Analysing Gene Networks with PDP Systems. *Arabidopsis thaliana*, a Case Study

Luis Valencia–Cabrera[1], Manuel García–Quismondo[1], Mario J. Pérez–Jiménez[1], Yansen Su[2], Hui Yu[2], and Linqiang Pan[2]

[1] Research Group on Natural Computing
Dpt. of Computer Science and Artificial Intelligence, University of Sevilla
Avda. Reina Mercedes s/n. 41012 Sevilla, Spain
{lvalencia,mgarciaquismondo,marper}@us.es

[2] Department of Control Science and Engineering, Huazhong University of Science and Technology
1037 Luoyu Road, Wuhan, China
suyansen1985@163.com, yuhuihustac@gmail.com, lqpan@mail.hust.edu.cn

Summary. Gene Regulatory Networks (*GRNs*) are a useful tool for biologists to understand the interactions among genes in living organisms. A special kind of GRNs known as Logic Networks (*LNs*) has been recently introduced. These networks consider that the state of one or more genes can influence another one. In a previous work, we proposed a Membrane Computing model which simulates the dynamics of LNs by drawing on the improved LAPP algorithm. In this paper we provide a case study for our LN model on a network which regulates the circadian rhythms of long–term studied plant *Arabidopsis thaliana*. We outline the software tools employed and propose a methodology for analysing LNs on our Membrane Computing model. At the end of the paper, some conclusions and future work are included.

Keywords: Bioinformatics, Genetics, Gene networks, Membrane Computing, MeCoSim, Software engineering, Modelling, LAPP, Logic networks

1 Introduction

Since its very beginning, Membrane Computing [13] has been employed as a modelling framework for biochemical phenomena. Although the current landscape is more focused on metabolite–oriented dynamics, gene regulatory networks (GRNs) have also been modelled by means of P systems as part of this framework. In a previous work, we followed this line of research by proposing a Membrane Computing model for a specific type of gene networks known as Logic Networks (*LN*) [16]. This model describes a P systems family known as LN Dynamic P systems (*LN*

DP systems), within the framework of PDP systems [11]. LNs are a specific type of GRNs in which the combination of states of several genes, rather than the single state of any of them, influence another one. Bowers *et al.* [2] proposed a methodology for the construction of logic networks out of statistical data, known as Logical Analysis of Phylogenesis Profiles (LAPP). In our model, these combinations are limited to at most two genes affecting a third one. The model, in conjunction with DCBA algorithm[4], intends to capture the behaviour of the Improved LAPP Method introduced by Wang *et al.* [18]. In their work, they propose a case study on a gene network associated to *Arabidopsis thaliana*'s flowering process. We intend to reproduce this case study by using our Membrane Computing model. We also include a guide for generating custom simulators on MeCoSim for LN DP systems, depicting a step–by–step guide on MeCoSim tool [12]. Finally, the data employed in this case study is provided as an appendix, thus easing cross–checking of results. This paper is structured as follows. Section 1.1 introduces the Logic Network to be studied, a GRN associated to the flowering process of *Arabidposis thaliana*. Section 2 outlines the LN DP system model presented in [16], in order to make the current work self contained, as it is used to analyse our case study. Section 3 consists of a guide to simulate LNs from scratch on MeCoSim [14, 12]. This guide complements the simulation methodology described in [16]. Section 4 describes a case study on a real–world logic network on Arabidopsis Thaliana, in order to experimentally verify the behaviour of the model on complex gene networks. Finally, section 5 lists the conclusions obtained and proposes some open problems.

1.1 A Logic Network on *Arabidopsis thaliana* flowering processes

Arabidopsis is a long–day plant. Zhang and Zuo [19] stated that long–day conditions can promote reproductive growth and induce early flowering. However, short–day conditions can promote vegetative growth and induce late flowering or even no flowering. To understand the intrinsic mechanisms of Arabidopsis flowering in different lighting conditions, it is required to compare the relationships of related genes.

In the latest ten years, much work has been reported in the field about *A. thaliana* flowering. Imaizumi *et al.* [8] found that FKF1 is a blue light receptor which regulates flowering. Later, they also showed that FKF1 together with Flavin–Binding and Kelch Repeat degrade Cycling Dof Factor1 (CDF1) to eventually control carbon monoxyde [7]. In the same year, Abe *et al.* [1] found that Flowering Locus T (FT) together with FD activate Apetala1 (AP1) to initiate floral development and promote floral transition at the shoot apex. Previous work deal only with one or few genes related to flowering. However, the networks considered in this work focus on the relationships among a large number of genes systematically. Bowers *et al.* [2] proposed the Logic Analysis of Phylogenetic Profiles (LAPP) [2]. This method helps researchers to know biological functions of some genes or proteins on the basis of phylogenetic profiles, which has been developed both on theory and application ([3, 20, 17]). For example, Wang *et al.* [17]

developed the improved LAPP method, and reversely constructed a logic network of sixteen genes in shoot for Arabidopsis under salt stimuli.

2 Description of the model

This section summarizes both the P system family and the model (i.e., initial configuration and rule patterns) employed in this case study. For a detailed description of the model, see [16].

2.1 A family of P systems based on Logic Networks

The model depicted here is a P system of a family known as Logic Network Dynamic P systems (*LN DP systems*). An LN DP system is described within an expansion of Population Dynamics P systems (*PDP systems*) [16].

