CellLine, a stochastic cell lineage simulator: Applications

Andre S. Ribeiro, Daniel A. Charlebois, Jason Lloyd-Price

July 23, 2007

1 Introduction

Here we present a complete description of the example models that we tested and the results we obtained using *CellLine*. We also provide all reactions files (see Availability section).

To demonstrate the use of *CellLine*, we first simulate a model of the P53-Mdm2 feedback loop, where transcription and translation are modelled by multiple time delayed reactions [5]. This model is a modified version of a model proposed by Stolovitzky et al, [3]. Using this model, we reproduce the experimentally observed dynamics of single cells subject to ionizing radiation (IR), and the dynamics of cell lineages to reproduce recent observations that show that the oscillations of P53 and Mdm2 present in mother cells during division propagate to daughter cells. Also, it is shown that the average dynamics of multiple cells is in agreement with experimental observations.

Next, we use *CellLine* to simulate the dynamics of a commonly studied gene network, the repressilator [24]. The repressilator consists of a loop of three genes. The protein from the first gene inhibits the second gene, whose protein product in turn inhibits the third gene, and finally the protein from the third gene inhibits the first gene, completing the cycle.

To exemplify how *CellLine* can add substances (including gene promoter regions) in the simulation at runtime, we simulate a cell lineage where the mother cell initially consists of a 3 gene repressilator. One of the genes of the repressilator has then a probability of being duplicated at any given moment in the simulation. Additionally, it is shown how these events are passed on to the daughter cells and future generations of the cells where it occurs.

If an extra gene is added to the loop, becoming a 4 gene repressilator, the system behavior changes dramatically, becoming bistable [25]. Here, we model a cell lineage's dynamics where each cell consists of a 4 gene repressilator and show how one can use *CellLine* to simulate a model of cells of a lineage that can dynamically evolve in a such a way that some branches of the lineage will reach one "stable" state, and the other branches, the opposite "stable" state. Also, we use a modified version of this model to demonstrate how any given initial state can be imposed on the cells of lineages.

For our final example we chose to simulate the dynamics of a recently uncovered gene network [26] that controls the choice between pale versus yellow ommatidia in *Drosophila melanogaster*. It was reported that the Drosophila dioxin receptor Spineless is both necessary and sufficient for the formation of the ommatidial mosaic. This study suggests that the retinal mosaic required for color vision is defined by the stochastic expression of a single transcription factor, Spineless (SS) [26]. The stochastic expression of SS acts as a binary switch in terms of determining the fate of the cells. Here, we implement a model that accounts for all the features observed experimentally.

This model is an example of a gene regulatory system capable of probabilistic regulation that leads to a differentiated state, out of two possible fates, which remains stable from there on.

All simulations were made on a Pentium 4 at 3GHZ with 1GB of RAM.

2 The P53-Mdm2 negative feedback loop.

The tumor suppressor protein P53 has a fundamental role in cellular response to a variety of environmental stresses that can affect DNA structure and replication. Under stress, P53 can induce cell cycle arrest, DNA repair, and apoptosis. Depending on the causes of stress, P53 can activate several genes responsible for regulating processes such as cell cycle arrest, DNA repair, and apoptosis [16]. Mutations in the gene that transcribes p53 RNA have been found in about 50% of human tumors [17].

Under normal conditions, P53 concentrations are kept low by an Mdm2 protein [19], a down regulator of P53. These two proteins form a negative feedback loop responsible for the oscillatory dynamics in their concentrations in cells exposed to radiation that induces DNA damage [18]. When under stress, P53 concentration can rapidly increase up to 16 times the basal concentration [20].

Here we focus on mimicking experimental measurements of P53 and Mdm2 concentrations in the nucleus of individual cells following ionizing irradiation.

These observations in populations of cells subject to irradiation, show that the number of DNA double strand breaks (DSBs) in the cells can be assumed to follow a Poisson distribution whose average is proportional to the radiation dose [15, 3]. In agreement, DSBs are treated as a "chemical species" and are inserted in the system at user defined times of a simulation, in a quantity randomly generated from a Poisson distribution. A file named "poisson.lua" provides these quantities when called from the reactions files. Also, once the DSBs are repaired, the oscillations cease in a time interval that can vary from 12 to 24 hours [2].

P53 and Mdm2 oscillations have an approximately constant frequency before stopping. The number of oscillations vary from cell to cell and, although a damped oscillation of P53 and Mdm2 is observed in the cell population average, in single cell measurements these oscillations are only slightly damped and appear to cease abruptly [3]. The number of oscillations before stoping varies significantly from cell to cell [3].

Finally, we point out that in the stochastic framework, the entire system of reactions can be slowed down or sped up by varying the assumed volume or temperature. Therefore, e.g., given a substance whose quantity oscillates periodically, a specific period can be imposed by varying one of these two parameters [9, 10].

For example, given a temperature T and a volume V, if the rate constant of a bimolecular reaction is equal to 1 s^{-1} (where s stands for second), that reaction is expected to occur at the rate of 1 per second, times the quantities of the reactants. Doubling the volume will decrease its propensity to half that value (as it was demonstrated by Gillespie in [9, 10], under the SSA assumptions). For this reason, it is possible to match any simulation time to the time duration of any given experiment.

2.1 Single Cell model of the P53-Mdm2 negative feedback loop.

The model of the P53-Mdm2 feedback loop here simulated consists of a "reduced version" of the ODE model proposed in [3].

The original model consists of 3 modules, connected by the sharing of a single chemical substance, produced in one module and acting as reactant in the next. Since we focus on the P53-Mdm2 loop's dynamics, we do not include modules 1 and 2. However, we do include DSBs as the initiators of the P53-Mdm2 oscillations.

First, DSBs are added at user defined moments of the simulation following a Poisson distribution. A first order reaction of the DSBs decay (representing that they have been repaired) is added to the model to mimic the dynamics of the DSBs' quantity in time, reported in [3].

The oscillations of P53 and Mdm2 are activated only when DSBs appear in the system, that is, they will not begin until that moment and will die out some time after DSBs are removed from the

system.

Cells suffer complex and simple DSBs when irradiated by IR [3]. These two kinds of breaks have different effects in modules 1 and 2 but similar effects in module 3. Thus we tested the dynamical response of module 3 to a single quantity ("DSB"), which is the sum of both types of DSBs. The amount of DSBs to be introduced is randomly generated from a Poisson distribution with a mean of 50 (35 for simples breaks and 15 for complex breaks) since the sum of two random variables following a Poisson distribution also follows a Poisson distribution where the mean is the sum of the means of the two initial distributions.

To do this, *CellLine* calls the function "poisson" from a Lua file (also provided by us) named "poisson.lua", that should be in the same directory as the reactions file and DynSim executable.

It has been shown that transcription and translation are more correctly modelled as multiple time-delayed events [5], rather than instantaneous reactions. In agreement, the reactions responsible for creating p53 RNA, P53, mdm2 RNA, and Mdm2, are here modelled as multiple time-delayed reactions. The promoter regions of the genes from which p53 RNA and mdm2 RNA are transcribed from are explicitly represented by ProP53 and ProMdm2, respectively.

