

# GINsim tutorial (DRAFT)

February 27, 2007

## 1 Introduction

### 1.1 About GINsim

GINsim (see [1]) is a tool supporting the definition, the simulation and the analysis of regulatory graphs, based on a **multilevel logical formalism** (see Table 2). This formalism leans on the definition of two types of graphs: **regulatory graphs**, which model regulatory networks, and **state transition graphs**, which represent the dynamical behaviours, for given initial state(s) and under some updating assumptions.

### 1.2 Availability

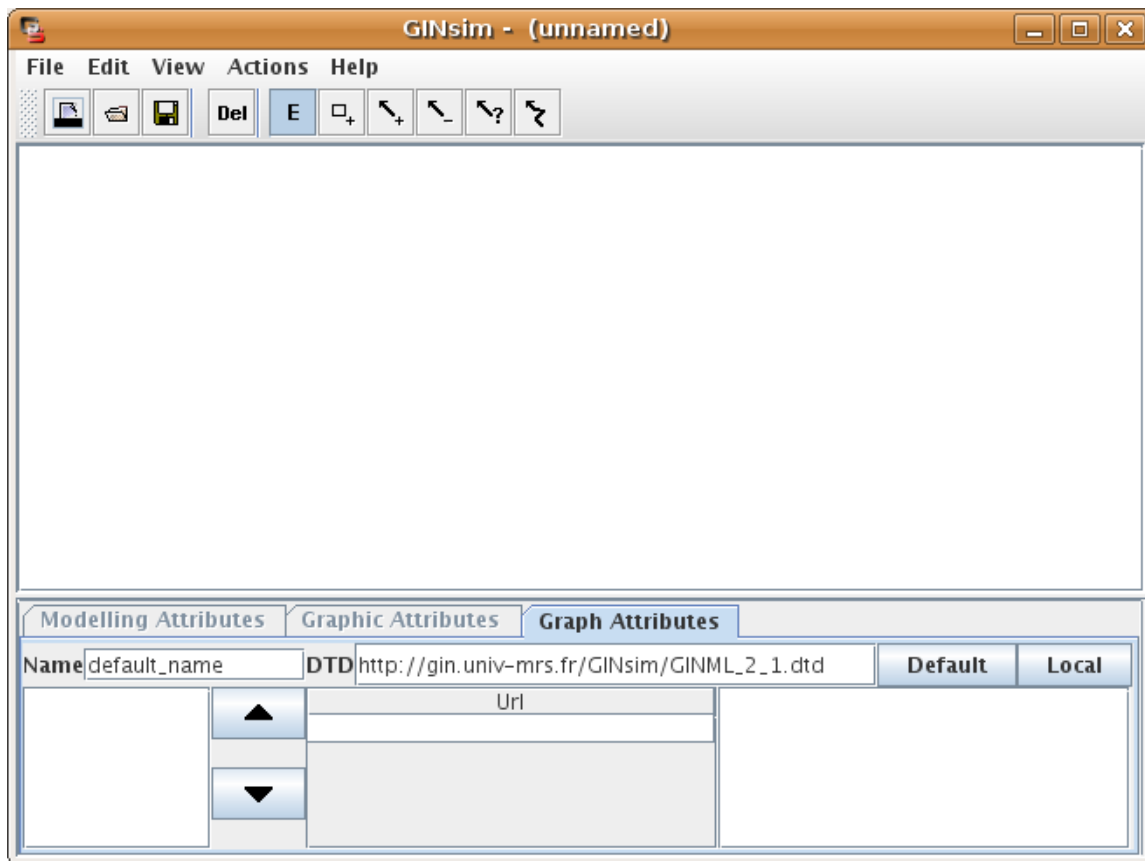
GINsim 2.2 is freely available, under the terms of the *General Public License*. The GINsim website (<http://gin.univ-mrs.fr/GINsim>) contains up to date informations, along with the latest official version of GINsim and of this documentation. To get started with GINsim, download the .jar file from the website and launch it (either by double-click or with the command `java -jar GINsim-$version.jar`).

### 1.3 Overview of this tutorial

This short tutorial introduces the main steps to get started with GINsim:

- the specification of a regulatory graph, illustrated through a simple model of the lysis-lysogeny decision in the phage lambda,
- the simulation of this model (i.e the construction of the state transition graph),
- the use of some of the available analysis tools.

We refer to the GINsim user manual for more detailed information.



The main window of GINsim, featuring an empty model.

Figure 1: Main window of GINsim

## 2 Defining a regulatory network model


In the logical formalism, a regulatory network is modelled as a **regulatory graph**. In this graph, nodes represent genes or regulatory products, and arcs represent interactions between genes. Modelling attributes determine the dynamical behaviour of the network.

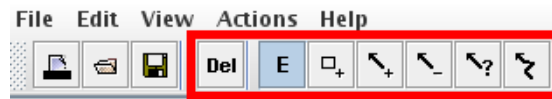
GINsim allows to define interactively such models through a dedicated graphical interface. A graph editor allows the addition/removal of genes and interactions. Then, the property panel at the bottom of the main GINsim window enables the specification of the modelling attributes.

### 2.1 Drawing the regulatory graph

The first step for the definition of a regulatory model, is the definition of the graph *per se*. Genes and interactions can be interactively added, selected, moved or deleted, depending of the current editing mode. This editing mode can be changed using the edit menu or the toolbar buttons.

