

# Mathematical Modeling Of Systems Biology

Aaditya Prasad Gupta<sup>1</sup>

<sup>1</sup>Tilottama Campus, Yogikuti, Rupandehi, Nepal

\*aadiyaprasad.gupta@gmail.com

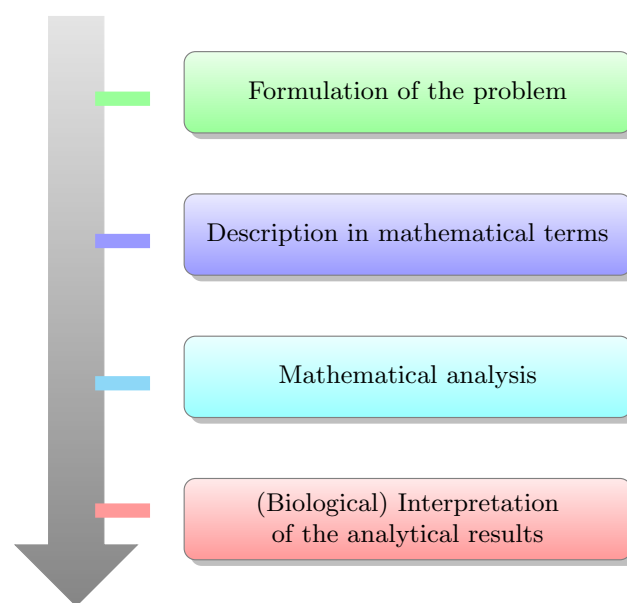
## ABSTRACT

A modeling is a mathematical tool, like a microscope, which allows consequences to logically follow from a set of assumptions by which a real world problem can be described by a mathematical formulation. It has become indispensable tools for integrating and interpreting heterogeneous biological data, validating hypothesis and identifying potential diagnostic markers. The modern molecular biology that is characterized by experiments that reveal the behaviours of entire molecular systems is called systems biology. A fundamental step in synthetic biology and systems biology is to derive appropriate mathematical model for the purposes of analysis and design. This manuscript has been engaged in the use of mathematical modeling in the Gene Regulatory System (GRN). Different mathematical models that are inspired in gene regulatory network such as Central dogma, Hill function, Gillespie algorithm, Oscillating gene network and Deterministic vs Stochastic modelings are discussed along with their codes that are programmed in Python using different modules. Here, we underlined that the model should describes the continuous nature of the biochemical processes and reflect the non-linearity. It is also found that the stochastic model is far better than deterministic model to calculate future event exactly with low chance of error.

**Keywords:** Mathematical models, Central dogma, Hill function, Gillespie algorithm, Oscillating gene network, Deterministic modeling, Stochastic modeling

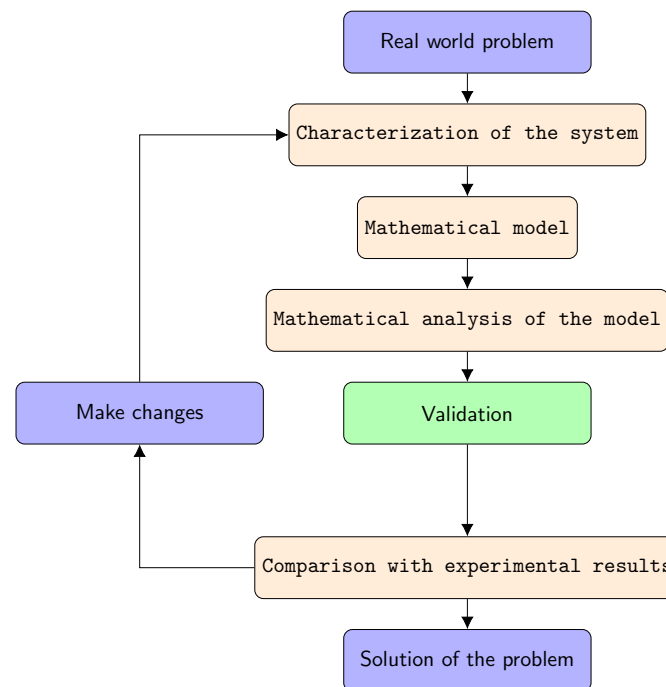
## 1 Introduction

Mathematical modeling is a process by which a real world problem is described by a mathematical formulation. However, neither all details of single processes will be described nor all aspects concerning the problem will be included. Models are designed to focus on certain aspects of the objects of study. A simple sketch of the mathematical modeling of systems biology is shown in figure 1 which is inspired from the reference<sup>1</sup>. The current modern molecular biology that is characterized by experiments that reveal the behaviours of entire molecular systems is called systems biology<sup>2</sup>.