For a transcription reaction to occur, an RNA polymerase (RNAp) must bind to a gene promoter region. While the RNAp is bound to the promoter, no other molecule can bind to it. Therefore, each time a transcription reaction occurs, the promoter is only released from the waitlist after a small time delay, once again becoming available for transcription. RNAp's are not explicitly represented since we assume they are never depleted and exist in a sufficient quantity so that the fluctuations of the total number of RNAps do not affect any reaction propensity [10]. To produce an RNA, the RNAp must process the set of nucleotides of the gene. Thus, a time delay for the completion of the RNA production is also introduced (much larger than the delay on the promoter and depending on the gene length) [5]. In general, a transcription reaction of gene A, with a rate constant k_{transc}, where the promoter region of A is released after τ_A seconds, and the RNA (RNA_A) is complete after τ_{rna_A} seconds, is here represented by (eq. 1):

$$Pro_A \xrightarrow{k_{transc}} Pro_A(\tau_A) + RNA_A(\tau_{rna_A}) \tag{1}$$

Here we assume that translation of the RNA can only begin as soon as the RNA has been completely transcribed. In Prokaryotes, translation can begin as soon as the initial codons of the RNA have been transcribed, since they constitute the ribosome binding site (RBS) region. For Eukaryotes, one needs to account for the time it takes for the RNA to leave the nucleus so that it can be translated by ribosomes. We make the same assumption that we did for RNAp's, regarding the number of existing ribosomes, and so in general, a translation reaction can be modelled as:

$$RNA_A \xrightarrow{k_{transl}} RNA_A(\tau^{RBS}_{rna_A}) + p_A(\tau_{p_A}) \tag{2}$$

In eq. 2, k_{transl} is the rate constant of translation, p_A is the protein A that results from translating the RNA_A, $\tau^{RBS}_{rna_A}$ is the time delay for releasing the ribosome binding site of RNA_A, and τ_{p_A} is the time it takes for protein A to be translated and folded (becoming available for future reactions) after the reaction begins.

In the P53-Mdm2 loop model there are two genes, one responsible for the transcription of p53 RNA and the other of mdm2 RNA. Our model consists of the following reactions:

$$DSB \xrightarrow{DSB_{decay}} \emptyset \tag{3}$$

$$ProP53 \xrightarrow{Sp53/5} ProP53(\tau_{ProP53}) + pr53(\tau_{pr53}) \tag{4}$$

$$ProP53 +^* DSB \xrightarrow{4 \cdot Sp53} ProP53(\tau_{ProP53}) + pr53(\tau_{pr53}) \tag{5}$$

$$pr53 \xrightarrow{Gp53} \emptyset \tag{6}$$

$$pr53 \xrightarrow{rp53} pr53(\tau_{p53}^{RBS}) + P53(\tau_{P53}) \tag{7}$$

$$P53 \xrightarrow{up53} \emptyset \tag{8}$$

$$P53 +^{*} MDM2 \xrightarrow{vp53} \emptyset \tag{9}$$

$$ProMdm2 \xrightarrow{Smdm2/10} ProMdm2(\tau_{ProMdm2}^{1}) + mdm2(\tau_{mdm2}) \tag{10}$$

$$ProMdm2 +^{*} P53 \xrightarrow{Smdm2} ProMdm2(\tau_{ProMdm2}^{2}) + mdm2(\tau_{mdm2}) \tag{11}$$

$$ProMdm2 +^{*} DSB \xrightarrow{2 \cdot Smdm2} ProMdm2(\tau_{ProMdm2}^{3}) + mdm2(\tau_{mdm2}) \tag{12}$$

$$mdm2 \xrightarrow{Gmdm2} \emptyset \tag{13}$$

$$mdm2 \xrightarrow{rmdm2} mdm2(\tau_{mdm2}^{RBS}) + MDM2(\tau_{MDM2}) \tag{14}$$

$$MDM2 \xrightarrow{umdm2} \emptyset \tag{15}$$

$$^{*}P53 + MDM2 \xrightarrow{vp53} \emptyset \tag{16}$$

The rate constants are set at: $DSB_{decay} = 0.003\ s^{-1}$, Sp53 = 0.02 s^{-1}, Gp53 = 0.02 s^{-1}, Smdm2 = 0.045 s^{-1}, Gmdm2 = 0.02 s^{-1}, rp53 = 0.6 s^{-1}, up53 = 0.02 s^{-1}, vp53 = 9.2 s^{-1}, rmdm2 = 0.04 s^{-1}, and umdm2 = 0.14 s^{-1}.

Here, we set the time delays as: $\tau_{mdm2} = 100s$, $\tau_{MDM2} = 10s$, $\tau_{pr53} = 100s$, $\tau_{P53} = 10s$, $\tau_{ProP53} = 1s$, $\tau_{ProMdm2}^{1} = 1s$, $\tau_{ProMdm2}^{2} = 0.01s$, $\tau_{ProMdm2}^{3} = 0.05s$, $\tau_{mdm2}^{RBS} = 1s$, and $\tau_{p53}^{RBS} = 0.1s$.

We also set the following initial quantities: P53 = 0, mdm2 = 0 (mdm2 RNA), MDM2 = 0, pr53 = 0 (p53 RNA), ProP53 = 1 (promoter region of the gene from which the p53 RNA is transcribed), and ProMdm2 = 1 (promoter region of the gene from which the mdm2 RNA is transcribed).

Using *CellLine* the delays in transcription and translation can be drawn from distributions instead of constant values (please refer to the Manual for details), for example, normal distributions as suggested in [1]. If the standard deviations are small relative to the mean, the behavior of the system will not change significantly [1]. For large standard deviations, the dynamics will become more "noisy" and oscillations less robust (data not shown).

Take reaction 4 as an example of how multiple time-delayed reactions are dealt with by *CellLine*. When this reaction is chosen to occur, ProP53 (p53 promoter) and the pr53 (p53 RNA) that will be produced are placed on a waitlist of events. The waitlist stores what elements are to be released and the time at which they should be released [5]. Once released, it is available for reactions.

In this case, ProP53 is released back into the system and is again able to express, τ_{ProP53} seconds after reaction 4 occurs. After τ_{pr53} seconds (larger than τ_{ProP53}), a pr53 is released into the system. This pr53 can then be translated into a P53 protein by ribosomes if reaction 7, representing translation, is selected to occur by the SSA. Once the translation process is complete (τ_{P53} seconds after it was chosen to occur), a P53 protein is released into the system.

A similar set of reactions (10) and (14) exist for mdm2 RNA transcription from ProMdm2 and Mdm2 translation from mdm2 RNA.

Additionally, we added transcription reactions to model the effect of the presence of some substances known to be directly or indirectly responsible for activating or repressing the expression of one of the two proteins (P53 and Mdm2). For example, reaction 12 models the activation of Mdm2 transcription due to the presence of DSBs in the system, since it is known that when DSBs exist, signaling molecules detect them and will then begin a cascade of events that will eventually lead to a higher expression of P53 and Mdm2 [3, 2]. Also, this reaction rate constant is set at a higher value than the rate constant of reaction 10, which models the basal level of transcription of mdm2 RNA (no activation required).

In Fig. 1 we show the results of a single simulation of the system of reactions described. Within a directory that contains NCellsGen, DynSim, Model7.g, and poisson.lua (to generate the DSBs from a Poisson distribution), the model was simulated by executing the following instruction which specifies the initial reactions filename, the number of cells one wants to model, total simulation time, and sampling time of the system state (in this order): "NCellsGen Model7.g 1 5000 10".

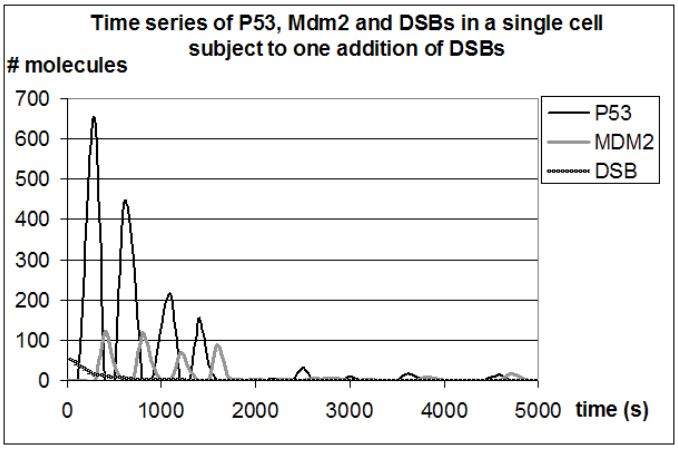

Figure 1: One time series of P53, Mdm2 and DSB's. Sampling period is 10 s. A single introduction of DSBs is made at t = 0 s.

Qualitatively, the results (Fig. 1) are in agreement to those previously reported [3], in the number of oscillations as a response to a single addition of DSBs, in the P53 and Mdm2 relative peak intensity, and in the time interval between the peaks of the two substances. When DSBs are introduced, P53 and Mdm2 can oscillate between 1 and 4 times. Also, the oscillations are damped and their ending is rather abrupt. Once the oscillations stop, only the addition of more DSBs can "restart" the oscillations of P53 and Mdm2.

