
$ boolSim -f Th_Mendoza_Xenarios_2006.net -e Th_Mendoza_Xenarios_2006_experiment.exp
parsing network file
geneDbOrg=23 geneDbReduced=13
processing experiment file level 1
-- reachability --
SS_1_1 ----> SS_2_1
SS_1_2 ----> SS_2_2
SS_1_3 ----> SS_2_1
processing experiment file level 2
-- reachability --
SS_2_1 ----> SS_3_1
SS_2_2 ----> SS_3_2
Results are written in the file 'results/reach_.txt'
```



Output:

```
$ cat results/reach_.txt
#####

##### unperturbed network #####

States of the attractor 1 are written in the file results/_SS_1_1.txt
States of the attractor 2 are written in the file results/_SS_1_2.txt
States of the attractor 3 are written in the file results/_SS_1_3.txt
#####

##### IL4, over-expressed #####

States of the attractor 1 are written in the file results/_SS_2_1.txt
States of the attractor 2 are written in the file results/_SS_2_2.txt
**** Reachability Analysis ****
SS_1_1 ----> SS_2_1
SS_1_2 ----> SS_2_2
SS_1_3 ----> SS_2_1
#####

##### IFNg, over-expressed #####

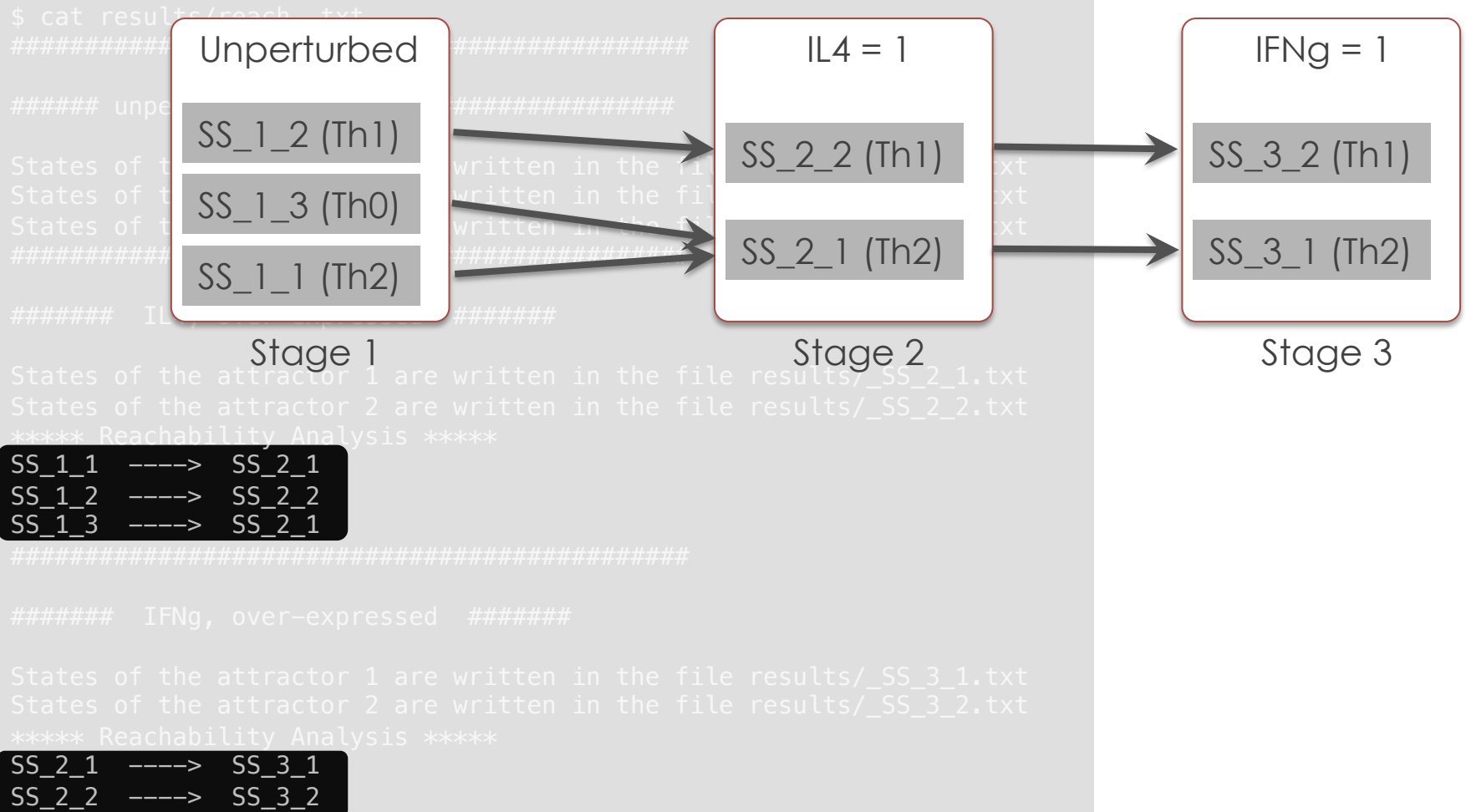
States of the attractor 1 are written in the file results/_SS_3_1.txt
States of the attractor 2 are written in the file results/_SS_3_2.txt
**** Reachability Analysis ****
SS_2_1 ----> SS_3_1
SS_2_2 ----> SS_3_2
```

Stage 1

Stage 2

Stage 3