Available editing modes:	
<b>E</b>	Default editing mode: allows to select and move objects.
	Gene insertion mode: when selected, clicking on the graph panel adds a new gene.
	Interaction insertion mode: when selected, interactions are added by dragging from one gene to another. The interactions must be complemented by the definition of the logical parameters for the target variable (see below). The three buttons allow to add different kinds of interactions: activation, inhibitions or undefined.
<b>Del</b>	Deletes all selected items.

<b>Available editing modes:</b>	
	Add/Remove intermediate points to arcs.



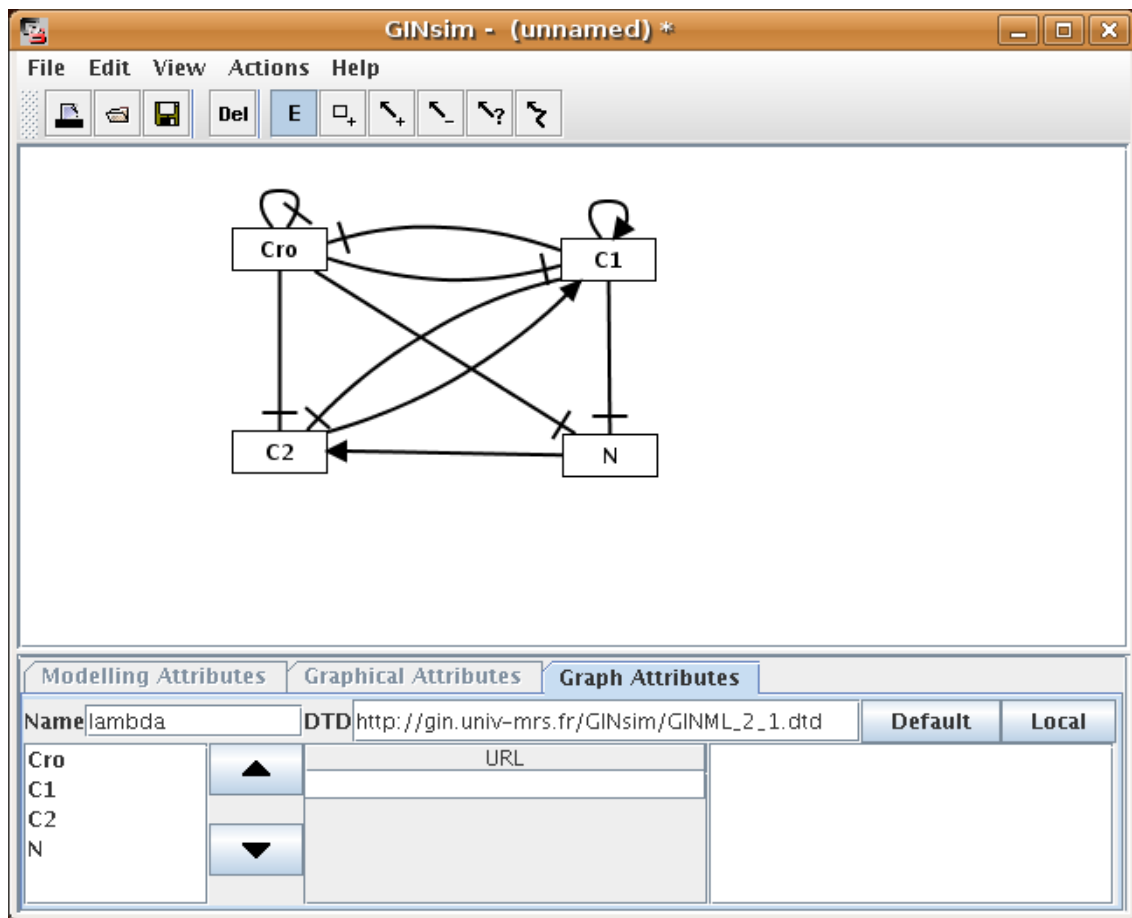
GINsim has several editing modes, available in the edit menu but also in the main toolbar. The pushed button reflects the current mode.

Figure 2: Graph editing buttons in the toolbar



**Tip** The selection of non-default editing modes is transient by default: once an action has been performed, GINsim switches back to the default mode (E). GINsim can be locked in any mode by double-clicking on the corresponding toolbar button.

Let us take the example of a simple model of the lysis-lysogeny decision in the phage lambda. This tutorial focuses on the definition and the simulation of the model in GINsim. A more complete description of the model is available, along with a GINsim model definition file, in the [model repository](#). This model, involving 4 genes, is called "lambda4".



This GINsim window shows the four actors of the model and their interactions (blunt arrows stand for inhibitory effects, normal ones for activations).