Interestingly, the system responds quite diversely to each addition of DSBs, in amplitude and number of oscillations. The similarity between different responses is the period of the oscillations, in agreement with experimental observations [3]. The number and intensity of oscillations also depends, on average, on the number of DSBs added. That is, adding more DSBs will originate a stronger response on average. Also, if the DSBs have a weaker decay, resulting in them remaining in the system for a longer time interval, the oscillations will also last longer.

However, we notice that the multiple delayed reactions made the system's dynamics less susceptible to small variations in other parameters, such as rate constants. The number of oscillations as a response to the addition DSBs usually varies between 2 and 5 in agreement with experimental observations [2].

To attain a behavior such as the one reported in [2] (where the Mdm2 protein exists in the same quantities as the P53), one can increase the rate constants that control the conversion of mdm2 RNA into Mdm2 protein, so that the protein becomes the one existing in larger quantities. Here we opted to maintain mdm2 RNA as the most abundant substance of the two [3].

The behavior of this model, namely the number of observed oscillations, is highly dependent on the presence of DSBs. To increase the number of observed oscillations after a single addition of DSBs, all that is necessary is to decrease the decay rate of the DSBs. As soon as no more DSBs are present in the system, P53 and Mdm2 will only oscillate from 1 to 4 times.

2.2 Adding substances at runtime.

To demonstrate the ability of *CellLine* to add substances into the system at runtime, in Fig. 2 we show the time series of P53 and Mdm2 resulting from the introduction of DSBs in the system at two separate moments (reactions file "Model7b.g"). The P53-Mdm2 system responded both times, mimicking the P53-Mdm2 loop's behavior in real cells subject to several doses of irradiation (Fig. 2). The model was simulated by executing the following instruction: "NCellsGen Model7b.g 1 10000 10".

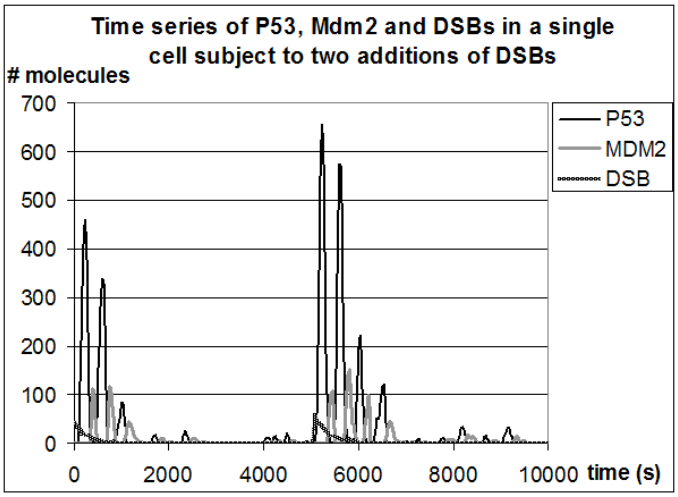

Figure 2: One time series of the P53-Mdm2 model where DSBs are introduced at two moments in the system: at t = 0 s and t = 5 000 s. Sampling period is 10 s.

Increasing the delay associated with transcription will cause a decrease in P53 and Mdm2 quantities and in the number of oscillations. The simulations in Figs. 1 and 2 took ~ 2 seconds each.

2.3 Simulating multiple identical cells.

As an example of using the "NCellsGen" module to generate a set of independent cells, we generated 10 cells where each contains an identical P53-Mdm2 oscillatory network (see supplementary material reactions file Model8_Multiple.g).

For this and the following models we set the rate of DSB decay to 0.001 s^{-1}, instead of 0.003 s^{-1} previously used. Additionally we changed the following delays to: $\tau_{mdm2} = 200s$, $\tau_{MDM2} = 30s$, $\tau_{pr53} = 200s$, and $\tau_{P53} = 30s$. The consequences of these changes were an increase in the average number of oscillations from 3 to 5 for each introduction of DSBs following the distribution described above, and an increase in the period of the oscillations, caused by the overall "slowing down" of the dynamics. The increase in the number of oscillations makes it easier to observe the wide range of distinct behaviors (in the number of oscillations of P53-Mdm2 due to a single addition of DSBs) that can arise from a population of identical cells. The overall slowing down allows the P53-Mdm2 oscillations to propagate to the third generation of cells using CellLineGen, which otherwise quickly dampens out in the first or second generation.

We set the initial reactions file, "Model8_Multiple.g", such that each cell will be subject to a different irradiation dose, i.e., a distinct number of DSBs in each cell. This is achieved by generating the number of DSBs from a Poisson distribution at run time. Additionally, since each cell's reactions file is provided a unique random seed, their dynamics will differ.

Thus, this case is an example of how to use *CellLine* to simulate multiple cells, all identical in

terms of their reactions and initial quantities of substances, but where each cell is subject to a distinct external perturbation. The initial cell can be, for example, a cell obtained in a given cell line generation for which more data is desired by the user.

The model was simulated by executing the following instruction: "NCellsGen Model8_Multiple.g 10 10000 10".

From the initial cell, NCellsGen made 10 copies and simulated their dynamics for 10000 s each, with a sampling period of 100 s. Each cell is subject to a single perturbation (by the addition of DSBs at t = 0 s). As a result, *CellLine* outputs 20 files, consisting of 10 reactions files named "cell_[cell index].g" and one time series for each reactions file, named "out_[cell index].xls".

Observing each cell independently it is visible that the P53-Mdm2 dynamics varies significantly from cell to cell, in the amplitude of the peaks and the number of oscillations. Very rarely is there no response of P53 or Mdm2 to the addition of DSBs.

In Fig. 3 we plot the average number of P53 proteins in all 10 cells. The result is a damped oscillation, such that the amplitude decreases almost at the same rate, from one oscillation to next the one, in agreement with measurements from multiple cells [1].

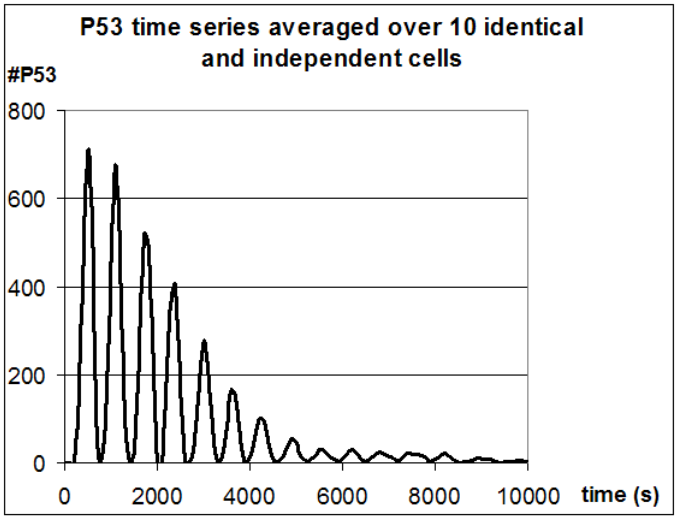

Figure 3: Average concentration of P53 over 10 cells. Sampling period is 10 s.

This simulation took less than 20 s (each cell's dynamics took approximately 2 s).

2.4 Example of a Cell Lineage of a cell containing a P53-Mdm2 loop.

Here we demonstrate the use of *CellLine* to create a cell lineage from an initial mother cell which contains the P53-Mdm2 network. Mother and daughter cells have the same chemical reactions. The initial cell is identical to the cell of the previous subsection.

Within a directory that contains CellLineGen, DynSim, Model9_Line.g, and poisson.lua, the model was simulated by executing the following instruction, which specifies the initial reactions filename, the number of cell generations one wants to model, total simulation time of each cell, and sampling time of the system state (in this order): "CellLineGen Model9_Line.g 3 1000 10".