**Figure 1.** Mathematical formulation for interpreting the analytical biological result

A main problem in systems biology is to find an appropriate mathematical formulation. Then, for further studies of the model, common mathematical tools can be used or new ones are developed. Further, a model should be as simple and detailed as necessary or vice versa. One mathematical formulation may be appropriate for several real-world problems, even from very different domains.



**Figure 2.** Mathematical model to solve the real world problem

A great challenge of modelling is to bring together the abstract, mathematical formulation and concrete experimental data. The modelling process can be roughly described as shown in figure 2. The figure 2 is inspired from the reference<sup>3</sup>.

Mathematical modeling has been applied to biological systems for decades, but with respect to gene expression, too few molecular components have been known to build useful, predictive models. There are many different modelling approaches and their number is still increasing. Some of the very famous models in molecular biology are discussed in this manuscript. In section 2 we discuss about the Central dogma. In section 3 Hill function is discussed while Oscillating gene network in section 4, Gillespie algorithm in section 5 and Deterministic vs Stochastic modelings in section 6. Finally, the aspects and scope of mathematical modeling to the above stated models are discussed in section 7. The codes used for the modelling are shown in the appendix along with a brief explanation and a much detailed version on github<sup>1</sup>.

## 2 Central Dogma

Central dogma of molecular biology is an explanation of the flow of genetics information within a biological system. It is often stated as "DNA makes RNA and RNA makes proteins"<sup>4</sup>. The transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein may be possible but transfer from protein to protein or from protein to nucleic acid is impossible. Information means here precise determination of sequence either of bases in the nucleic acid or of amino acid residue in protein.

The dogma is a framework for understanding the transfer of sequence information between information-carrying bio-polymers in the most common or general case in living organism. There are 3 major classes of such bio-polymers: DNA and RNA (both nucleic acids) and protein. There are  $3 \times 3 = 9$  conceivable direct transfer of information that can occur between these. The dogma classes these into 3 groups of 3<sup>5</sup>:

- 3 general transfers(believed to occur normally in most cells),

<sup>1</sup>[github.com/aaditya-pdgupta/mathematical-modeling-in-biology](https://github.com/aaditya-pdgupta/mathematical-modeling-in-biology)

- 3 special transfers(known to occur, but only under specific conditions in case of some viruses or in a laboratory),
- 3 unknown transfers(believed never to occur).

The general transfers describe the normal flow of biological information: DNA can be copied to DNA (DNA replication), DNA information can be copied into mRNA, (transcription), and proteins can be synthesized using the information in mRNA as a template (translation).

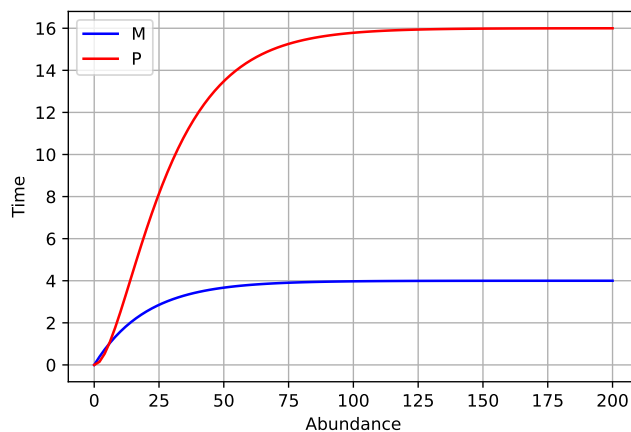
The coupled differential equations representing the model of Central dogma are shown in equation 1 & equation 2.