Figure 3: The lambda4 model

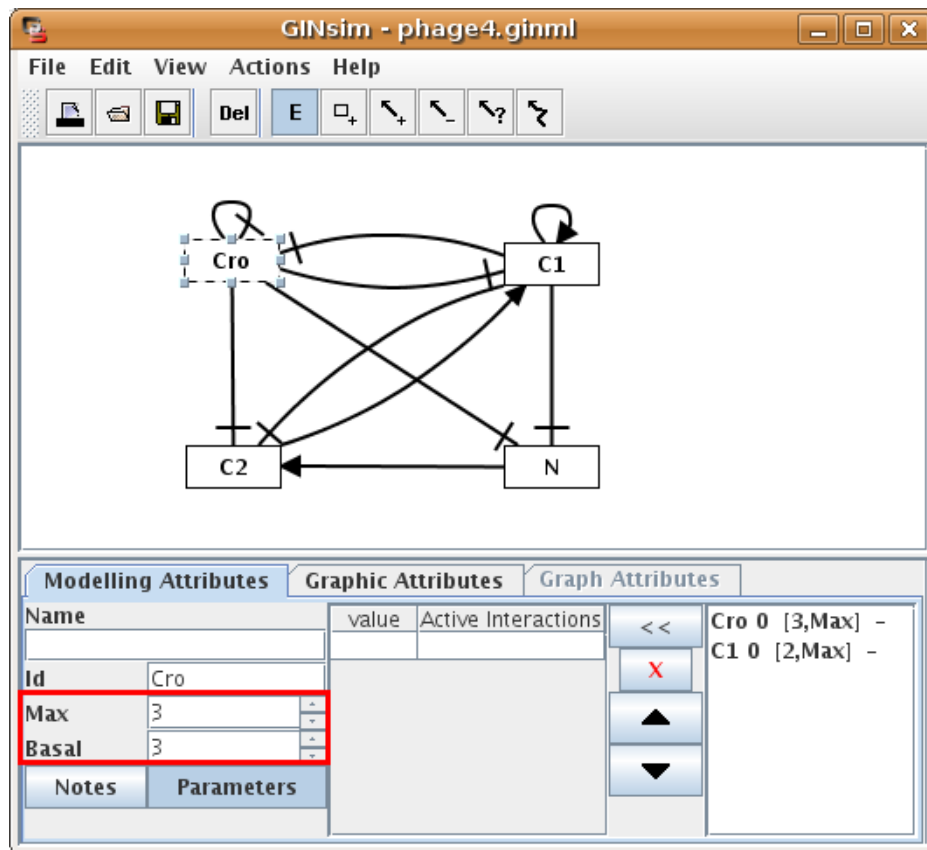
## 2.2 Entering parameters for genes and interactions

Each gene has a maximal and a basal expression level. The basal level corresponds to the situation where no incoming interaction is active. For this model, the values are listed in Table 1. GINsim assigns a default maximal level of 1 and a basal level of 0 to newly

Gene	Maximal	Basal
Cro	3	3
C1	2	2
C2	1	0
N	1	1

Table 1: Genes and their expression levels

created genes. To change these values for a given gene, one needs to select the gene and then to specify them through the *modelling attributes* tab in the bottom panel, as shown in Figure 4.



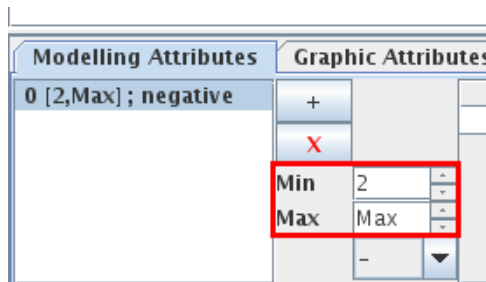
When a gene is selected, the modelling attributes tab in the bottom panel allows to edit properties of this gene. Name (optional), Id (unique identifier) and expression levels can be edited in the left part, while the right part is dedicated to the logical parameters (see below).

Figure 4: Specification of maximal and basal levels for Cro

Interactions operate for a given range of activity levels of their source. The next step consists in defining these ranges for each interaction, whenever necessary (per default, the interval is set to  $[1-\text{max}]$ , where "max" is the maximal expression level of the source). For the selected interaction, these values can be specified through the *modelling attributes* area (see Figure 5). The Table 2 lists interactions between genes of the lambda4 model.

Source	Target	Range
Cro	Cro	[3,Max]
Cro	C1	[1,Max]
Cro	C2	[1,Max]
Cro	N	[2,Max]
C1	Cro	[2,Max]
C1	C1	[2,Max]
C1	C2	[2,Max]
C1	N	[1,Max]
C2	C1	[1,Max]
N	C2	[1,Max]

Table 2: Interactions and their activity ranges



When an interaction is selected, the modelling attributes tab of the bottom panel allows to edit its properties.



**Note** In the non-boolean case, a gene can have several distinct effects on another one, depending on its activity level. In this case, only one arc is drawn and all distinct effects appear in the left part of this configuration panel.

Figure 5: Specification of the interval associated with the Cro-N interaction

### 2.3 Editing Logical Parameters

Logical parameters are attached to each gene, to describe its dynamical behaviour. Each logical parameter is defined as a set of incoming interactions and a target value. It denotes that, during the simulation, when

- the specified incoming interactions are all active (i.e. the expression level of their source genes are in the related activity ranges),
- and none of the other incoming interactions is active,

then the expression level of the considered gene tends toward the value attached to the logical parameter.

Per default, all parameters are considered to be zero. Consequently, only the non-zero parameters have to be explicitly defined (listed in Table 3 for the lambda4 model).