As a result, CellLineGen creates a cell lineage with 3 generations and simulates the dynamics of each cell for 1000 s with a sampling period of 10 s. It outputs 3 files for each cell in the lineage: i) a reactions file named "Gener_[generation index]_Cell_[cell index].g", with the set of reactions of the cell from generation "*generation index*" and index within that generation "*cell index*", ii) a "snapshot"

of the cell state when its division occurred, and from which the daughter cells are created, called "Gener_[generation index]_cell_[cell index]_snap.g", and iii) one time series for each cell in files called "Gen_[generation index]_cell_[cell index]_out.xls".

Notice that CellLineGen will keep track of the "total experiment time", that is, for example cell (1,1) will exist between 0 and 1000 s, cells (2,1) and (2,2) between 1000 s and 2000 s, and so on. This is very important, especially if one wants to set perturbations to the cells of a certain generation at a given time. For example, to add DSBs in the cells of the 3^{rd} generation, DSBs should be added to the system between 2000 s and 3000 s (since each cell's life time is 1000 s). That information must be present in the mother cell's reactions file, after the block "reactions". In this case, for example, to add 10 DSBs in all cells at t = 2500 s, one would add the following instruction in the initial cell's reactions file, after the reactions block: "queue [10]DSB(2500);".

In our simulation, only the mother cell is subject to an initial perturbation (by the addition of DSBs following a Poisson distribution at t = 0 s). The time series of the P53 protein of cells of the lineage are shown in Fig. 4.

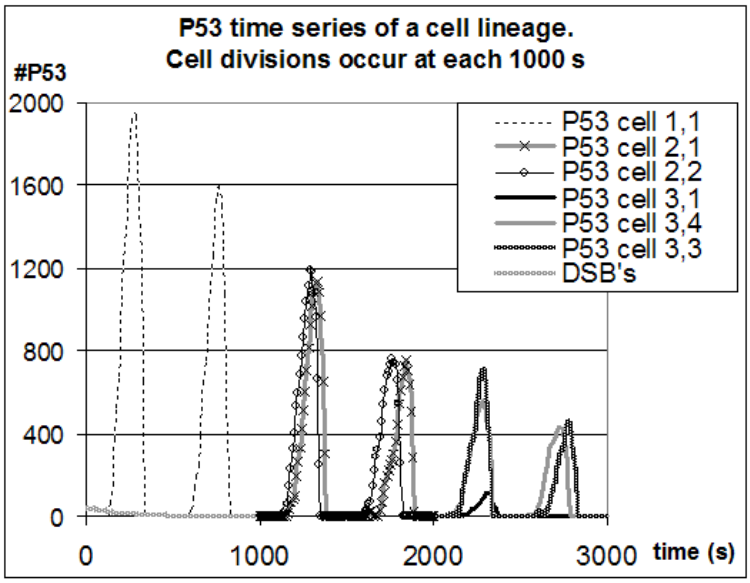

Figure 4: Time series of P53 in each cell of the cell line (except cell (3,2) since its time series is very similar to cell (3,1)). At each 1000 s, cells divide and two daughter cells are created from each existing cell. Notice that cells (3,3) and (3,4) have similar dynamics (almost synchronized both in phase and amplitude) since they are daughters from the same mother cell (cell (2,2)), but are almost uncorrelated to cell (3,1) generated from a different mother cell (cell (2,1)). Sampling period is 10 s.

From Fig. 4, one observes that the oscillations in mother cells continue in the daughter cells as the experiments report [2]. Additionally, and also matching experimental measurements, as the cells get more farther apart in the cell line "tree", their dynamics differ more. They become more unrelated both in phase and in amplitude. In the extreme case, in some lines the oscillations might have ceased while persisting in other lines. As seen in Fig. 4, another experimentally observed phenomenon, also present here, is that the oscillations can end abruptly at the single cell level [3].

Also observable from Fig. 4 is something which is observed in real cells when they divide, with respect to the concentration of P53 [2]. As the two daughter cells of the same mother cell evolve in time, although their oscillations are perfectly correlated in the beginning, they lose correlation in both frequency and amplitude of oscillations (this is more easily observed in the second generation).

This simulation took approximately 14 s to be completed (approximately 2 s per cell).

3 The repressilator

Originally, a repressilator consists of a 3-element negative feedback transcriptional loop [24]. The protein from the first gene inhibits the second gene, whose protein product in turn inhibits the third gene, and finally the protein from the third gene inhibits the first gene, completing the cycle.

A mean-field model showed that this configuration can produce oscillating levels of each repressor protein [24]. In the same work, it was reported that a repressilator, constructed in E. coli, showed such predicted oscillations.

Here, we simulate a 3 gene repressilator and qualitatively study this genetic oscillator. The model is described in detail in [4]. For simplicity, we consider only the symmetric case where all the three genes are identical.

If an extra gene is added to the loop, becoming a 4 gene repressilator, the system behavior changes dramatically. Namely, it becomes a bistable system (in a loose sense, since this is a stochastic system) [25]. This gene network is here used to exemplify how one can use *CellLine* to simulate a model of cells of a lineage that can dynamically evolve in a such a way that some branches of the lineage will reach one "stable" state, and the other branches, the opposite "stable" state.

3.1 3 gene repressilator subject to gene duplication at runtime.

In this example, we set up a gene network of 3 genes, where each gene represses the next gene, forming a repressilator. The system can be described by the following set of chemical reactions (implemented in the file "Model12_3_ring.g"):

$$\mathrm{RNA}p + \mathrm{Pro_i} \xrightarrow{0.01} \mathrm{Pro_i}(1) + \mathrm{RNA}p(20) + \mathrm{p_i}(100) \tag{17}$$

$$\mathrm{Pro_j} + \mathrm{p_i} \underset{0.08}{\overset{1}{\rightleftarrows}} \mathrm{Pro_j p_i} \tag{18}$$

$$p_i \xrightarrow{0.01} \emptyset \tag{19}$$

$$\mathrm{Pro_i p_j} \xrightarrow{0.01} \mathrm{Pro_i} \tag{20}$$

In these reactions (Eqs. 17 to 20), $N = 3$, $i = 1, ..., N$ and $j = i + 1$, except for $i = N$, where $j = 1$. In reaction 17, a time delay τ is associated to each product X of the reaction representing gene expression, using the notation: $X(\tau)$. Reaction 18 represents two independent reactions: binding and unbinding of the repressor to the promoter. The rate constants of these two reactions, represented in the numbers associated to the arrows, are not equal. The unbinding reaction allows the repressor to disassociate from the promoter. The repressor can also decay while on the promoter via reaction 20. This reaction is needed to allow the protein to decay when bound to the promoter at the same rate as if not bound. If this reaction was absent, binding to the promoter would act as a "protection" against decay. Proteins also decay via reaction 19. The system is initialized with the 3 promoters, and 300 RNAp's.

In this example, we model a case where each cell has, during its life time, a probability of duplicating the gene responsible for expressing protein p_1. The quantity of this promoter in a cell, propagates down the cell lineage. The increase in the number of promoters Pro_1 by gene duplication is expected to boost the number of transcription reactions [9] that occur per time unit.

The gene copy event can be modelled, e.g., as a source reaction with a small rate constant. Another possible method to create more promoters during run-time would be to use the queue list (see

manual for details). To do this we introduce a new rate constant in the lua block of the reactions file: "chance_duplicate = 1;". We then add the reaction in the initial cell's set of reactions (see Model12_3_ring.g file):

$$\xrightarrow{chance\_duplicate/10000} Pro_1 \tag{21}$$

CellLine when simulating this cell's dynamics according to the delayed SSA, will select this reaction to occur, on average, every 10 000 s. Since a cell's lifetime is set to 5000 s in this model, during this time the cell has a 50% chance of duplicating its promoter once. Any new promoters created during a cell lifetime are passed on to its daughter cells. The dynamics of the cell lineage were then simulated, over 3 generations, by executing the following command: "CellLineGen Model12_3_ring.g 3 5000 10".