An LN DP system Π_{LN} of degree (q, m) with $q, m \geq 1$, taking $T \geq 1$ time units, is a tuple

$$\Pi_{LN} = (G, \Gamma, \Sigma, T, R_E, \mu, R, \{f_{r,j} : \ r \in R, 1 \leq j \leq m\}, \\ \{\mathcal{M}_{ij} : \ 1 \leq i \leq q, \ 1 \leq j \leq m\}, \{\mathcal{M}_j : 1 \leq j \leq m\})$$

where:

- $(G, \Gamma, \Sigma, T, R_E, \mu, R, \{f_{r,j} : \ r \in R, 1 \leq j \leq m\}, \{\mathcal{M}_{ij} : \ 1 \leq i \leq q, \ 1 \leq j \leq m\})$ is a PDP system.
- $\{f_{r,j} = 1 : \ r \in R, 1 \leq j \leq m\}$.
- For each j $(1 \leq j \leq m)$, $\mathcal{M}_j$ are multisets over Γ, describing the objects initially placed in the m environments e_j.

In this paper, in the description of an LN PDP System, functions $f_{r,j}$ are omitted. They are all equal to 1, so it is not necessary to make them explicit.

2.2 The model

Here the model for the family of Logic Network Dynamic P systems is outlined. This model covers any possible P system in this family, so the multisets, rules, etc. depend on the P system which represent each specific instance of a logic network. The definition of the general model requires the use of parameters in our constructs, as explained at the end of this subsection.

Let LN be a logic network. Let ng, nu, nb be the number of genes, unary and binary interactions of LN, respectively. Let $n = ng + nu + nb$. The model consists of the following PDP system of degree $(1, n)$,

$$\Pi_{LN} = (G, \Gamma, \Sigma, T, R_E, \mu, R, \{\mathcal{M}_{ij} : \ 0 \leq i \leq q-1, \ 1 \leq j \leq m\}, \{\mathcal{M}_j : \ 1 \leq j \leq m\})$$

where:

- G is a directed graph containing a node (environment) for each gene, unary or binary interaction, following this order.
- In the alphabet Γ, we represent gene states, interaction types, contribution weights and targets.

$$\begin{aligned}\Gamma = {} & \{a_i, b_i, c_i : 0 \leq i \leq 1\} \cup \{go, d_0\} \cup \{unop_j, binop_j : 1 \leq j \leq 4\} \cup \\ & \{auxDest_{i,g_{j,1},k} : 0 \leq i \leq 1, 1 \leq j \leq ng, 1 \leq k \leq nb + nu\} \cup \\ & \{dest_{i,g_{j,1},t_{k,1}+ng} : 0 \leq i \leq 1, 1 \leq j \leq ng, 1 \leq k \leq nb\} \cup \\ & \{dest_{i,g_{j,1},unt_{k-nb,1}+ng+nb} : 0 \leq i \leq 1, 1 \leq j \leq ng, nb+1 \leq k \leq nb+nu\} \cup \\ & \{e_{t_{k,4}*i+(1-i)*(1-t_{k,4}),t_{k,1}+ng} : 0 \leq i \leq 1, 1 \leq k \leq nb\} \cup \\ & \{e_{t_{k,6}*i+(1-i)*(1-t_{k,6}),t_{k,1}+ng} : 0 \leq i \leq 1, 1 \leq k \leq nb\} \cup \\ & \{e_{unt_{k-nb,4}*i+(1-i)*(1-unt_{k-nb,4}),unt_{k-nb,1}+ng+nb} : \\ & \ 0 \leq i \leq 1, nb+1 \leq k \leq nb+nu\} \cup \\ & \{eF_{t_{k,8}*i+(1-i)*(1-t_{k,8}),t_{k,1}+ng} : 0 \leq i \leq 1, 1 \leq k \leq nb\} \cup \\ & \{eF_{i,(unt_{k,1}+ng+nb)} : 0 \leq i \leq 1, 1 \leq k \leq nu\} \cup \\ & \{clock_j : 0 \leq j \leq cc+3\}\end{aligned}$$

- The environment alphabet is $\Sigma = \Gamma \setminus \{d_0\}$
- Each cycle to evolve from a real network configuration to the next one involves 15 computational steps, so $T = 15 \cdot Cycles$, where $Cycles$ is the number of cycles to simulate.
- $\mu = [\ \ ]_1$ is the membrane structure.
- The initial multisets are:
  - $\mathcal{M}_{g_{k,1}} = \{\ {a_1}^{g_{k,3}}, {a_0}^{1-g_{k,3},go} : 1 \leq k \leq ng\}$.
  - $\mathcal{M}_{ng+t_{i,1}} = \{\ binop_{t_{i,2}} : 1 \leq i \leq nb\}$.
  - $\mathcal{M}_{ng+nb+unt_{i,1}} = \{\ unop_{unt_{i,2}} : 1 \leq i \leq nu\}$.
- The rules of R and R_E to apply are showed below. They are put together to follow the sequential order of execution. Environment rules start with re and skeleton rules start with rs.
  - $rs_{1,i} \equiv go\, a_i[\,]_1 \longrightarrow c_i {b_i}^{max*i} {b_0}^{threshold} clock_0[\,]_1 : 0 \leq i \leq 1$
  - For each source gene environment:

$$\begin{aligned}re_{2,i,j,k} \equiv {} & (c_i \longrightarrow \{auxDest_{i,g_{j,1},k} : \{1 \leq k \leq nb+nu\}\})_{g_{j,1}} \\ & : 0 \leq i \leq 1, 1 \leq j \leq ng \\ re_{3,i,j,k} \equiv {} & (auxDest_{i,g_{j,1},k} \longrightarrow dest_{i,g_{j,1},t_{k,1}+ng})_{g_{j,1}} \\ & : 0 \leq i \leq 1, 1 \leq j \leq ng, 1 \leq k \leq nb \\ re_{4,i,j,k} \equiv {} & (auxDest_{i,g_{j,1},k} \longrightarrow dest_{i,g_{j,1},untk-nb,1+ng+nb})_{g_{j,1}} \\ & : 0 \leq i \leq 1, 1 \leq j \leq ng, nb+1 \leq k \leq nb+nu \\ re_{5,i,k} \equiv {} & (dest_{i,t_{k,3},t_{k,1}+ng} \longrightarrow e_{t_{k,4}*i+(1-i)*(1-t_{k,4}),t_{k,1}+ng})_{t_{k,3}} \\ & : 0 \leq i \leq 1, 1 \leq k \leq nb \\ re_{6,i,k} \equiv {} & (dest_{i,t_{k,5},t_{k,1}+ng} \longrightarrow e_{t_{k,6}*i+(1-i)*(1-t_{k,6}),t_{k,1}+ng})_{t_{k,5}} \\ & : 0 \leq i \leq 1, 1 \leq k \leq nb \\ re_{7,i,k} \equiv {} & (dest_{i,unt_{k-nb,3},unt_{k-nb,1}+ng+nb} \longrightarrow \\ & e_{unt_{k-nb,4}*i+(1-i)*(1-unt_{k-nb,4}),unt_{k-nb,1}+ng+nb})_{unt_{k-nb,3}} \\ & : 0 \leq i \leq 1, nb+1 \leq k \leq nb+nu\end{aligned}$$

$re_{8,i,k} \equiv \ (\quad)_{t_{k,1}+ng}(e_{i,t_{k,1}+ng})_{t_{k,3}} \longrightarrow (a_i)_{t_{k,1}+ng}(\quad)_{t_{k,3}}$
$: 0 \leq i \leq 1, 1 \leq k \leq nb$
$re_{9,i,k} \equiv \ (\quad)_{t_{k,1}+ng}(e_{i,t_{k,1}+ng})_{t_{k,5}} \longrightarrow (a_i)_{t_{k,1}+ng}(\quad)_{t_{k,5}}$
$: 0 \leq i \leq 1, 1 \leq k \leq nb$
$re_{10,i,k} \equiv (\quad)_{unt_{k-nb,1}+ng+nb}(e_{i,unt_{k-nb,1}+ng+nb})_{unt_{k-nb,3}} \longrightarrow$
$(a_i)_{unt_{k-nb,1}+ng+nb}(\quad)_{unt_{k-nb,3}}$
$: 0 \leq i \leq 1, nb+1 \leq k \leq nb+nu$

– Evaluation of the result of the interactions (1/2).

$rs_{11} \equiv binop_1\, {a_0}^2[]_1 \longrightarrow binop_1\, c_0[]_1$
$rs_{12} \equiv binop_1\, {a_1}^2[]_1 \longrightarrow binop_1\, c_1[]_1$
$rs_{13} \equiv binop_1\, a_1\, a_0[]_1 \longrightarrow binop_1\, c_1[]_1$
$rs_{14} \equiv binop_2\, {a_1}^2[]_1 \longrightarrow binop_2\, c_1[]_1$
$rs_{15} \equiv binop_2\, {a_0}^2[]_1 \longrightarrow binop_2\, c_0[]_1$
$rs_{16} \equiv binop_2\, a_1\, a_0[]_1 \longrightarrow binop_2\, c_0[]_1$
$rs_{17} \equiv binop_3\, {a_1}^2[]_1 \longrightarrow binop_3\, c_0[]_1$
$rs_{18} \equiv binop_3\, {a_0}^2[]_1 \longrightarrow binop_3\, c_0[]_1$
$rs_{19} \equiv binop_3\, a_1\, a_0[]_1 \longrightarrow binop_3\, c_1[]_1$
$rs_{20,i} \equiv unop_1\, a_i[]_1 \longrightarrow unop_1\, c_i[]_1 : 0 \leq i \leq 1$
$rs_{21,i} \equiv unop_2\, a_i[]_1 \longrightarrow unop_2\, c_{i-1}[]_1 : 0 \leq i \leq 1$
$rs_{22,i} \equiv unop_3\, a_i[]_1 \longrightarrow unop_3\, {c_i}^i[]_1 : 0 \leq i \leq 1$
$rs_{23,i} \equiv unop_4\, a_i[]_1 \longrightarrow unop_4\, {c_{1-i}}^i[]_1 : 0 \leq i \leq 1$

– Evaluation of the result of the interactions (2/2).

$re_{24,i,k} \equiv (c_i)_{t_{k,1}+ng}(\quad)_{t_{k,7}} \longrightarrow$
$(\quad)_{t_{k,1}+ng}(eF_{tk,8*i+(1-i)*(1-t_{k,8}),t_{k,1}+ng})_{t_{k,7}}$
$: 0 \leq i \leq 1, 1 \leq k \leq nb$
$re_{25,i,k} \equiv (c_i)_{unt_{k,1}+ng+nb}(\quad)_{unt_{k,5}} \longrightarrow$
$(\quad)_{unt_{k,1}+ng+nb}(eF_{i,(untk,1+ng+nb)})_{unt_{k,5}}$
$: 0 \leq i \leq 1, 1 \leq k \leq nu$

– Calculation of contributions.

$rs_{26,i,k} \equiv eF_{i,(t_{k,1}+ng)}[]_1 \longrightarrow {b_i}^{t_{k,9}}[]_1 : 0 \leq i \leq 1, 1 \leq k \leq nb$
$rs_{27,i,k} \equiv eF_{i,(unt_{k,1}+ng+nb)}[]_1 \longrightarrow {b_i}^{unt_{k,6}}[]_1 : 0 \leq i \leq 1, 1 \leq k \leq nu$

– Elimination of different–signed contributions.