$$\frac{dm}{dt} = k_m - \gamma_m m \quad (1)$$

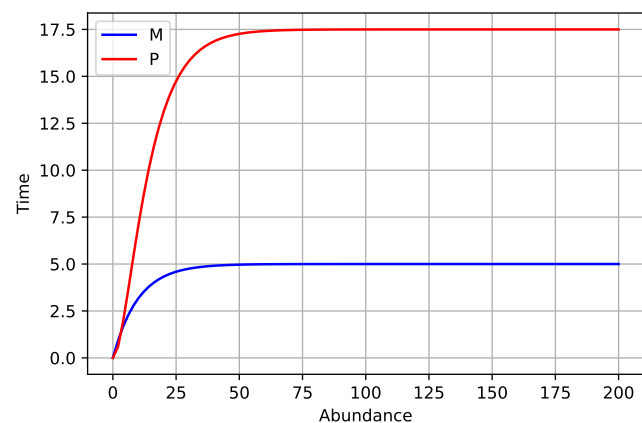
$$\frac{dp}{dt} = k_p m - \gamma_p p \quad (2)$$

where

- p is protein
- m is mRNA
- $k_m$  is the production rate for the mRNA
- $\gamma_m$  is the degradation rate for mRNA
- $\gamma_p$  is the degradation rate of protein
- $k_p$  is the production rate for the proteins



**Figure 3.** Abundance of mRNA over time with the initial conditions:  $k_m = 0.2$ ,  $\gamma_m = 0.05$ ,  $k_p = 0.4$ ,  $\gamma_p = 0.1$



**Figure 4.** Abundance of mRNA over time with the initial conditions:  $k_m = 0.5$ ,  $\gamma_m = 0.1$ ,  $k_p = 0.7$ ,  $\gamma_p = 0.2$

The figures 3 & 4 show the relation between abundance of mRNA and protein over time. Along the y-axis, we have the time axis and along x-axis we have abundance of both mRNA and protein. We observe that M(blue line) denotes mRNA in the above figure 3 & 4 while P(red line) denotes protein. In figure 3 both of them starts at zero and then they increases until they reach their steady state. The steady state for mRNA is 4 and for protein is 16 above. The abundance of protein is more than mRNA because it depends upon the mRNA production (  $k_p * m$  ) while mRNA production is independent ( $k_m$ ).

If we change the initial parameters such as  $k_m$ ,  $\gamma_m$ ,  $k_p$ ,  $\gamma_p$  greater than that are used in figure 3, we came to know that the steady state for both mRNA and protein increases by same amount and the nature of the curves remain almost same which is shown in figure 4. In figure 4 the steady state approaches faster than that in figure 3 due to larger degradation rate.

### 3 Hill Function

The so called Hill function were introduced by A.V Hill in 1910 to describe the binding of oxygen to hemoglobin. Subsequently, they have been widely used in biochemistry, psychology and mathematical modeling gene expression<sup>6</sup>. Different mathematical frameworks have been proposed to derive the mathematical model. In particular the use of sets of nonlinear ordinary differential

equation(ODE) has been proposed to model the dynamics of the concentrations of mRNAs and proteins. These models are usually characterized by the presence of highly nonlinear Hill function terms<sup>7</sup>. Hill functions follow from the equilibrium state of the reaction in which n ligands simultaneously bind a single receptor<sup>6</sup>.

The Central dogma of molecular biology states that DNA makes and RNA makes proteins. The process by which DNA is copied to RNA is called transcription and by which RNA is used to produce protein is translation. The Hill function is expressed as follow:

- Activation Hill function
- Repression Hill function

### 3.1 Activation Hill Function

In this function Gene first ( $G_1$ ) acts as an activator for Gene second ( $G_2$ ) and it increases the probability of transcription often by increasing probability of RNA polymerase binding.

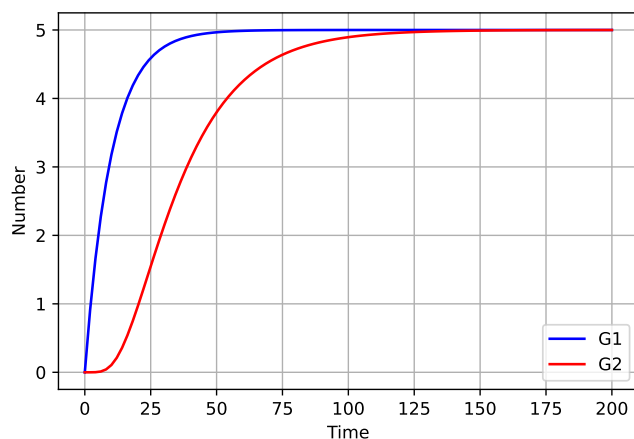
The coupled differential equations representing the model of activation Hill function are shown in equation 3 & equation 4.