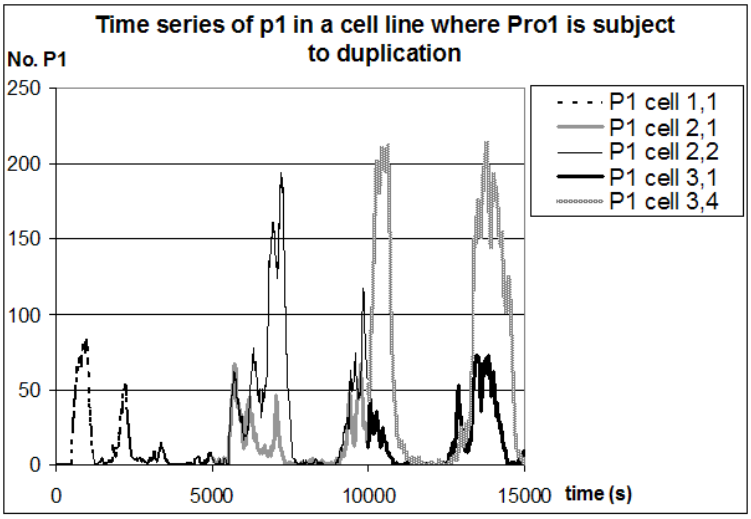

Figure 5: Time series of the proteins in cells of the lineage, each cell containing a 3 gene repressilator and (Pro_1) is subject to duplication. At each 5 000 s, cells divide and from each cell, two daughter cells are generated. The effects of this event, which in this simulation only occurred in cell (2,2), is propagated to its daughter cells, (3,3) (not shown) and (3,4), explaining the discrepancies in the time series of p_1 between cells (3,1) and (3,4).

In Fig. 5, the time series of protein p_1 is shown for some cells of the lineage.

As said, there is a reaction by which Pro_1 can be duplicated. In this particular run, such an event occurred only once, in cell (2,2) (one of the daughter cells of the initial mother cell). For that reason, the level of p_1 is higher in this cell and in its daughter cells (cells (3,3) and (3,4)), than in any other cells of the lineage.

The promoter duplication caused an increase in the time duration and amplitudes of the oscillations of p_1 in the cells with 2 promoters. Also, the proteins of the gene repressed by p_1 consequently have lower amplitudes (not shown).

In Fig. 6 we show such effect of gene duplication on the other proteins levels. Shown, is the time series of all proteins, of cell (2,2), where the gene duplication occurred. At first, when p_1 starts increasing to higher levels than previously, p_2 drops significantly. This decrease will allow p_3 to reach higher levels, which will in turn limit p_1's levels significantly.

To demonstrate *CellLine*'s ability to calculate the Fourier spectrum of a time series, we plot in Fig. 7 the Fourier spectrum of the time series of the mother cell of the system whose time series is shown in Fig. 5. This is done by adding the instruction "fourier_file FourierSpectrum.xls" in the reactions

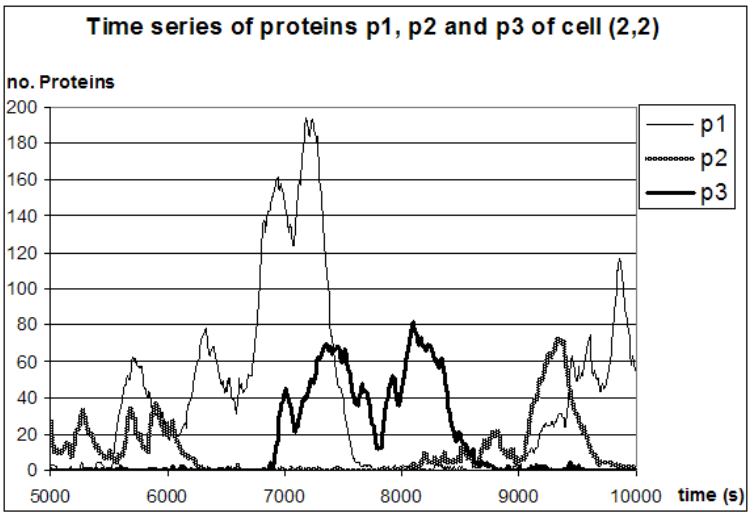

Figure 6: Time series of the three proteins (p_1, p_2 and p_3) of cell (2,2) of the lineage. At each 5 000 s, cells divide and from each cell, two daughter cells are generated. The effects of the duplication of Pro_1 is visible, resulting in higher levels of protein p_1, and lower levels of protein p_2, than the others.

file (see reactions file "Model12_3_ring.g", as an example). The Fourier spectrum can only be obtained for the first cell of the lineage, or using NCellsGen, for any single cell run.

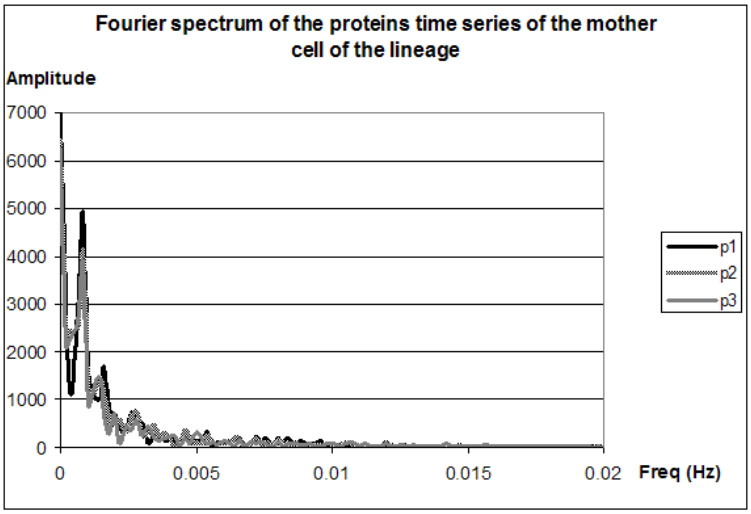

Figure 7: Fourier spectrum of the time series of a single cell containing a 3 gene repressilator where one of the genes is subject to duplication.

From Fig. 7, it is visible that the three proteins oscillate with the same period. The main frequency of oscillation is, approximately, 0.001 Hz.

3.2 4 gene repressilator

In this example, we set up a gene network of 4 genes, where each gene represses another gene, forming a ring of repression (here referred to as a 4 gene repressilator). The 4 gene repressilator can be described by the following reactions:

$$\mathrm{RNA}p + \mathrm{Pro_i} \xrightarrow{0.01} \mathrm{Pro_i}(1) + \mathrm{RNA}p(20) + \mathrm{p_i}(100) \tag{22}$$

$$\mathrm{Pro_j} + \mathrm{p_i} \underset{0.08}{\overset{1}{\rightleftarrows}} \mathrm{Pro_j p_i} \tag{23}$$

$$p_i \xrightarrow{0.01} \emptyset \tag{24}$$

$$\mathrm{Pro_i p_j} \xrightarrow{0.01} \mathrm{Pro_i} \tag{25}$$

In these reactions, $N = 4$, $i = 1, ..., N$ and $j = i + 1$, except for $i = N$, where $j = 1$. The system is initialized with the 4 promoters, and 400 RNAp.

What is interesting about this system is that there are two possible distinct "stable" states the system can reach. Also, once reaching such states, there is still a very small probability of "flipping" to the other state due to stochastic fluctuations in the proteins concentrations.

The rate constants of the reactions and the time delays were set so that the system can reach a "stable state", in the sense that either genes 1 and 3 are "on" (high number of proteins expressed by these genes present in the system) and genes 2 and 4 are almost completely repressed, or the the opposite (odd genes "off" and even genes "on"). Also, during a cell lifetime, no toggling between the two "stable" states is observed.

The cell is initialized without any proteins, meaning that it is not in either of the two possible "stable states". Thus, there is always a transient time until the system reaches one of these two states. Because cell lifetimes were set to be smaller than this transient, this choice usually occurs in the second generation of the lineage. Since it is equally likely to choose either of the two states, sometimes two daughter cells of the same progenitor cell will settle in opposite stable states.