Output:



Experiment file format: another example

Stage 1	3
	2 1 0
	IL4
	IFNg
Stage 2	IL10
	0 0 1
Stage 3	IL10
	0 1 2
	TCR
	IFNg
	IL4

3 stages

Stage 1: 2 nodes fixed to 0, 1 node fixed to 1, 0 unconstrained node

IL4 is fixed to state 0

IFNg is fixed to state 0

IL10 is fixed to state 1

Stage 2: 0 node fixed to 0, 0 node fixed to 1, 1 unconstrained node

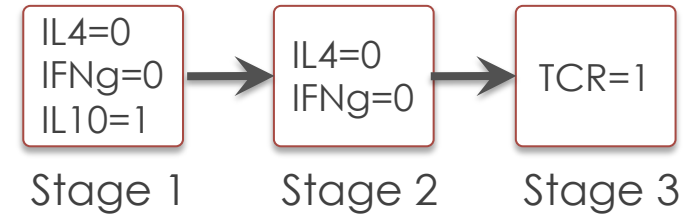
IL10 is unconstrained (IL4 and IFNg are still fixed to state 0)

Stage 3: 0 node fixed to 0, 1 node fixed to 1, 2 unconstrained node

TCR is fixed to state 1

IFNg is unconstrained

IL4 is unconstrained



boolSim & multivalued networks

boolSim accepts only Boolean networks!

BUT:
GINsim can be used to convert a multivalued network to a Boolean network¹.

Node X with values in $\{0, 1, \dots, n\}$ replaced by n Boolean nodes $X_{b1}, X_{b2}, \dots, X_{bn}$ with the mapping:

X	\leftrightarrow	X_{b1}	X_{b2}	...	X_{bn}
0	\leftrightarrow	0	0	...	0
1	\leftrightarrow	1	0	...	0
2	\leftrightarrow	1	1	...	0
...
n	\leftrightarrow	1	1	...	1

Example:

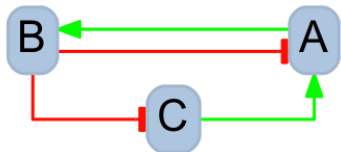
STAT5	\leftrightarrow	STAT5_b1	STAT5_b2
0	\leftrightarrow	0	0
1	\leftrightarrow	1	0
2	\leftrightarrow	1	1

¹G. Didier, E. Remy & C. Chaouiya. Journal of theoretical biology 270, 177 (2011).