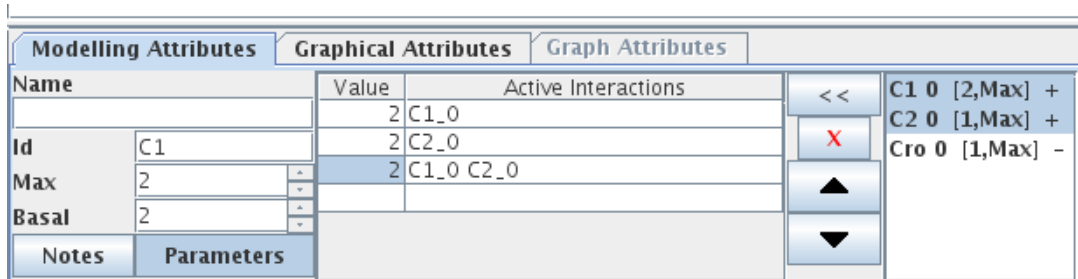
Gene	active interactions	target value
Cro	Cro	2
C1	C1	2
	C2	2
	C1,C2	2
C2	N	1

Table 3: Non-zero logical parameters

When a gene is selected, the main part of the *modelling attributes* tab enables the definition of the logical parameters, as shown in Figure 6.

The panel dedicated to the definition of logical parameters is divided in three parts:

- On the left, a table lists all defined logical parameters, showing their target value and list of interactions.
- On the right, the list of all incoming interactions on the selected gene.
- The central part contains the following buttons:
  - "«" assigns the set of selected incoming interactions (from the right list) to the selected logical parameter (use *shift* and *ctrl* to select several interactions).
  - "X" deletes the selected logical parameter,
  - up and down arrows reorder the selected logical parameters.

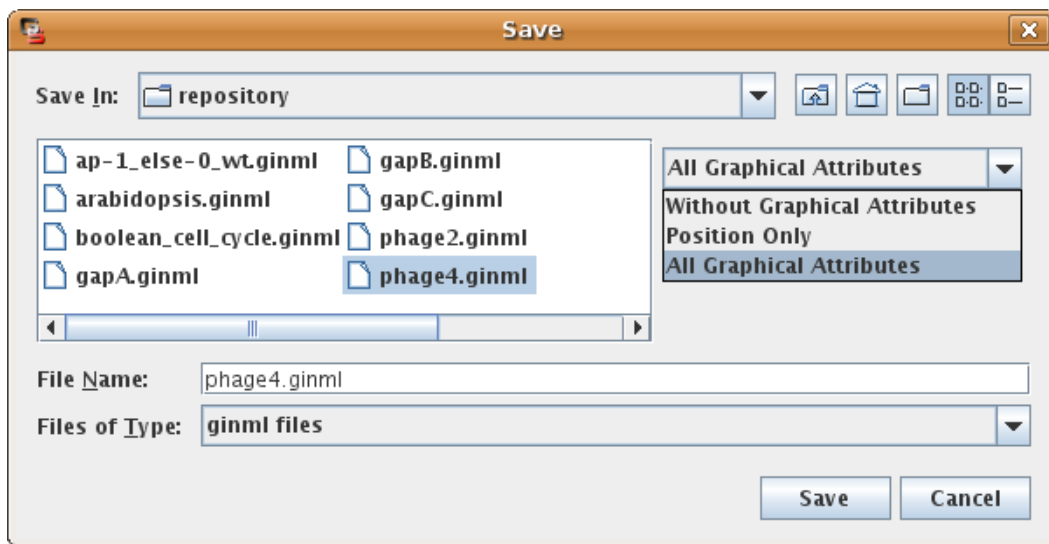


The logical parameter panel, showing all parameters for C1.

Figure 6: Definition of non zero parameters for C1

## 2.4 Saving the graph

The *File* menu contains the classical actions allowing to save, open or export files. GINsim uses a dedicated XML format: **GINML**. The Save dialog box enables the saving of the graphical attributes. The default is to save all graphical attributes. Otherwise, the sole structure of the graph or only the position of genes can be saved (see Figure 7).



GINsim uses the default java save dialog box, allowing to browse and create folders. By default, only folders and GINML files are shown. Saving requires to choose the name of the file.

Figure 7: The Save dialog box

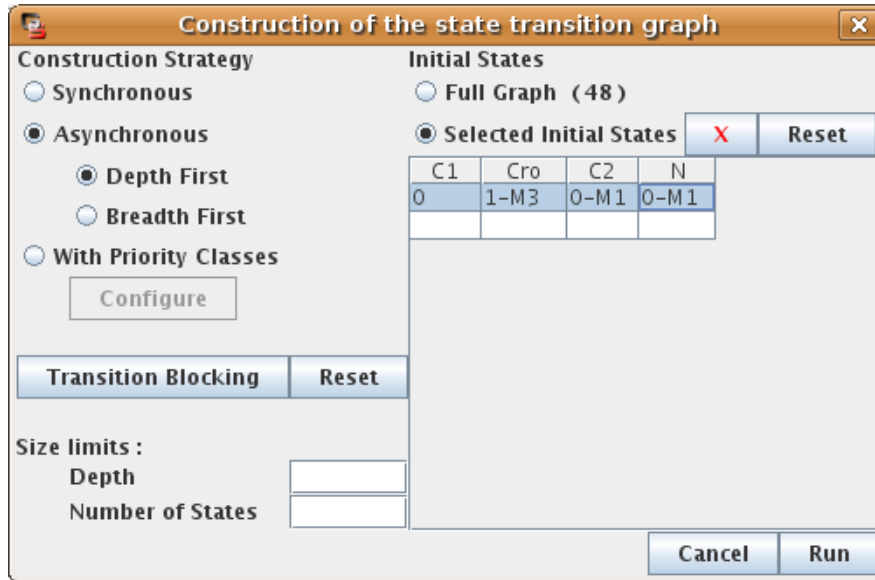
## 3 Running a simulation

Once a regulatory model has been defined, a simulation can be launched through the option *Run Simulation* option of the *Action* menu.