Once the choice is made, its stability is maintained and propagated to the next generation. Thus, this case intends to be a simplistic model of a cell differentiation line, i.e., starting with an undifferentiated progenitor cell, it generates two cells with distinct gene expression patterns (which are therefore, different cell types).

Since in all cases, by knowing one of the gene's states we also know the states of the other genes, for easier visualization in Fig. 8, we only plot the time series of p_1. The model was simulated by executing the following command: "CellLineGen Model12_4_ring.g 3 10000 10".

The time series of protein p_1 of some of the cells of the lineage is plotted in Fig. 8. It shows that different lines of the lineage are able to choose different "stable states". If the mother cell is initiated in a state between these two stable ones and will equally likely chose either of them, then on average 50% of the cells will make one choice, and the other 50% the other choice.

Notice also that the choice for one of the two stable states propagates through the lineage. In this particular experiment, half the cells made one choice, while the other half made the opposite choice. Once that choice is made, the future generations will remain faithful to that decision. This is visible in the figure in the separation between the levels of p_1 in cells (3,1) and (3,4), which are daughters of different cells from the previous generation, respectively, cells (2,1) and (2,2) (the ones that made opposite choices, regarding what stable state to choose).

3.3 4 gene repressilator where a given initial state is imposed on each cell of the lineage.

As an example of how one can impose on all cells of a given lineage, a desired initial system state, we take the previous set of reactions, and add the following reactions (i = 1,...,4):

$$\mathrm{p_i} + X \xrightarrow{1000} \emptyset \tag{26}$$

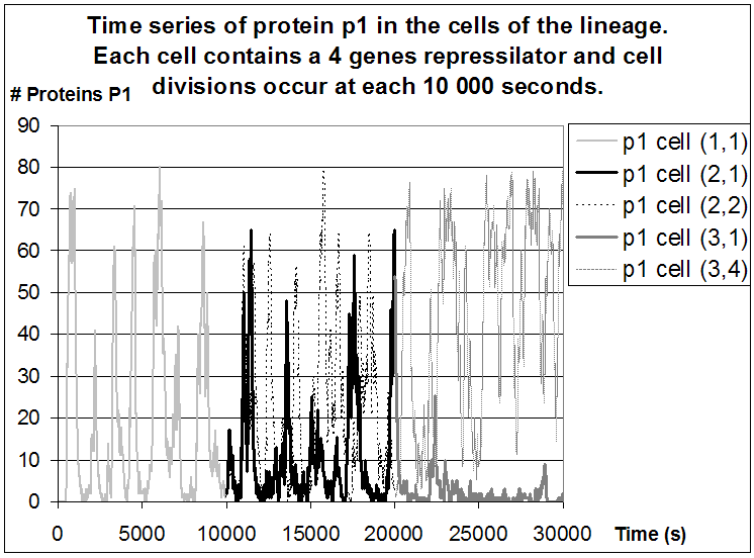

Figure 8: Time series of protein p_1 in some cells of the lineage. At each 10 000 s, cells divide and from each cell, two daughter cells are generated. Around t = 18 000 s cell (2,1) stabilizes in a state where p_1 level is very low, while independently at around the same time, cell (2,2) makes the opposite choice, i.e., stabilizes in a state where p_1 level is high. These decisions (stochastically driven) are propagated to their respective daughter cells, and therefore, the level of p_1 in cell (3,1) is always low and in cell (3,4) is always high. Sampling period is 10 s.

This substance "X" is introduced in the system (using the queue list) at the same moment the cell is set to divide, that is, at t = 10 000 s. It's function is to delete (via reaction 26) all existing proteins in the cells. By doing so, it also "self destructs", and after some time (once X is no longer present in the system), the proteins levels can again increase (see Fig 9). Since the mother cell was initialized with no proteins, this method allows imposing exactly the same initial state to its daughter cells, rather than inheriting the mother cell state (number of proteins, etc, as in the example of the P53-Mdm2 network) at the moment of division.

This example shows how to "reset" cells state at the beginning of their lifetimes. Using similar intervention techniques to those used in this example, one could have a model of symmetric cell division. Given the quantities of all elements in the mother cell, one can impose that these elements are, once passed to the daughter cells, roughly divided in half.

This could be done, for example, by using a substance X which removes $\sim 50\%$ of the proteins from the daughter cells, in the beginning of their life time. To remove that quantity of, e.g., a protein A, X is introduced in the system during the cells first second of life and the following reaction is defined:

$^*X + A - -[math.log(2)] - - >$;

Given this reaction, and X lifetime (1 s), on average, independently of how many A's exist in the cell, half of the A molecules will be destroyed during that first second of lifetime.

4 Modelling stochastic differentiation of cells determining the color vision mosaic in *Drosophila melanogaster*

Color vision in Drosophila is achieved by a set of R7 and R8 photoreceptor cells present in every ommatidium. The fly retina contains two types of ommatidia, "pale" and "yellow", defined by different

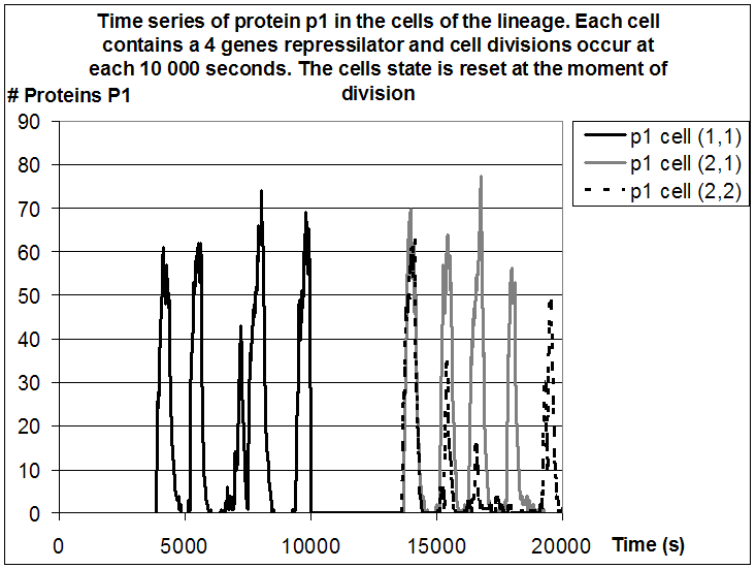

Figure 9: Time series of protein p_1 in a mother cell and its two daughters. When t = 10000 s, the mother cell divides into two daughter cells. The cells' state is reset by imposing the deletion of all existing proteins at the beginning of each cell's lifetime. Thus, the two daughter cells start at the same state that the mother cell did (with no proteins). Proteins start appearing in the cells approximately after 4000 s. Sampling period is 10 s.

rhodopsin pairs expressed in R7 and R8 cells. These subtypes are randomly spatially distributed in the retina. The choice between pale versus yellow ommatidia is made in R7 cells, which then impose their fate onto R8 cells.

It was reported that the Drosophila dioxin receptor Spineless (SS) is both necessary and sufficient for the formation of the ommatidial mosaic [26]. Namely, the creation of the retinal mosaic is driven by a burst of expression of a particular gene (spineless) during a specific stage of development. This example shows that in some cases, cells' fate is determined by a single stochastic variable [27, 26].

The stochastic expression SS acts as a binary switch in terms of determining the cell's fate. First, R7 cells are stochastically divided into two subtypes depending on the concentration of SS. SS-positive R7 cells express Rh4, whereas the remaining R7 choose the opposite fate and express Rh3 by default. Second, only those R7 cells that did not express SS in sufficient amounts retain the ability to induce the R8 cell fate, where gene Rh5 is expressed, whereas if SS is sufficiently expressed, R8 cells wont be influenced by R7 cells and express gene Rh6 by default.

Here we implement the model proposed in [27] that accounts for all the features described in [26].

The stochasticity of the choice of the pathway of differentiation of each cell in our model is in the random assignment of the quantity of SS at the beginning of the simulation.

This model is an example of a gene regulatory system capable of probabilistic regulation that leads to a differentiated state, out of two possible fates, which remains stable from there on.