boolSim: under the hood

Reduced ordered binary decision diagrams (ROBDD)

Boolean equations:

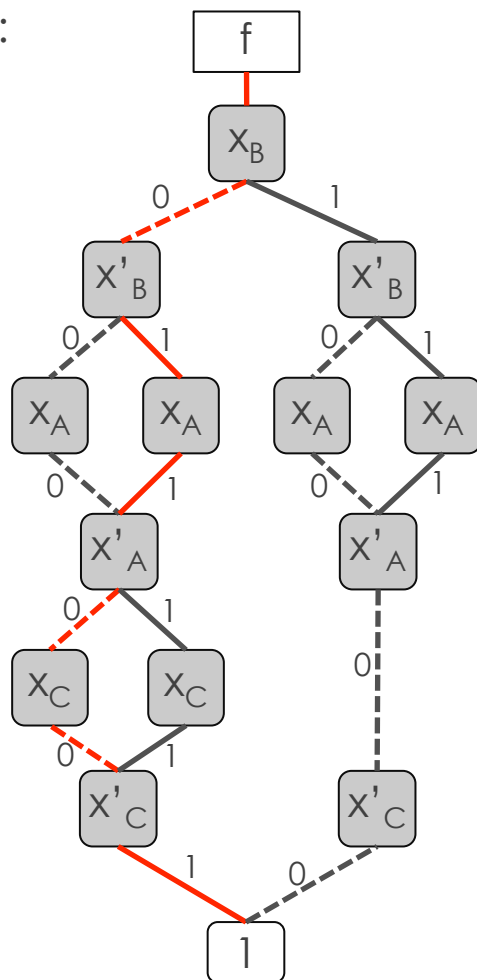


$$x_A' = x_C \wedge \neg x_B$$

$$x_B' = x_A$$

$$x_C' = \neg x_B$$

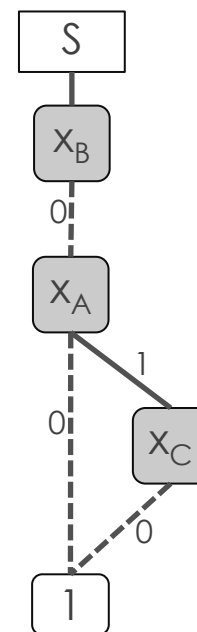
$$\begin{aligned} x_B &= 0 \\ x_B' &= 1 \\ x_A &= 1 \\ x_A' &= 0 \\ x_C &= 0 \\ x_C' &= 1 \end{aligned}$$

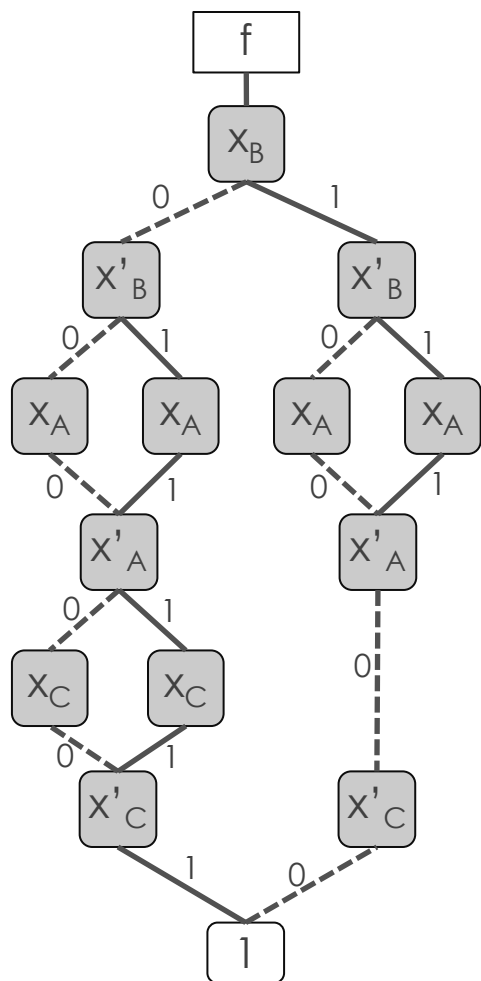


Sets of states:

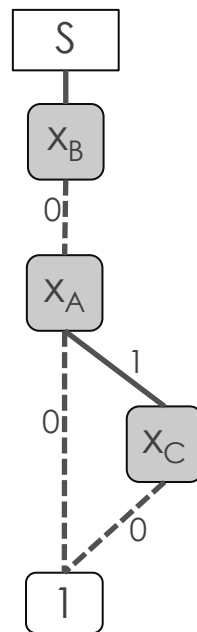
$$\{(0,0,0), (0,0,1), (1,0,0)\}$$