$rs_{28} \equiv \ b_1\, b_0[]_1 \longrightarrow []_1$
$rs_{29,i} \equiv clock_{i-1}[]_1 \longrightarrow clock_i[]_1 : 1 \leq i \leq cc+3$

– Calculation of the next gene state.

$rs_{30} \equiv b_0[]_1^- \longrightarrow [d_0]_1^-$
$rs_{31} \equiv b_1[]_1^- \longrightarrow []_1^-$
$rs_{32,i,j,k} \equiv dest_{i,j,t_{k,1}+ng}[]_1^- \longrightarrow []_1^- : 0 \leq i \leq 1, 1 \leq j \leq ng, 1 \leq k \leq nb$
$rs_{33,i,j,k} \equiv dest_{i,j,unt_{k-nb,1}+ng+nb}[]_1^- \longrightarrow []_1^-$
$: 0 \leq i \leq 1, 1 \leq j \leq ng, nb+1 \leq k \leq nb+nu$
$rs_{34} \equiv [d_0]_1^- \longrightarrow []_1^+$
$rs_{35} \equiv clock_{cc+3}[]_1^+ \longrightarrow go\, a_0[]_1^0$
$rs_{36} \equiv clock_{cc+3}[]_1^- \longrightarrow go\, a_1[]_1^0$

In this section, only input parameters are described. This way, details about the model dynamics are left aside. These parameters are described in table 1.

Table 1. Parameters

Parameter	Description
	General parameters for the system
ng	Number of genes in the network
nb	Number of binary interactions
nu	Number of unary interactions
$threshold$	Maximum strength for an interaction
cc	Clock control
	Gene configuration parameters
$g_{i,1}$	Gene number (id)
$g_{i,3}$	Initial state of the gene
	Binary interactions parameters
$t_{i,1}$	Binary interaction number (id)
$t_{i,2}$	Interaction type (or: 1, and: 2, xor: 3)
$t_{i,3}$	1^{st} source gene number (id)
$t_{i,4}$	1^{st} source gene contribution (positive: 1, negative: 0)
$t_{i,5}$	2^{nd} source gene number (id)
$t_{i,6}$	2^{nd} source gene contribution (positive: 1, negative: 0)
$t_{i,7}$	Destination gene number (id)
$t_{i,8}$	Influence over destination gene (positive: 1, negative: 0)
$t_{i,9}$	Strength of the destination
	Unary interactions parameters
$unt_{i,1}$	Unary interaction number (id)
$unt_{i,2}$	Interaction type (strong promotion: 1, inhibition: 2; weak ones: 3, 4)
$unt_{i,3}$	Source gene number (id)
$unt_{i,4}$	Source gene contribution (positive, negative)
$unt_{i,5}$	Destination gene number (id)
$unt_{i,6}$	Influence over destination gene (positive, negative)

2.3 Model output

The state of the network is encoded as the multiplicity of objects a_1 and a_0 in each gene environment. The presence of objects a_1 inside a gene environment represents that its gene is active (a_0 for inactive). Due to the nature of the system, membrane genes cannot have objects a_1 and a_0 simultaneously. Therefore, to know the final state of the network, it suffices to identify which environments contain object a_1 and which ones a_0 at configuration T.

3 Modelling and simulation on MeCoSim

This section explains some relevant issues concerning the software environment, putting the focus on the needed changes in P-Lingua framework and the configuration of MeCoSim. The P–Lingua definition used to analyse the PDP model adheres to P–Lingua version 4 standard, available at [10].

3.1 Custom interface in MeCoSim

In our previous work [16], a P–Lingua model for the family of logic networks based on PDP systems has been extensively described. This model contains a number of parameters representing relevant information about each specific scenario. Thus, although a general model has been presented, a mechanism to ease the task of introducing the specific data for each scenario is needed. This task is performed through the software environment provided by MeCoSim [14, 12]. MeCoSim permits the definition of a custom visual simulator. This simulator includes an interface with the needed inputs, outputs, and a way to translate the input data into parameters for the model. The simulation engine is provided by pLinguaCore, available at [10]. The most relevant facts of this process are listed below.

Definition of a custom visual simulator for Logic Networks

Here, the process for defining a custom simulator based on MeCoSim is provided. This process is very simple, and consists of the following steps:

Configuration file: The first step is to define a spreadsheet file containing the configuration for the definition of visual tabs, input tables, output tables and charts, and the mechanism to generate both model parameters from input tables and outputs from the simulation results. The contents of the simulation parameters tab in the file is shown in figure 1. The configuration file is available by contacting the authors.

Param Name	Param Value	Index 1	Index 2
ng	<4,1,1>		
nb	<4,1,2>		
nu	<4,1,3>		
max	<4,1,4>		
cc	<4,1,5>		
g	<5,1,2>	[1..ng]	[1..2]
t	<2,1,2>	[1..nb]	[1..<@c,2>]
unt	<3,1,2>	[1..nu]	[1..<@c,3>]

Fig. 1. MeCoSim configuration file. Simulation params

Loading configuration file on MeCoSim: That file is loaded through the main window in MeCoSim by clicking the "Load config file" button, choosing the file, selecting "Update all information" option and pressing "Update config info" button. After these steps, the configuration file is loaded, so the custom simulator is ready to use. Finally, the message "The Application has been successfully initialized" is prompted in MeCoSim main display.