Following [27] we simulate the following set of reactions:

$$\text{RNA}p + \text{Pro\_Rh3} \xrightarrow{0.1} \text{Pro\_Rh3}(1) + \text{RNA}p(20) + \text{p\_Rh3}(100) \tag{27}$$

$$\text{RNA}p + \text{Pro\_Rh4} \xrightarrow{0.00001} \text{Pro\_Rh4}(1) + \text{RNA}p(20) + \text{p\_Rh4}(100) \tag{28}$$

$$\text{RNA}p + \text{Pro\_Rh4\_SS} \xrightarrow{0.05} \text{Pro\_Rh4\_SS}(1) + \text{RNA}p(20) + \text{p\_Rh4}(100) \tag{29}$$

$$^{*}p\_Rh3 +^{*} p\_Rh3 + \text{RNA}p + \text{Pro\_Rh5} \xrightarrow{0.001} \text{Pro\_Rh5}(1) + \text{RNA}p(20) + \text{p\_Rh5}(100) \tag{30}$$

$$\text{RNA}p + \text{Pro\_Rh6} \xrightarrow{0.1} \text{Pro\_R}\text{h6}(1) + \text{RNA}p(20) + \text{p\_Rh6}(100) \tag{31}$$

$$SS + Pro\_Rh4 \underset{0.1}{\overset{1}{\rightleftarrows}} Pro\_Rh4\_SS \tag{32}$$

$$p\_Rh3 + Pro\_Rh5 \underset{0.1}{\overset{1}{\rightleftarrows}} Pro\_Rh5\_p\_Rh3 \tag{33}$$

$$Pro\_Rh3 + p\_Rh4 \underset{0.1}{\overset{1}{\rightleftarrows}} Pro\_Rh3\_p\_Rh4 \tag{34}$$

$$Pro\_Rh6 + p\_Rh5 \underset{0.1}{\overset{1}{\rightleftarrows}} Pro\_Rh6\_p\_Rh5 \tag{35}$$

$$p\_Rh3 \xrightarrow{0.01} \emptyset \tag{36}$$

$$p\_Rh4 \xrightarrow{0.01} \emptyset \tag{37}$$

$$p\_Rh5 \xrightarrow{0.01} \emptyset \tag{38}$$

$$p\_Rh6 \xrightarrow{0.01} \emptyset \tag{39}$$

$$Pro\_Rh3\_p\_Rh4 \xrightarrow{0.01} Pro\_Rh3 \tag{40}$$

$$Pro\_Rh6\_p\_Rh5 \xrightarrow{0.01} Pro\_Rh6 \tag{41}$$

$$Pro\_Rh5\_p\_Rh3 \xrightarrow{0.01} Pro\_Rh5 \tag{42}$$

Notice that reaction 29 has two "virtual" substrates ($^{*}p\_Rh3$). These are virtual in the sense that they are not consumed in the reaction although they do contribute to the calculation of that reaction's propensity (according to the standard SSA). For details refer to the Manual.

In this particular case, the use of these two virtual reactants causes the propensity of that reaction to be proportional to the square of the concentration of p_Rh3. As this substance quantity increases, the more likely if for this gene (Pro_Rh5) to express.

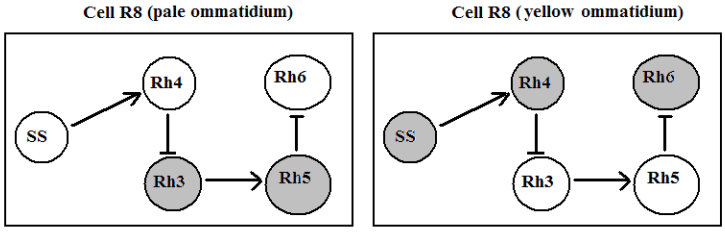

Figure 10: Gene regulatory network model [27] that controls the differentiation pathway of cell types Rh7 and Rh8, which will give rise to either cells Rh3 and Rh5 or Rh4 and Rh6. Genes colored grey are those active for each of the two cell types. The gene responsible for expressing protein SS, controls which of the differentiation pathways the cells follow. When grey, its expression is above the necessary threshold for the cell to commit to differentiate to cell type Rh4.

Additionally, in the reactions file, a random number between 1 and 100 is generated from a uniform distribution. If this number is larger than 30 then SS is given an initial quantity of 1. Otherwise, it

is 0. The model is simulated by the following command line: "NCellsGen Model_FlyEye.g 100 10000 10".

In Fig. 11, we present the time series from a case where SS = 1 at the beginning of the simulation, while in Fig. 12, we present the time series from a case where SS = 0 at the beginning of the simulation.

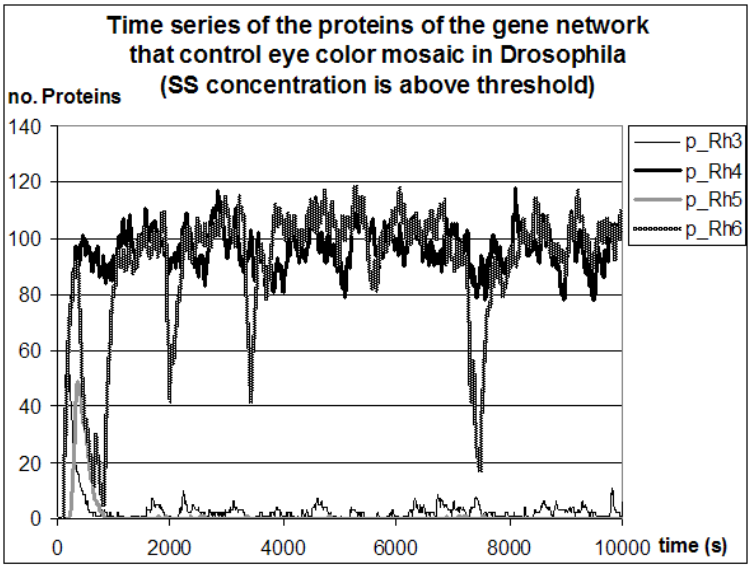

Figure 11: Proteins time series of the gene regulatory network that controls the differentiation pathway. In this case, due to the existence of SS in the cell, the proteins at high levels are p_Rh4 and p_Rh6 (resulting that this specific cell will be a "yellow" ommatidia).

Due to how the model was set up, namely, that the initial quantity of SS (present or absent) is determined randomly, from a uniform distribution, when simulating many cells, one obtains the measured ratios of "yellow" and "pale" cells (70% yellow and 30% pale). Since all cells differentiation fate is determined solely by their internal gene and proteins network dynamics, the spatial distribution of these two cell types is also random.

As said, this model aimed to mimic experimental studies [26] that suggest that the retinal mosaic is defined by the stochastic expression of a transcription factor (SS).

However, we point out that *CellLine* can also easily model differentiation lineages where differentiation is triggered by some external signal (e.g. addition of some chemical substance). A simple modification of the present model, such as assuming that SS is a substance external to the cell (e.g., added at a given moment in a quantity following some distribution), rather than produced by the cell, would be a good example of a model where cells' differentiation pathways are externally controlled.

5 Discussion

CellLine captures the stochastic nature of the real processes occurring in the cells. Also, it allows modelling complex multi-step reactions, such as transcription and translation, as single step multiple time delayed reactions [5], simplifying significantly computational processes.