$$\frac{dG_1}{dt} = k_1 - \gamma_1 G_1 \quad (3)$$

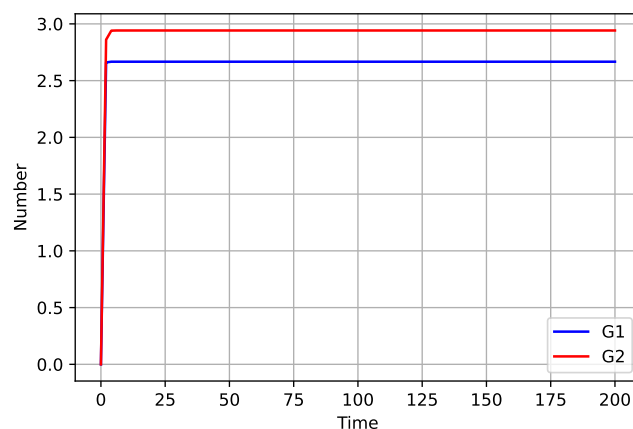
$$\frac{dG_2}{dt} = \left[ \frac{G_1^n}{c^n + G_1^n} \right] k_2 - \gamma_2 G_2 \quad (4)$$

where

- $k_1$  is production rate of  $G_1$
- $k_2$  is production rate of  $G_2$
- $c$  = constant
- $\gamma_1$  is degradation rate of  $G_1$
- $\gamma_2$  is degradation rate of  $G_2$
- $n$  = hill constant



**Figure 5.** Number of gene over time with the initial conditions:  $k_1 = 0.5$ ,  $\gamma_1 = 0.1$ ,  $k_2 = 0.5$ ,  $\gamma_2 = 0.05$ ,  $n=5$ ,  $c=5$



**Figure 6.** Number of gene over time with the initial conditions:  $k_1 = 8$ ,  $\gamma_1 = 3$ ,  $k_2 = 6$ ,  $\gamma_2 = 2$ ,  $n=4$ ,  $c=1$

The figures 5 & 6 show the relation between Gene first ( $G_1$ ) and Gene second ( $G_2$ ) over time. Along the x-axis, we have the time axis and along y-axis we have number of Gene first ( $G_1$ ) and Gene second  $G_2$ . We observe that  $G_1$  (blue line) denotes Gene first in the above figure 5 & 6 while  $G_2$  (red line) denotes Gene second. Here, ( $G_1$ ) quickly get activated and reaches to the steady point while  $G_2$  delays in figure 5. This is because for activating the  $G_2$ ,  $G_1$  should be produced.

If we change the initial parameters such as  $k_1$ ,  $\gamma_1$ ,  $k_2$ ,  $\gamma_2$  greater than that are used in figure 5 above we came to know that the steady state for both Gene first ( $G_1$ ) and Gene second ( $G_2$ ) changes and the nature of the curves also changes which is shown in figure 6. If  $G_1$  be the zero than the production rate of gene two will also be zero.

### 3.2 Repression Hill Function

In this function, Gene first ( $G_1$ ) acts as a repressor for Gene second ( $G_2$ ) and it decreases the probability of transcription often by decreasing probability of RNA polymerase binding. The damping of protein production by a repressive agent occurs linearly but fluctuations can show a maximum at intermediate repression<sup>8</sup>.

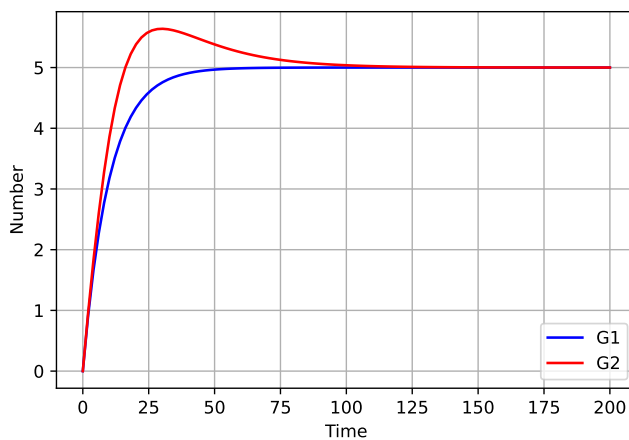
The coupled differential equations representing the model of repression Hill function are shown in equation 5 & equation 6.