Running custom simulator: The newly configured simulator is ready to use by selecting "Gene network" application and pressing "Run Application" button. Then, the custom interface is visualized, enabling the user to load the model (*.pli* file) and enter the input data for a specific scenario, as shown in figure 2.

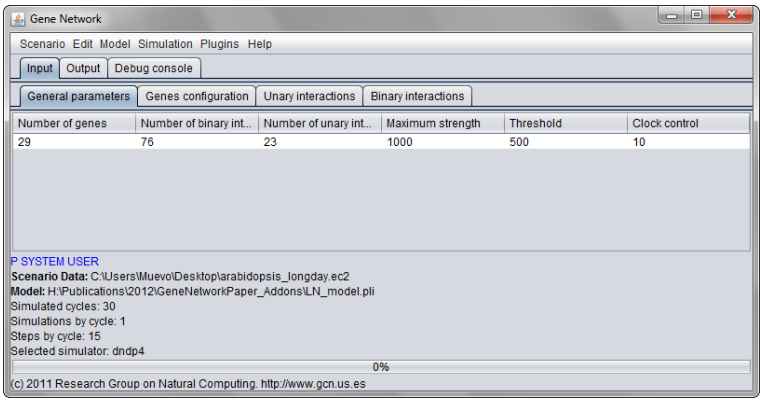


Fig. 2. MeCoSim window

3.2 Simulation methodology

In [16], we describe a methodology to simulate LN DP systems on MeCoSim. This methodology can be summarized in the following steps:

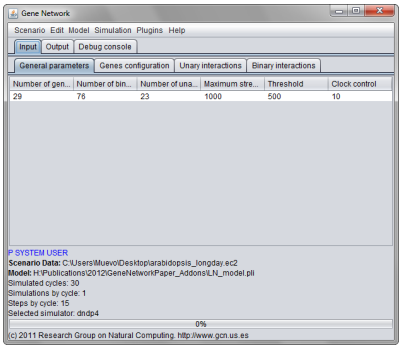
- Load the model specification by clicking on `Model > Set model`.
- Fill in the input tables in tab `Input`. Optionally, it is possible to save this data by clicking on `Scenario > Save`. This data can be loaded later by clicking on `Scenario > Open`.
- Set the number of steps on `Simulation > Number of steps`.
- Click on `Simulation > Simulate!`.
- Visualize the results in tab `Output`.

A toy example on a 3–gene logic network is provided in [16]. This network is taken from [15]. In this network, interactions have no associated weights. Hence, we presume all interactions to have the same weight (say 100). Although interaction scoring based on Pearson correlation coefficient is a rather widespread metric for measuring gene interaction strength [9], there is little literature on LNs, thus making it hard to find LN toy examples.

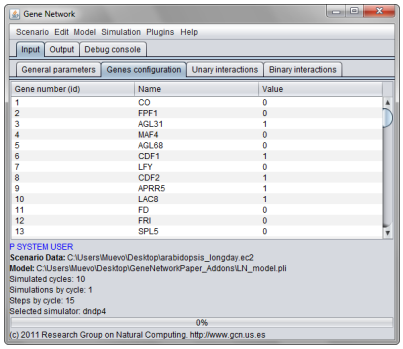
4 A case study on *Arabidopsis thaliana*

In order to experimentally verify our model, we have tested our algorithm by using a logic network which regulates flowering processes associated to *Arabidopsis thaliana* on a long day scenario. This relatively large network integrates gene interaction samples from NCBI/EBI database [5]. This logic network has been constructed according to the procedure described by Bowers *et al.* [2]. *A. thaliana* is a species widely used in genetic and protein interaction networks. The total number of genes in the network is 29, whereas the total number of interactions is 99. These interactions consist of 23 unary interactions and 76 binary interactions. We notice that only a few different types of all possible interactions are present in this network. In the case of unary interactions only strong promotions and strong inhibitions are present. When it comes to binary interactions, only *AND*–like and *OR*–like interactions are present. As regards to the distribution of the present interactions, the vast majority of them are *AND*–like interactions with both inputs in non–negated form (that is, $G'_j = G_j$ and $G'_k = G_k$), as well as non–negated result ($G'_l = G_l$).
Gene network data is provided as an appendix in section 7. Specifically, gene initial states are reflected in table 5. Unary gene interactions are reflected in figure 6. Similarly, binary gene interactions are reflected in figures 7, and 8. Figure 3 displays the MeCoSim input tables used in this case study.

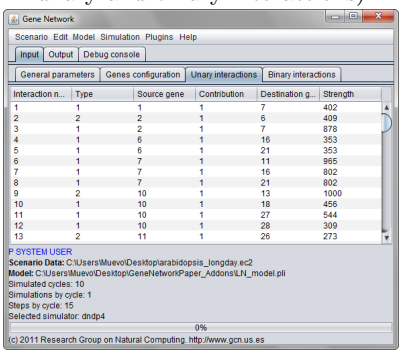
Eventually, we have simulated the corresponding P system for the *A. thaliana* network entered. The improved LAPP method (as presented in Wang et al. [18]) has been run for 30 steps on this data. Similarly, the LN DP model has been simulated for 30 cycles. As each cycle in an LN DP system consists of 15 computation steps, the total number of steps simulated in the model is $30 \times 15 = 450$. The results (see figure 4) match the ones obtained from the execution of the improved LAPP method on the same input data. Therefore, it is verified that, on this gene network and scenario, the P system model behaviour is analogous to that from the improved LAPP method.