The simulation computes a *state transition graph*, representing the dynamical behaviour of the model. The nodes of this graph, also called *states* of the system, are vectors giving the expression levels of all genes of the model.

At a given state, a specific logical parameter can be associated with each regulatory component. If the value of this parameter is smaller or greater than that of the concentration/activity level of the corresponding component, this level will tend to decrease or increase, respectively. Otherwise (when the parameter value and the corresponding component level are equal), the component will tend to keep its current value.

Building a state transition graph involves the selection of a "strategy" for its construction, the specification of "initial state(s)" as well as optional size constraints (not detailed here).



The simulation dialog box, ready to run an asynchronous depth first simulation, using a set of initial states. The set of initial states allows a single value for C1, three values (from 1 to 3) for Cro, and two values for C2 and N.  $1 \cdot 3 \cdot 2 \cdot 2 = 12$  initial states will be considered. "M" before a value emphasizes the selection of the maximal value.

Figure 8: Dialog box to configure the simulation

### 3.1 Initial state(s)

The right part of the dialog box is dedicated to the selection of the initial states of the simulation. The simulation will start with each of these initial states, unless they have already been reached by a previous simulation (i.e. from another initial states).

The simulation can consider all possible states as initial states (resulting to the *full* state transition graph), or use only a specific subset, defined in the table. Each line of this table describes one or several initial states, by mean of a set of allowed values for each gene of the system.

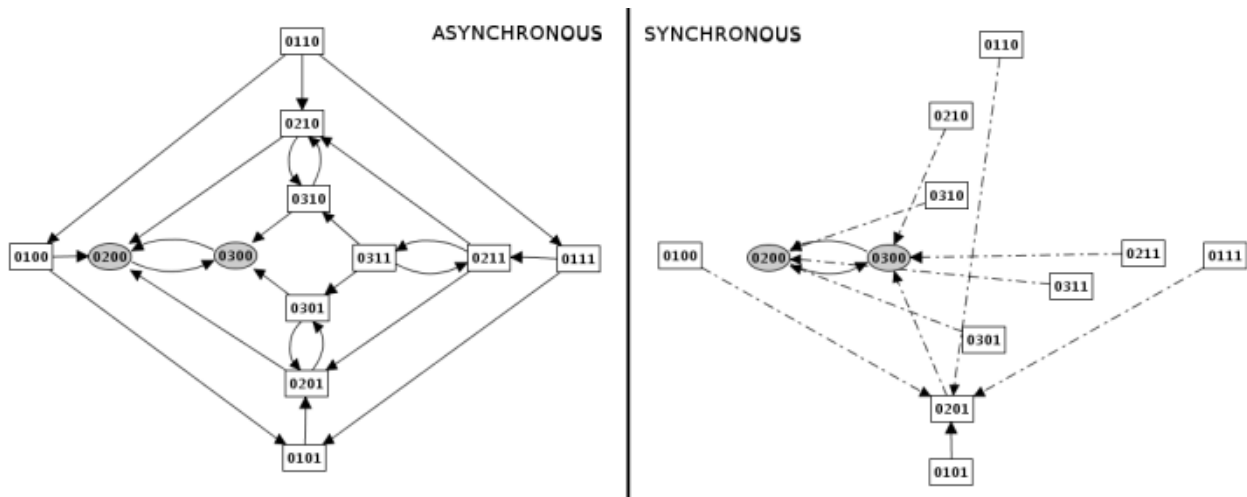
### 3.2 Construction strategy

When several genes are called to update their expression levels at the same time, several strategies can be applied. In GINsim, the following construction strategies are implemented:

- Synchronous: all the updating calls are executed simultaneously. This strategy is generally considered unrealistic, but it is still useful in some cases.
- Asynchronous: only one updating call is executed at each step, but all possible trajectories are generated in the resulting state transition graph. This is the default strategy.
- Priority classes: this is a mixed mode where updating calls are grouped in priority classes (see below).



**Note** Stable (steady) states are conserved whatever the selected updating assumption. However, their reachability may change.



Samples of simulation results for the lambda4 model, applying synchronous and asynchronous strategies to the same initial states (all states where  $C1=0$  and  $Cro>0$ ). Dotted arcs denotes transitions where several components are updated simultaneously.

Figure 9: Construction strategy: synchronous versus asynchronous

### 3.2.1 Priority classes

GINsim allows the user to group components into different classes, and to assign a priority level to each of these classes. In case of concurrent transition calls, GINsim first updates the gene(s) belonging to the class with the highest ranking. For each regulatory component class, the user can further specify the desired updating assumption, which then determines the treatment of concurrent transition calls inside that class. When several classes have the same ranking, concurrent transitions are treated under an asynchronous assumption (no priority).