$$\frac{dG_1}{dt} = k_1 - \gamma_1 G_1 \quad (5)$$

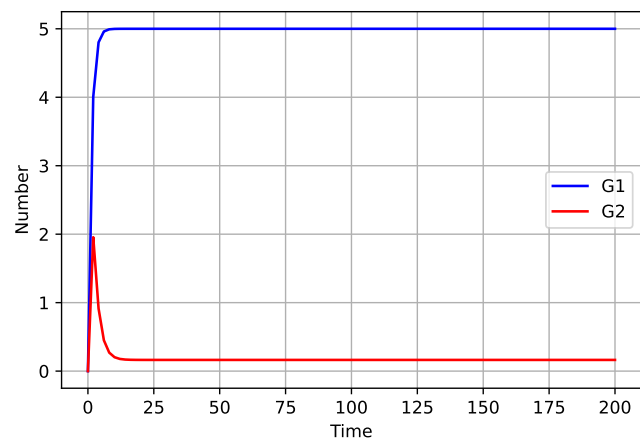
$$\frac{dG_2}{dt} = \left[ \frac{c^n}{c^n + G_1^n} \right] k_2 - \gamma_2 G_2 \quad (6)$$

where

- $k_1$  is production rate of  $G_1$
- $k_2$  is production rate of  $G_2$
- $c$  = constant
- $\gamma_1$  is degradation rate of  $G_1$
- $\gamma_2$  is degradation rate of  $G_2$
- $n$  = hill constant



**Figure 7.** Number of gene over time with the initial conditions:  $k_1 = 0.5$ ,  $\gamma_1 = 0.1$ ,  $k_2 = 0.5$ ,  $\gamma_2 = 0.05$ ,  $n=5$ ,  $c=5$



**Figure 8.** Number of gene over time with the initial conditions:  $k_1 = 4$ ,  $\gamma_1 = 0.8$ ,  $k_2 = 0.3$ ,  $\gamma_2 = 0.5$ ,  $n=7$ ,  $c=3$

The figures 7 & 8 show the relation between Gene first ( $G_1$ ) and Gene second ( $G_2$ ) over time. Along the x-axis, we have the time axis and along y-axis we have number of Gene first ( $G_1$ ) and Gene second ( $G_2$ ). We observe that  $G_1$  (blue line) denotes Gene first in the above figure 7 & 8 while  $G_2$  (red line) denotes Gene second. Here ( $G_2$ ) quickly get activated and reaches to the peak point but  $G_1$  protein start repressing it and suddenly it goes down in figure 7. This show that  $G_1$  act as repressor in this case.

If we change the initial parameters such as  $k_1$ ,  $\gamma_1$ ,  $k_2$ ,  $\gamma_2$  greater than that are used in figure 7, we came to know that the Gene first ( $G_1$ ) approaches near to zero. If  $G_1$  is very large, than the gene two will reach the peak point but  $k_2$  approaches very close to zero but never can be zero.

## 4 Oscillating Gene Network

Oscillating gene is a gene that is expressed in a rhythmic pattern or in periodic cycles<sup>9,10</sup>. Oscillating genes are usually circadian and can be identified by periodic changes in the state of an organisms. Oscillating gene model is a complex gene network. Same as above, the Central Dogma of molecular biology states that DNA makes and RNA makes proteins. The process by which DNA is copied to RNA is called transcription and by which RNA is used to produce protein is translation.

For this model, let us consider Gene first as ( $G_1$ ), Gene second as ( $G_2$ ) and Gene third ( $G_3$ ).  $G_1$  activates the  $G_2$  and facilitates the transcription of  $G_2$  as a result  $G_2$  get transcribed. It is positive interaction.  $G_2$  does same for the  $G_3$  and facilitates

the transcription of  $G_3$  as a result  $G_3$  get transcribed. But  $G_3$  comes back to repress the transcription of  $G_1$ .  $G_3$  is inhibiting the expression by blocking the transcription of  $G_1$ . It is known as negative feedback because it is cascading  $G_1$  being transcribed and then activating  $G_2$  which later transcribed and helps in activating  $G_3$ . But  $G_3$  is negatively feeding back. This cause oscillation during stimulation.

The coupled differential equations representing the model of Oscillating gene network are shown in equation 7 , equation 8 & equation 9.