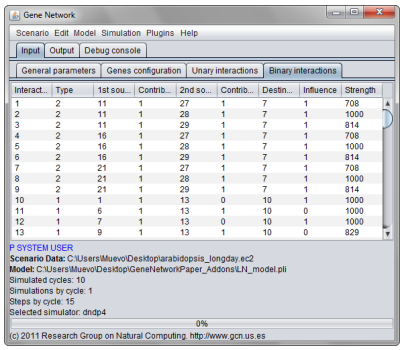
General parameters (that is, number of unary and binary interactions)



Initial state of each gene (active or inactive)



Unary interactions



Binary interactions

Fig. 3. Arabidopsis - MeCoSim Interface - Input Data

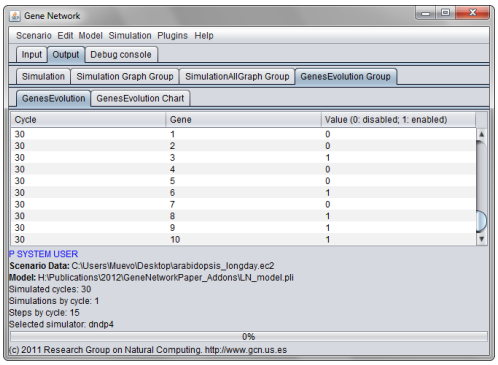


Fig. 4. Final gene states used for the simulation on MeCoSim interface

5 Conclusions

In this work, we have presented a case study on LN DP systems for a gene network which regulates the flowering process of *Arabidopsis thaliana*. We suplement this case study with a guide for generating a custom MeCoSim simulator for LN DP systems. In the case study, we validate the model against the improved LAPP method [18]. We conclude that our Membrane Computing model matches the output data obtained by the latter algorithm.
As a future work, it would be interesting to apply this model to gene networks with different biological functions, so as to test if the model matches the improved LAPP algorithm for a sufficiently representative number and variety of cases. This task can be complemented with a comparative study of the improved LAPP algorithm and different biochemical simulation methods (such as the Gillespie algorithm [6]) by means of Membrane Computing models.

6 Acknowledgements

Luis Valencia–Cabrera, Manuel García-Quismondo and Mario J. Pérez-Jiménez are supported by project TIN2012-37434 from "Ministerio de Ciencia e Innovación" of Spain, and "Proyecto de Excelencia con Investigador de Reconocida Valía P08-TIC-04200" from Junta de Andalucía, both co-financed by FEDER funds. Manuel García-Quismondo is also supported by the National FPU Grant Programme from the Spanish Ministry of Education.

References

1. Mitsutomo Abe, Yasushi Kobayashi, Sumiko Yamamoto, Yasufumi Daimon, Ayako Yamaguchi, Yoko Ikeda, Harutaka Ichinoki, Michitaka Notaguchi, Koji Goto, and Takashi Araki. (2005). Fd, a bzip protein mediating signals from the floral pathway integrator ft at the shoot apex. *Science*, 309(5737):1052–1056.
2. Peter M. Bowers, Shawn J. Cokus, Todd O. Yeates, and David Eisenberg. (2004). Use of logic relationships to decipher protein network organization. *Science*, 5705(306):2246–2249.
3. Peter M. Bowers, Brian D. O'Connor, Shawn J. Cokus, Eniat Sprinzak, Todd O. Yeates, and David Eisenberg. (2005). Utilizing logical relationships in genomic data to decipher cellular processes. *the FEBS journal*, 272(1):5110–5118.
4. Miguel A. Martínez del Amor, Ignacio Pérez-Hurtado, Manuel García-Quismondo, Luis F. Macías-Ramos, Luis Valencia-Cabrera, Álvaro Romero-Jiménez, Carmen Graciani-Díaz, Agustín Riscos-Núñez, M. Angels Colomer, and Mario J. Pérez-Jiménez. (2013). Dcba: Simulating population dynamics p systems with proportional objects distribution. *Lecture Notes in Computer Science*, pages 257–276.
5. National Center for Biotechnology Information, (March 2013). http://www.ncbi.nlm.nih.gov/.