By default, all transitions belong to the same asynchronous class. Running the simulation without further configuration is thus the same as using the asynchronous assumption.

The left part of the configuration dialog box shows a list of priority classes (see Figure 10). The name of a class can be edited and a checkbox allows to change its internal mode from asynchronous (unchecked) to synchronous (checked). Buttons enable to add (+), delete (X), order (using the arrows) and group/ungroup (G) them.

The central column lists transitions that belong to the currently selected class, whereas the column on the right lists all other transitions (belonging to other classes). To add transitions to the current class, select these transitions in the right list and click on the "<<" button. The ">>" button removes the selected transitions in the central list from the current class (and add them to the first class of the list).

Finally, a choice list on the bottom left, allows to apply some predefined settings.

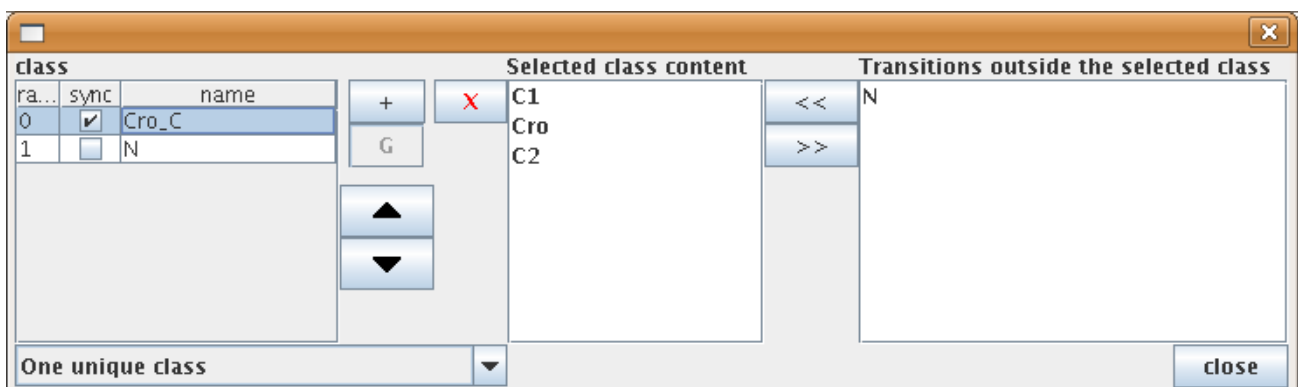
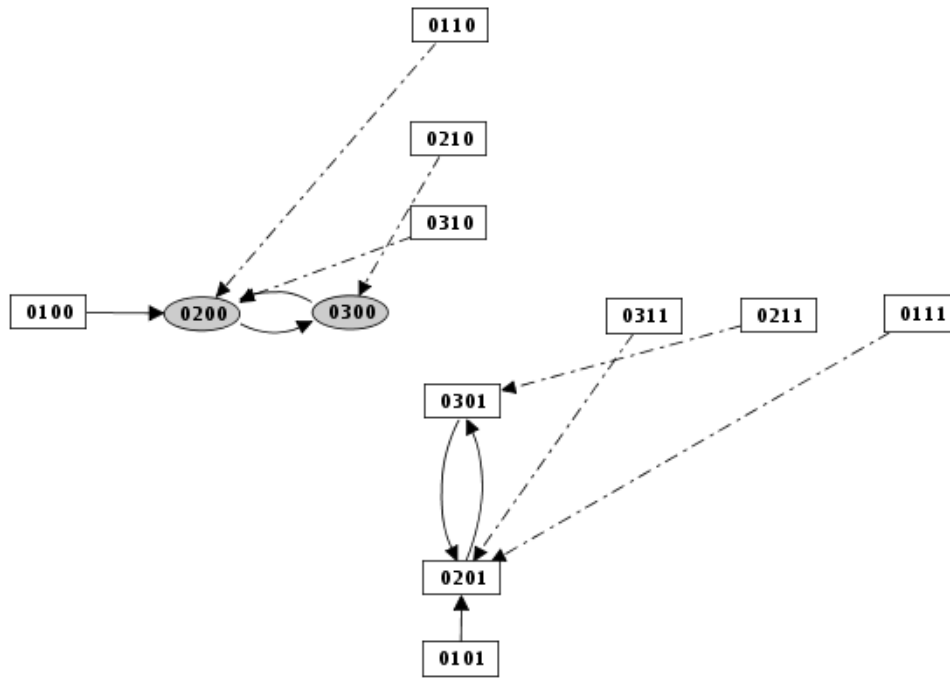


Figure 10: Definition of Priority classes



Example of simulation using priority classes. Two priority classes have been created. The highest ranked one is synchronous and contains C1, C2 and Cro. The other class contains only N. The resulting state transition graph is splitted into two parts: N expressed vs. N not expressed.