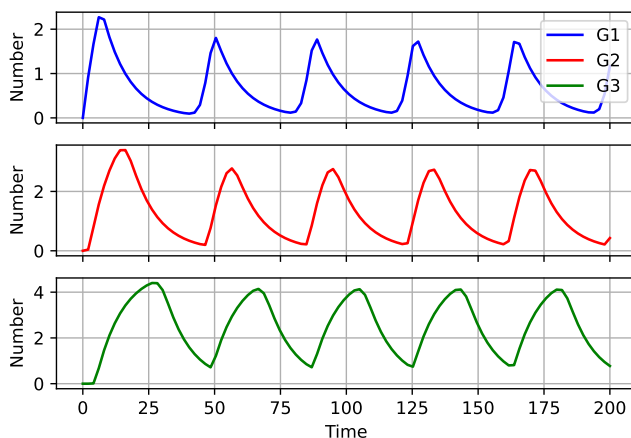
$$\frac{dG_1}{dt} = \left[ \frac{c^n}{c^n + G_3^n} \right] k_1 - \gamma_1 G_1 \quad (7)$$

$$\frac{dG_2}{dt} = \left[ \frac{G_1^n}{c^n + G_1^n} \right] k_2 - \gamma_2 G_2 \quad (8)$$

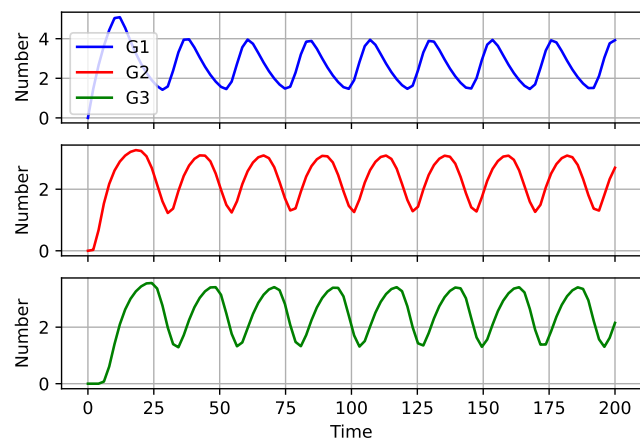
$$\frac{dG_3}{dt} = \left[ \frac{G_2^n}{c^n + G_2^n} \right] k_3 - \gamma_3 G_3 \quad (9)$$

where

- $k_1$  is production rate of  $G_1$
- $\gamma_1$  is degradation rate of  $G_1$
- $k_2$  is production rate of  $G_2$
- $\gamma_2$  is degradation rate of  $G_2$
- $k_3$  is production rate of  $G_3$
- $\gamma_3$  is degradation rate of  $G_3$
- $c$  = constant
- $n$  = hill constant



**Figure 9.** Number of gene over time with the initial conditions:  $k_1 = 0.5$ ,  $\gamma_1 = 0.1$ ,  $k_2 = 0.5$ ,  $\gamma_2 = 0.01$ ,  $k_3 = 0.5$ ,  $\gamma_3 = 0.1$ ,  $n=9$ ,  $c=1$



**Figure 10.** Number of gene over time with the initial conditions:  $k_1 = 0.8$ ,  $\gamma_1 = 0.1$ ,  $k_2 = 0.7$ ,  $\gamma_2 = 0.2$ ,  $k_3 = 0.8$ ,  $\gamma_3 = 0.2$ ,  $n=6$ ,  $c=2$

The figures 9 & 10 show the relation between Gene first ( $G_1$ ), Gene second ( $G_2$ ) and Gene third ( $G_3$ ) over time. Along the x-axis, we have the time axis and along y-axis we have number of Gene first ( $G_1$ ), Gene second  $G_2$  and Gene third ( $G_3$ ). We observe that  $G_1$ (blue line) denotes Gene first,  $G_2$ (red line) denotes Gene second and  $G_3$ (green line) denotes Gene third. In figure 9, we find oscillation with 3G network. Gene third  $G_3$  is repressing  $G_1$  which is negative feedback. Here  $G_1$  is being produced that leads to  $G_2$  production and  $G_2$  is produced which leads to the production of  $G_3$ . But  $G_3$  produced stops  $G_1$  from being produced. So that we get waves like this.