NCellsGen module can simulate, from a single file of reactions, as many cells as desired. It allows the modelling of experimental measurements at the population level, simulating each individual cell's

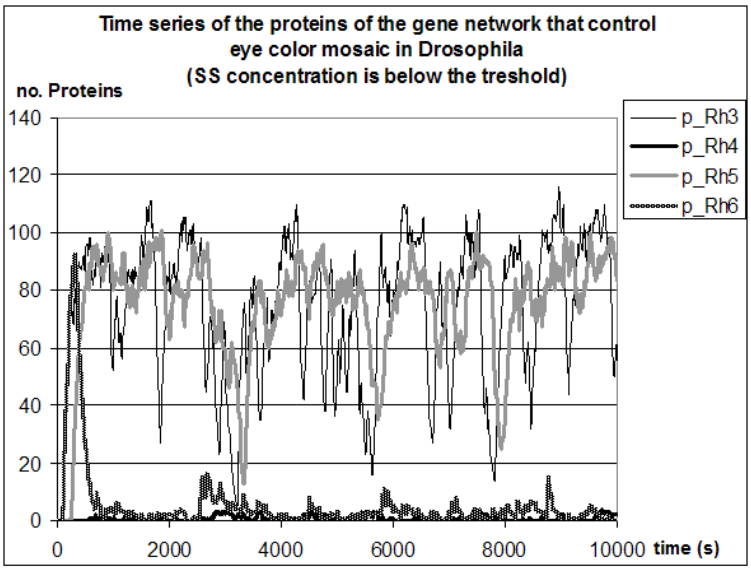

Figure 12: Proteins time series of the gene regulatory network that controls the differentiation pathway. In this case, due to the lack of SS in the cell, the proteins at high levels are p_Rh3 and p_Rh5 (resulting that this specific cell will be a "pale" ommatidia).

dynamics independently and each having a time series as output. This can be used to, for example, observe in detail the average behavior of a given cell obtained in a lineage growth.

Experiments are usually performed on many cells. In general, each cell of the population can be made unique by *CellLine*. For example, cells can have distinct rate constants, time delays or initial concentrations of any given substance. Even when identical in every feature, because *CellLine*ś dynamics is based on the delayed SSA and each cell is given a random seed, the stochastic nature of the dynamics results in identical cells having distinct behaviors.

The main module of *CellLine* (CellLineGen) allows the simulation of cell lineages. Starting from a single cell, the user defines its lifetime and how many generations to simulate. Each cell of the lineage is provided a unique random seed. At the end of its lifetime, each cell divides into two daughter cells. Daughter cells inherit any changes that occur during its mother cell's lifetime. For example, here we showed how P53 gene duplication events are propagated through the branches of the lineage. Importantly, CellLineGen keeps track of the lineage's total lifetime. Therefore one can, in the initial reactions file, set up perturbations to occur at a given generation of the cell lineage. For example, one can set up a substance X, to be introduced at time 3*T (where T is each cell's lifetime). This event can also have a random nature, i.e., a probability of being introduced at that given moment.

These features make *CellLine* capable of simulating a wide variety of virtual experiments. To our knowledge, no previous simulator currently exists that can automatically simulate cells' dynamics in the context of a cell lineage.

References

[1] Nicholas A.M. Monk, (2003) Oscillatory Expression of Hes1, p53, and NFκB Driven by Transcriptional Time Delays, *Current Biology*, **13**, 1409-1413.

[2] Naama Geva-Zatorsky, Nitzan Rosenfeld, Shalev Itzkovitz, Ron Milo, Alex Sigal, Erez Dekel, Talia Yarnitzky, Yuvalal Liron, Paz Polak, Galit Lahav, Uri Alon, (2006) Oscillations and variability in

the p53 system, *Molecular Systems Biology*, **2**, doi:10.1038/msb4100068.

[3] Lan Ma, John Wagner, John J. Rice, Wenwei Hu, Arnold J. Levine, Gustavo A. Stolovitzky, (2005) A plausible model for the digital response of p53 to DNA damage, *PNAS*, **102(40)**, 14266-14271.

[4] Rui Zhu, Andre S. Ribeiro, Dennis Salahub, Stuart A. Kauffman, (2005) Studying genetic regulatory networks at the molecular level: Delayed reaction stochastic models, *Journal of Theoretical Biology*, **246 (4)**, 725-745.

[5] M. R. Roussel and R. Zhu, (2006) Validation of an algorithm for delay stochastic simulation of transcription and translation in prokaryotic gene expression., *Phys. Biol.*, **3**, 274-284.

[6] G. I. Mihalas, Z. Simon, G. Balea, E. Popa, (2000) Possible Oscillatory Behavior in the P53-MDM2 Interaction Computer Simulation., *Journal of Biological Systems*, **8(1)**, 21-29.

[7] Ciliberto A., Novak B., Tyson J., (2005) Steady states and oscillations in the p53/Mdm2 network., *Cell Cycle*, **4**, 488-493.

[8] Tiana, G., Jensen, MH, Sneppen, K., (2002) Time delay as a key to apoptosis induction in the p53 network., *Eur Phys Jour. B*, **29**, 135-140.

[9] Gillespie, D.T., (1977) Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.*, **81(25)**, 2340-2361.

[10] Gillespie, D.T., (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, 1976, *J. Comput. Phys.*, **22**, 403-434.

[11] Andre S. Ribeiro and Jason Lloyd-Price, (2007) SGN Sim, a Stochastic Genetic Networks Simulator, *Bioinformatics*, **doi: 10.1093/bioinformatics/btm004**.

[12] Harris SL, Levine AJ, (2005) The p53 pathway: positive and negative feedback loops. *Oncogene* **24**, 2899-2908

[13] Banin S, Moyal L, Shieh S, Taya Y, Anderson CW, Chessa L, Smorodinsky NI, Prives C, Reiss Y, Shiloh Y, Ziv Y (1998) Enhanced phosphorylation of p53 by ATM in response to DNA damage. *Science* **281**, 1674-1677

[14] Andre S. Ribeiro, Rui Zhu and Stuart A. Kauffman, (2006) A General Modeling Strategy for Gene Regulatory Networks with Stochastic Dynamics (extended version). *Journal of Computational Biology* **13(9)**, 1630-1639

[15] Rief, N. and Lobrich, M., (2002) Efficient Rejoining of Radiation-induced DNA Double-strand Breaks in Centromeric DNA of Human Cells. *J. Biol. Chem.* **277**, 20572-82

[16] Vogelstein B, Lane D, Levine AJ (2000). Surfing the p53 network *Nature*, **408**, 307-10.

[17] Bennett, W., Hussain, S., Vahakangas, K., Khan, M., Shields, P., Harris, C., (1999) Molecular epidemiology of human cancer risk: gene-environment interactions and p53 mutation spectrum in human lung cancer. *J. Pathol.* **187 (1)**, 8-18.

[18] Bar-Or, R. L., Maya, R., Segel, L. A., Alon, U., Levine, A. J., Oren, M. (2000) Generation of oscillations by the p53-Mdm2 feedback loop: A theoretical and experimental study. *PNAS* **97 (21)**, 11250-55.

[19] Haupt, Y., Maya, R., Kazaz, A., Oren, M. (2005) Mdm2 promotes the rapid degradation of p53. *Nature* **387**, 296-9.

[20] Oliner JD, Pietenpol JA, Thiagalingam S, Gyuris J, Kinzler KW, Vogelstein B., (1993) Oncoprotein MDM2 conceals the activation domain of tumour suppressor p53., *Nature*, **362(6423)**, 857-60.

[21] Elowitz, M., Levine, A, Siggia, E., Swain, P., (2002) Stochastic gene expression in a single cell., *Science*, **297**, 1129-31.

[22] Arkin, A, Ross, J, McAdams, H.H., (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected E. coli cells *Genetics* **149(4)**, 1633-48.

[23] Kaern, M, Elston, T, Blake, W, Collins, J (2005) Stochasticity in gene expression: from theories to phenotypes *Nature Reviews Genetics* **6**, 451-64.

[24] Elowitz, M.B. and Leibler, S., (2000) A synthetic oscillatory network of transcriptional regulators, *Nature*, **403**, 335-338.

[25] Andre S. Ribeiro and Stuart A. Kauffman, (2007) Noisy Attractors and Ergodic Sets in Models of Genetic Regulatory Networks. *Journal of Theoretical Biology* **accepted**.

[26] Mathias F. Wernet, Esteban O. Mazzoni, Arzu Celik, Dianne M. Duncan, Ian Duncan, and Claude Desplan, (2006) Stochastic spineless expression creates the retinal mosaic for colour vision. *Nature*, **440**, 174-80.

[27] Michael S. Samoilov, Gavin Price, Adam P. Arkin, (2006) From Fluctuations to Phenotypes: The Physiology of Noise. *Science STKE* **366**.