Figure 11: Priority Class: example result

### 3.3 Transition blocking

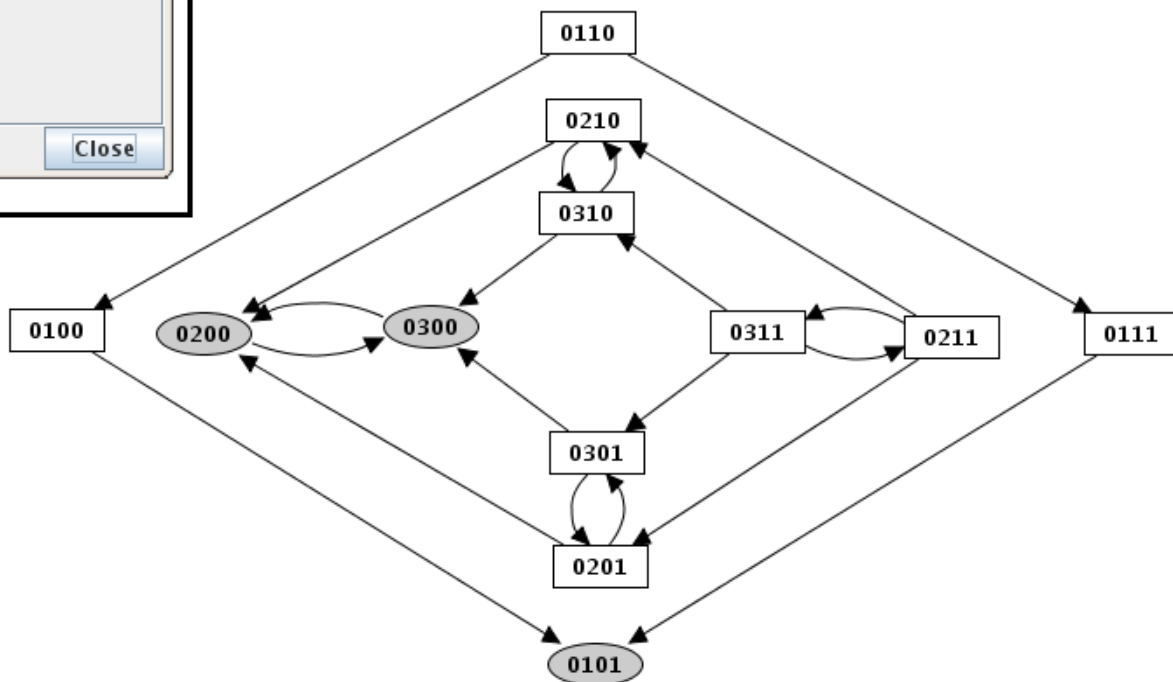
Blocking transition is putting some restrictions on the evolution of the expression level of genes. The *transition blocking* button opens a dialog box, allowing to specify minimal and maximal expression level for each gene.



**Warning** Transition blocking prevents the expression level of a gene from leaving a given range, but it does not affect transitions outside of this range. Completely locking the expression level requires to define initial states in the same range.

Transition blocking is especially useful to simulate simple or multiple mutations (knockouts or ectopic gene expression).

Node	Min	Max
C1		
Cro	1	1
C2		
N		

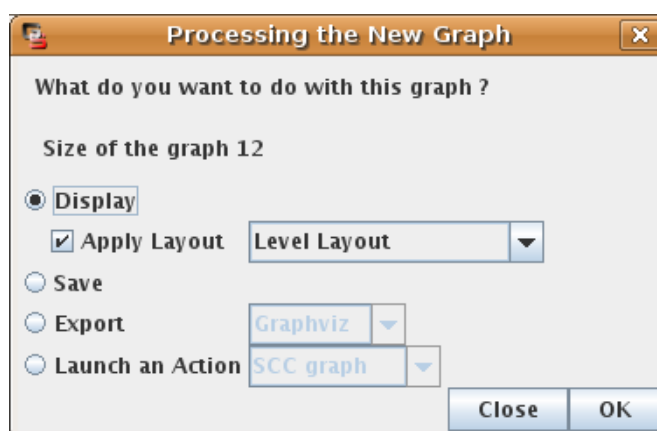


Result of an asynchronous simulation, where the expression level for Cro has been blocked at 1. The state transition graph is the same as the original asynchronous one, but all transitions where Cro leaves this value have been suppressed. This state transition graph is now composed of two disconnected parts, and a new stable state appeared.

Figure 12: Transitions blocking

### 3.4 Processing the new graph

At the end of the simulation, a dialog box provides some informations on the generated state transition graph: size (number of states) and number of stable states. By default, GINsim proposes to save or display the graph, but the user can also run different analysis on it (see Figure 13).



After a simulation, this dialog box allows to save the state transition graph or to perform further analysis without displaying it.

Figure 13: "Processing the new graph" dialog box

## 4 Analysis tools

### 4.1 Strongly connected component

GINsim enables the identification of the Strongly Connected Components graph (SCC graph, see [http://en.wikipedia.org/wiki/Strongly\\_connected\\_component](http://en.wikipedia.org/wiki/Strongly_connected_component)) of an existing graph. The SCC graph is an acyclic graph based on the original one: each node of the SCC graph is a cycle or a set of intertwined cycles of the original graph. To compute this graph, use the "SCC graph" action in the "Actions" menu.