6. Daniel T. Gillespie. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.
7. Takato Imaizumi, Thomas F. Schultz, Frank G. Harmon, Lindsey A. Ho, and Steve A. Kay. (2005). Fkf1 f-box protein mediates cyclic degradation of a repressor of constans in arabidopsis. *Science*, 309(5732):293–297.
8. Takato Imaizumi, Hien G. Tran, Trevor E. Swartz, Winslow R. Briggs, and Steve A. Kay. (2003). Fkf1 is essential for photoperiodic-specific light signaling in arabidopsis. *Nature*, 426:301–309. 10.1007/s11424-010-0205-0.
9. Bolan Linghu, EricA. Franzosa, and Yu Xia. (2013). Construction of functional linkage gene networks by data integration. In Hiroshi Mamitsuka, Charles DeLisi, and Minoru Kanehisa, editors, *Data Mining for Systems Biology*, volume 939 of *Methods in Molecular Biology*, pages 215–232. Humana Press.
10. P Lingua Web Page, (February 2009). http://www.p-lingua.org.
11. Miguel A. Martínez-del Amor, Ignacio Pérez-Hurtado, Adolfo Gastalver-Rubio, Anne C. Elster, and Mario J. Pérez-Jiménez. (2012). Population dynamics p systems on cuda. In David Gilbert and Monika Heiner, editors, *Computational Methods in Systems Biology*, Lecture Notes in Computer Science, pages 247–266. Springer Berlin Heidelberg.
12. MeCoSim Web Page, (July 2010). http://www.p-lingua.org/mecosim.
13. Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa. (2010). *The Oxford Handbook of Membrane Computing*. Oxford University Press, Inc., New York, NY, USA.
14. Ignacio Pérez-Hurtado, Luis Valencia-Cabrera, Mario J. Pérez-Jiménez, M. A. Colomer, and Agustín Riscos-Núñez. (2010). MeCoSim: a general purpose software tool for simulating biological phenomena by means of P systems. *IEEE Fifth International Conference on Bio-inpired Computing: Theories and Applications (BIC-TA 2010)*, I:637–643.
15. Thomas Schlitt and Alvis Brazma. (2007). Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, 8(Suppl 6):S9.
16. Luis Valencia-Cabrera, Manuel García-Quismondo, Yansen Su, Mario J. Perez-Jimenez, Hui Yu, and Linqiang Pan. (2013). Modeling logic gene networks by means of probabilistic dynamic p systems, (accepted publication). *International Journal of Unconventional Computing*.
17. Shudong Wang, Yan Chen, Qingyun Wang, Eryan Li, Yansen Su, and Dazhi Meng. (2006). Analysis for stimuli to shoot genes of arabidopsis thaliana based on logical relationships. *Optimization and Systems Biology*, 11(-):435–447.
18. Shudong Wang, Yan Chen, Qingyun Wang, Eryan Li, Yansen Su, and Dazhi Meng. (2010). Analysis for gene networks based on logic relationships. *Journal of Systems Science and Complexity*, 23:999–1011. 10.1007/s11424-010-0205-0.
19. Su-Zhil Zhang and Zuo Jian-Rhu. (2006). Advance in the flowering time control of arabidopsis. *Progress in Biochemistry and Biophysics*, 33:301–309.
20. Xin Zhang, Seungchan Kim, Tie Wang, and C. Baral. (October 2006). Joint learning of logic relationships for studying protein function using phylogenetic profiles and the rosetta stone method. *Trans. Sig. Proc.*, 54(6):2427–2435.

7 Appendix A: Gene Network Data

Gene number	Initial state
1	0
2	0
3	1
4	0
5	0
6	1
7	0
8	1
9	1
10	1
11	0
12	0
13	0
14	1
15	1

Gene number	Initial state
16	0
17	1
18	1
19	1
20	0
21	0
22	1
23	1
24	0
25	1
26	0
27	1
28	1
29	1

Fig. 5. Initial gene states in the *Arabidosis thaliana* gene network on the longday scenario taken as case study

ID	Logic	Weight
1	$g_1 \rightarrow g_7$	0.402
2	$g_2 \rightarrow \neg g_6$	0.409
3	$g_2 \rightarrow g_7$	0.878
4	$g_6 \rightarrow g_{16}$	0.353
5	$g_6 \rightarrow g_{21}$	0.353
6	$g_7 \rightarrow g_{11}$	0.965
7	$g_7 \rightarrow g_{16}$	0.802
8	$g_7 \rightarrow g_{21}$	0.802
9	$g_{10} \rightarrow \neg g_{13}$	0.1000
10	$g_{10} \rightarrow g_{18}$	0.456
11	$g_{10} \rightarrow g_{27}$	0.544
12	$g_{10} \rightarrow g_{28}$	0.309

ID	Logic	Weight
13	$g_{11} \rightarrow \neg g_{26}$	0.273
14	$g_{12} \rightarrow g_{16}$	0.282
15	$g_{12} \rightarrow g_{21}$	0.282
16	$g_{16} \rightarrow \neg g_{29}$	0.713
17	$g_{17} \rightarrow g_{24}$	0.425
18	$g_{17} \rightarrow g_{26}$	0.389
19	$g_{19} \rightarrow g_{29}$	0.551
20	$g_{20} \rightarrow \neg g_{22}$	0.303
21	$g_{21} \rightarrow \neg g_{29}$	0.713
22	$g_{22} \rightarrow g_{26}$	0.439
23	$g_{28} \rightarrow g_{29}$	0.292

Fig. 6. Unary gene interactions present in the logic network associated to the behaviour of *Arabidosis thaliana* taken as case study

ID	Logic	Weight
1	$g_{11} \wedge g_{27} \rightarrow g_7$	0.708
2	$g_{11} \wedge g_{28} \rightarrow g_7$	1
3	$g_{11} \wedge g_{29} \rightarrow g_7$	0.814
4	$g_{16} \wedge g_{27} \rightarrow g_7$	0.708
5	$g_{16} \wedge g_{28} \rightarrow g_7$	1
6	$g_{16} \wedge g_{29} \rightarrow g_7$	0.814
7	$g_{21} \wedge g_{27} \rightarrow g_7$	0.708
8	$g_{21} \wedge g_{28} \rightarrow g_7$	1
9	$g_{21} \wedge g_{29} \rightarrow g_7$	0.814
10	$g_1 \vee \neg g_{13} \rightarrow g_{10}$	1
11	$g_6 \wedge g_{13} \rightarrow \neg g_{10}$	1
12	$g_7 \vee \neg g_{13} \rightarrow g_{10}$	1
13	$g_9 \wedge g_{13} \rightarrow \neg g_{10}$	0.829
14	$g_{11} \vee \neg g_{13} \rightarrow g_{10}$	1
15	$g_{12} \vee \neg g_{13} \rightarrow g_{10}$	0.829
16	$\neg g_{13} \vee g_{16} \rightarrow g_{10}$	1
17	$\neg g_{13} \vee g_{18} \rightarrow g_{10}$	0.728
18	$g_{13} \wedge g_{19} \rightarrow \neg g_{10}$	0.829
19	$\neg g_{13} \vee g_{21} \rightarrow g_{10}$	1
20	$\neg g_{13} \vee g_{27} \rightarrow g_{10}$	1
21	$g_{27} \vee \neg g_{28} \rightarrow g_{10}$	0.728
22	$g_{10} \wedge g_{16} \rightarrow g_{11}$	0.741
23	$g_{10} \wedge g_{21} \rightarrow g_{11}$	0.741
24	$g_{14} \wedge g_{16} \rightarrow g_{11}$	0.741
25	$g_{14} \wedge g_{21} \rightarrow g_{11}$	0.741
26	$g_{15} \wedge g_{16} \rightarrow g_{11}$	0.741
27	$g_{15} \wedge g_{21} \rightarrow g_{11}$	0.741
28	$g_{16} \wedge g_{17} \rightarrow g_{11}$	0.741
29	$g_{16} \wedge \neg g_{20} \rightarrow g_{11}$	0.741
30	$g_{16} \wedge g_{21} \rightarrow g_{11}$	0.741