$$\neg x_A \wedge \neg x_B \vee x_A \wedge \neg x_B \wedge \neg x_C$$

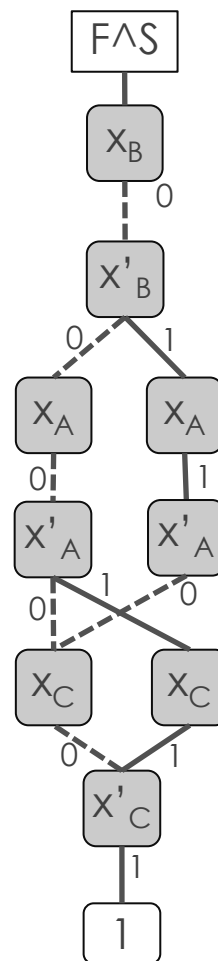




AND



=



x_A	x_B	x_C		x'_A	x'_B	x'_C
0	0	0	→	0	0	1
0	0	1	→	1	0	1
1	0	0	→	0	1	1

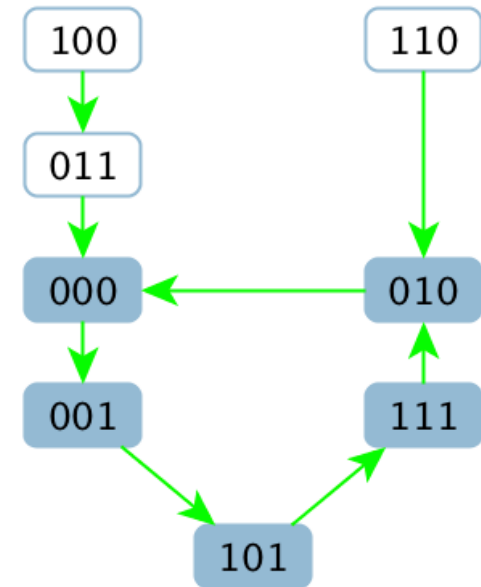
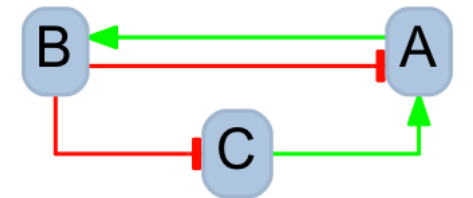
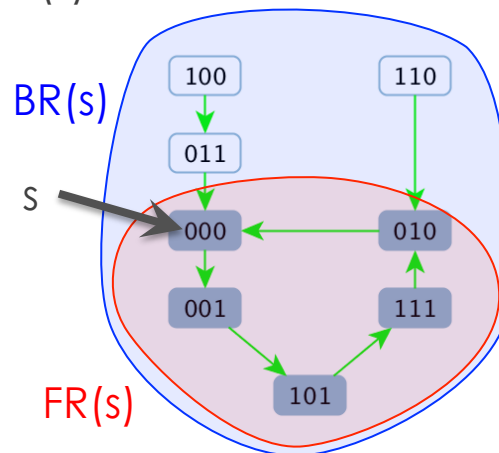
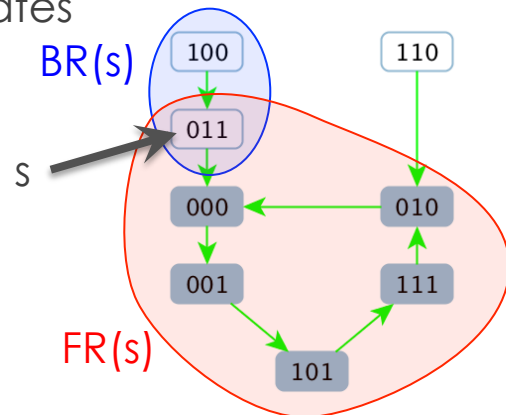
FR(s)=Set of forward reachable states
 BR(s)=Set of backward reachable states

Theorem¹:

A state s is a part of an attractor if and only if $FR(s) \subseteq BR(s)$.