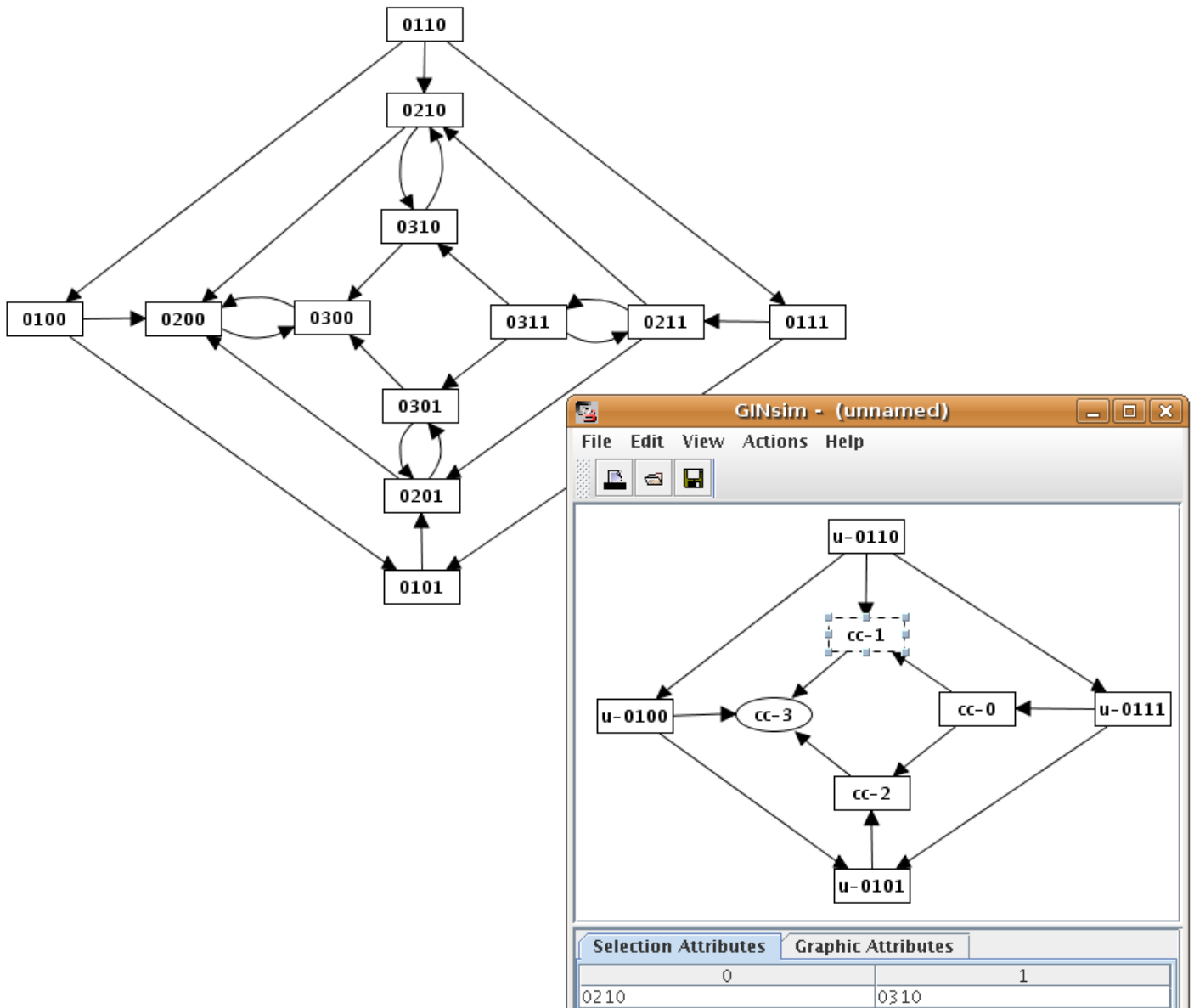


Figure 14: Strongly Connected Components Graph

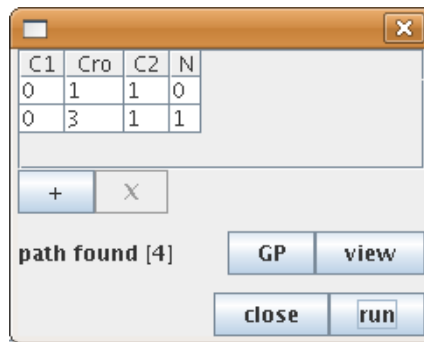


**Tip** While displaying a SCC graph, the "selection attribute" tab in the bottom panel shows the content of the selected component (i.e. the list of corresponding nodes in the original graph).

### 4.2 Path finder

It is often important to find a path between different states of the state transition graph.

The "search path" action allows to find a path between different states of the state transition graph. Selecting the "search a path" action in the action menu opens the configuration dialog. This dialog allows to specify the starting state, the targeted state, and intermediate states (using the "+" and "X" buttons to add/remove them). GINsim then performs a shortest path search (If several paths exist between these states, the result will thus be the first shortest path).



The search tool has found a path from the state 0110 to 0311, involving four steps.

Figure 15: Search for a path in the state transition graph

The animation tool allows to select a path interactively. It highlights genes of the regulatory graph according to the selected state.

## 5 References

- [1] Claudine Chaouiya, Elisabeth Remy, Brigitte Mossé, and Denis Thieffry, *Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework*, Lectures Notes in Control and Information Sciences, 2003, 294, 119-126.
- [2] Aitor Gonzalez Gonzalez, Aurélien Naldi, Lucas Sanchez, Denis Thieffry, and Claudine Chaouiya, *GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks*, Biosystems, 2006, 84, 91-100, doi:10.1016/j.biosystems.2005.10.003.