ID	Logic	Weight
31	$g_{16} \wedge g_{22} \rightarrow g_{11}$	0.741
32	$g_{16} \wedge g_{23} \rightarrow g_{11}$	0.741
33	$g_{16} \wedge \neg g_{24} \rightarrow g_{11}$	0.741
34	$g_{16} \wedge g_{25} \rightarrow g_{11}$	0.741
35	$g_{16} \wedge \neg g_{26} \rightarrow g_{11}$	0.741
36	$g_{16} \vee g_{29} \rightarrow g_{11}$	0.741
37	$g_{17} \wedge g_{21} \rightarrow g_{11}$	0.741
38	$\neg g_{20} \wedge g_{21} \rightarrow g_{11}$	0.741
39	$g_{21} \wedge g_{22} \rightarrow g_{11}$	0.741
40	$g_{21} \wedge g_{23} \rightarrow g_{11}$	0.741
41	$g_{21} \wedge \neg g_{24} \rightarrow g_{11}$	0.741
42	$g_{21} \wedge g_{25} \rightarrow g_{11}$	0.741
43	$g_{21} \wedge \neg g_{26} \rightarrow g_{11}$	0.741
44	$g_{21} \vee \neg g_{29} \rightarrow g_{11}$	0.741
45	$g_8 \wedge g_{21} \rightarrow g_{16}$	0.801
46	$g_{10} \wedge g_{21} \rightarrow g_{16}$	1
47	$g_{11} \vee g_{21} \rightarrow g_{16}$	1
48	$g_{11} \vee \neg g_{29} \rightarrow g_{16}$	1
49	$g_{14} \wedge \neg g_{19} \rightarrow g_{16}$	0.801
50	$g_{14} \wedge g_{21} \rightarrow g_{16}$	1
51	$g_{15} \wedge g_{21} \rightarrow g_{16}$	1
52	$g_{17} \wedge g_{21} \rightarrow g_{16}$	1
53	$\neg g_{19} \wedge g_{21} \rightarrow g_{16}$	0.801
54	$\neg g_{20} \wedge g_{21} \rightarrow g_{16}$	1
55	$g_{21} \wedge g_{22} \rightarrow g_{16}$	1
56	$g_{21} \wedge g_{23} \rightarrow g_{16}$	1
57	$g_{21} \wedge \neg g_{24} \rightarrow g_{16}$	1
58	$g_{21} \wedge g_{25} \rightarrow g_{16}$	1
59	$g_{21} \wedge \neg g_{26} \rightarrow g_{16}$	1
60	$g_{21} \vee \neg g_{29} \rightarrow g_{16}$	1

Fig. 7. Binary gene interactions present in the logic network associated to the behaviour of *Arabidosis thaliana* taken as case study (1/2)

ID	Logic	Weight
61	$g_8 \wedge g_{16} \rightarrow g_{21}$	0.801
62	$g_{10} \wedge g_{16} \rightarrow g_{21}$	1
63	$g_{11} \vee g_{16} \rightarrow g_{21}$	1
64	$g_{11} \vee \neg g_{29} \rightarrow g_{21}$	1
65	$g_{14} \wedge g_{16} \rightarrow g_{21}$	1
66	$g_{14} \wedge \neg g_{19} \rightarrow g_{21}$	0.801
67	$g_{15} \wedge g_{16} \rightarrow g_{21}$	1
68	$g_{16} \wedge g_{17} \rightarrow g_{21}$	1
69	$g_{16} \wedge \neg g_{19} \rightarrow g_{21}$	0.801
70	$g_{16} \wedge \neg g_{20} \rightarrow g_{21}$	1
71	$g_{16} \wedge g_{22} \rightarrow g_{21}$	1
72	$g_{16} \wedge g_{23} \rightarrow g_{21}$	1
73	$g_{16} \wedge \neg g_{24} \rightarrow g_{21}$	1
74	$g_{16} \wedge g_{25} \rightarrow g_{21}$	1
75	$g_{16} \wedge \neg g_{26} \rightarrow g_{21}$	1
76	$g_{16} \vee \neg g_{29} \rightarrow g_{21}$	1

Fig. 8. Binary gene interactions present in the logic network associated to the behaviour of *Arabidosis thaliana* taken as case study (2/2)

Gene number	Initial state
1	0
2	0
3	1
4	0
5	0
6	1
7	0
8	1
9	1
10	1
11	0
12	0
13	0
14	1
15	1
16	0
17	1
18	1
19	1
20	0
21	0
22	1
23	1
24	0
25	1
26	1
27	1
28	1
29	1

Fig. 9. Final gene states in the *Arabidosis thaliana* gene network on the longday scenario taken as case study