 State s is transient otherwise.

Theorem¹:

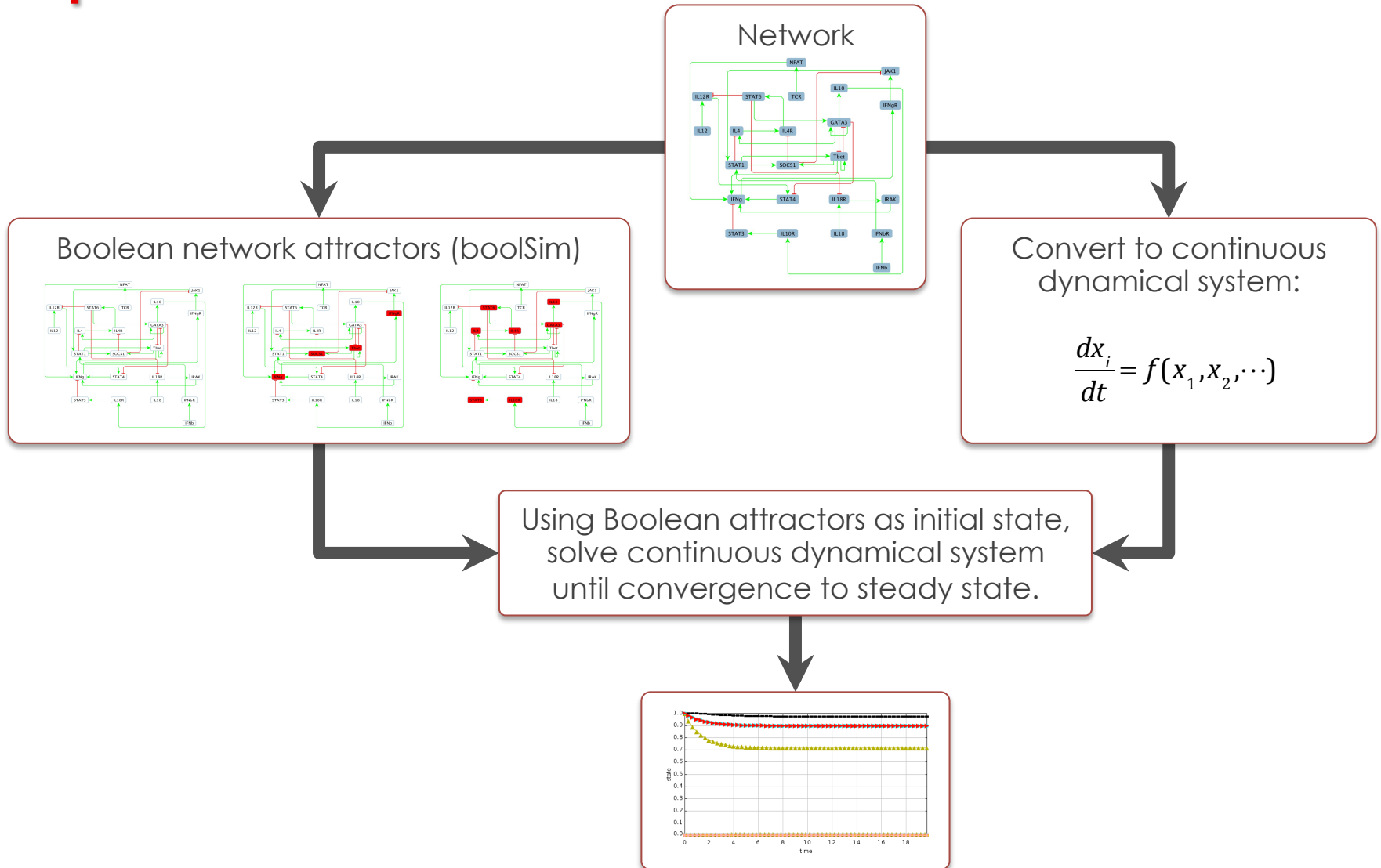
If state s is transient, then states in $BR(s)$ are all transient.
 If state s is a part of an attractor, then all the states in $FR(s)$ are also part of the same attractor.
 In the latter case, the set $BR(s) - FR(s)$ has all the transient states



Synchronous state transition graph

¹Xie and Beerel, Proceedings of DAC (1998)

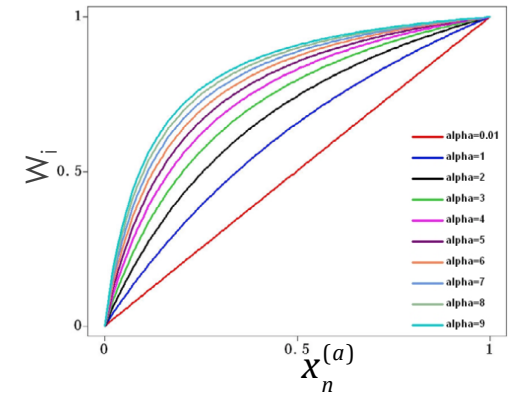
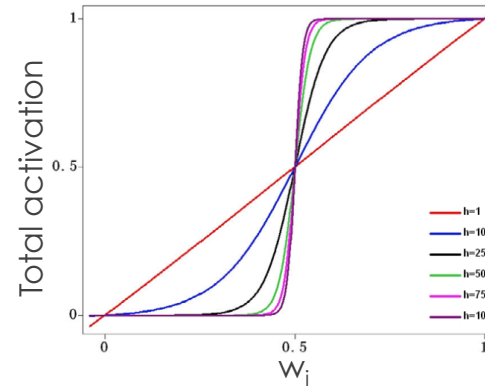
Squad: workflow



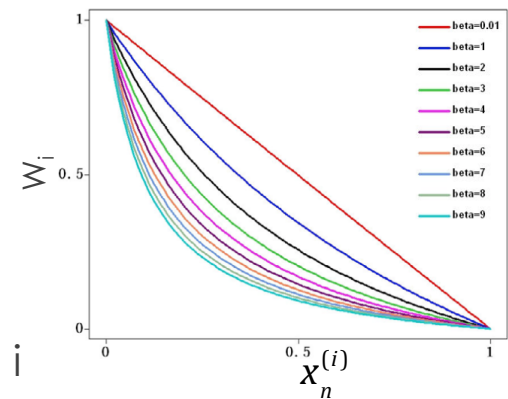
Squad: from Boolean to continuous system¹

$$\frac{dx_i}{dt} = \frac{-e^{\frac{1}{2}h_i} + e^{-h_i(w_i - \frac{1}{2})}}{\underbrace{\left(1 - e^{\frac{1}{2}h_i}\right)\left(1 + e^{-h_i(w_i - \frac{1}{2})}\right)}} - \gamma_i x_i$$

Total activation on node i



$$w_i = \underbrace{\left(\frac{1 + \sum \alpha_n}{\sum \alpha_n} \right) \left(\frac{\sum \alpha_n x_n^{(a)}}{1 + \sum \alpha_n x_n^{(a)}} \right)}_{\text{Activators}} \underbrace{\left(1 - \left(\frac{1 + \sum \beta_n}{\sum \beta_n} \right) \left(\frac{\sum \beta_n x_n^{(i)}}{1 + \sum \beta_n x_n^{(i)}} \right) \right)}_{\text{Inhibitors}}$$



x_i = State of node i

w_i = Total input on node i

h_i = Gain for node i

γ_i = decay rate of node i

$\{x_n^{(a)}\}$ = Set of activators of node i

$\{x_n^{(i)}\}$ = Set of inhibitors of node i

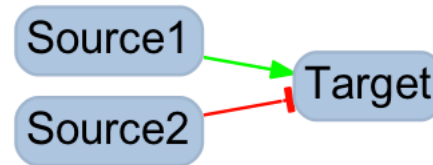
α_n = Strength of activator n

β_n = Strength of inhibitor n

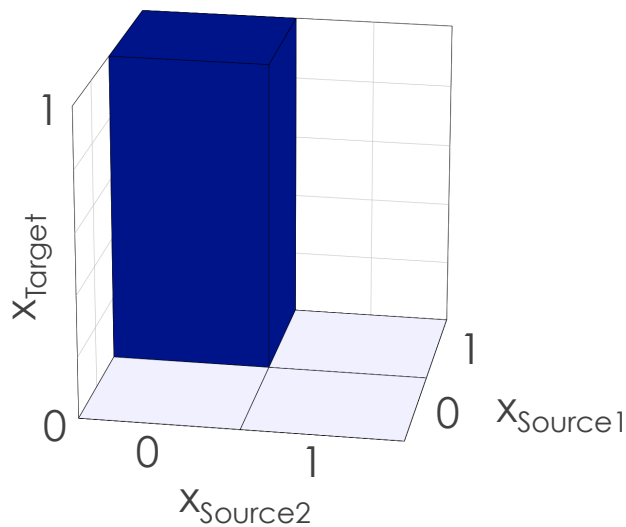
¹ A. Di Cara, A. Garg, G. De Micheli, I. Xenarios & L. Mendoza, BMC Bioinformatics (2007).

$$\frac{dx_i}{dt} = \frac{-e^{\frac{1}{2}h_i} + e^{-h_i(w_i - \frac{1}{2})}}{\underbrace{\left(1 - e^{\frac{1}{2}h_i}\right)\left(1 + e^{-h_i(w_i - \frac{1}{2})}\right)}} - \gamma_i x_i$$

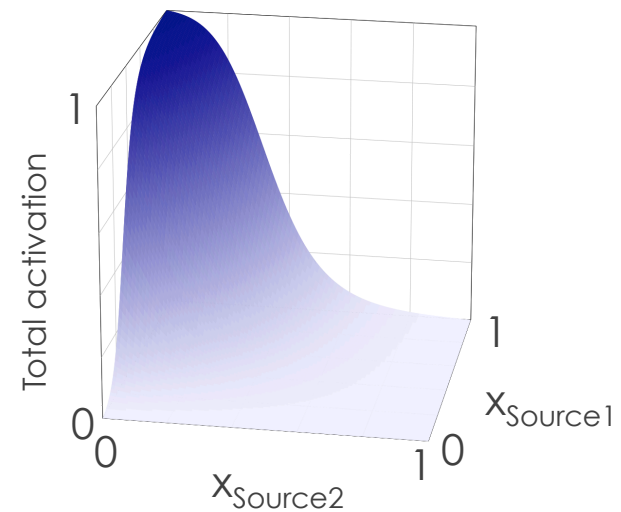
Total activation on node i



Boolean

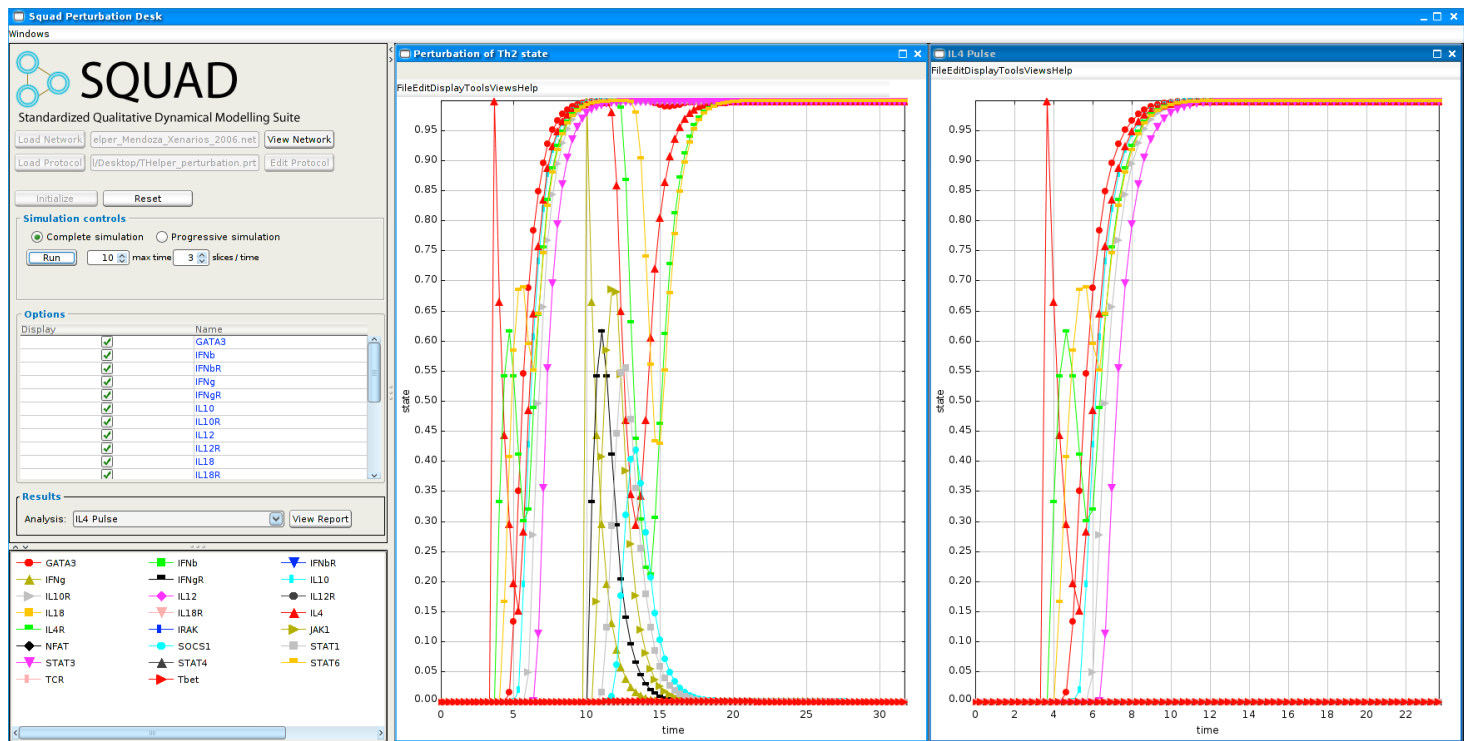


Continuous



Squad: usage

More in the tutorial



<http://compbio.igc.gulbenkian.pt/nmd/node/63>