If we change the initial parameters such as  $k_1$ ,  $\gamma_1$ ,  $k_2$ ,  $\gamma_2$ ,  $k_3$  and  $\gamma_3$  greater than that are used in figure 9 came to know that the nature of oscillation of all gene remain same but peak points of the all gene varies from figure 9 for figure 10.

## 5 Gillespie Algorithm

The Gillespie algorithm(or occasionally the Doob-Gillespie algorithm) generates a statistically correct trajectory(possible solution) of a stochastic equation system for which the reaction rates are known. The algorithm was first presented by Doob in the mid 1940s<sup>11,12</sup>. It was implemented by Kendall in the 1950s<sup>13,14</sup>. However it wasn't until the mid 1970s<sup>15,16</sup> that Gillespie re-derived the method by studying physical systems that it became widely used<sup>17</sup>. It is also sometime called as SSA method.

For this let consider a gene as  $x$  and some mRNA is transcribed from  $x$  with production rate  $k$  and it will be degraded with rate  $\gamma_x$ . For this list, all the rate in array as  $r = [k, \gamma_x]$ . For this, let choose a time point of the reaction. If we are at time  $t$  which could be zero or can be any time between algorithm. Then next time is going to be  $t + \tau$ . Here we choose  $\tau$  from exponential random distribution with the parameter lambda which is the sum of all rates at time  $t$ . After choosing the time point it is important to know whether it will be production of mRNA or breakdown of mRNA. The rate for breakdown depends on the current level of  $x$ . If there will be  $x$  zero than  $\gamma_x$  will be zero and there will be only production. But suppose, if we have couple of  $x$  with some positive number. So to choose the event going to happen we take random draw between two of them and weight the probabilities accordingly. Here probability of production of mRNA that means probability of  $x$  going to  $x + 1$   $\{ P(x \rightarrow x + 1) = (\frac{k}{\sum(r)}) \}$  and probability of breaking down of mRNA is that means probability of  $x - 1$   $\{ P(x \rightarrow x - 1) = (\frac{\gamma_x}{\sum(r)}) \}$ .

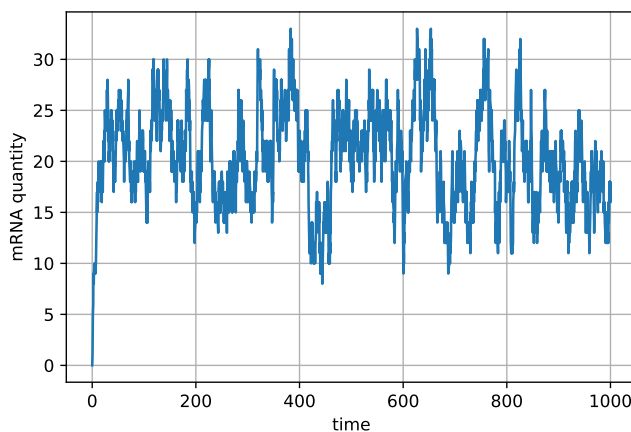
The differential equation representing the model of repression Gillespie algorithm is shown in equation 10.

$$\frac{dx}{dt} = k - \gamma_x \quad (10)$$

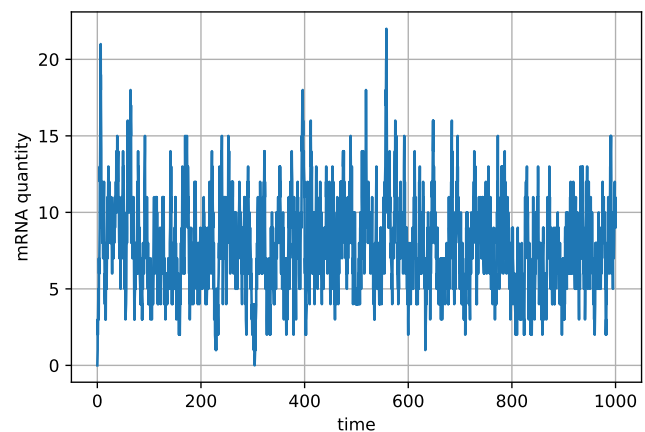
Events	Rates
$x \rightarrow x + 1$	$k$
$x \rightarrow x - 1$	$\gamma_x$

where

- $k$  is the production rate for mRNA
- $\gamma_x$  is the degradation rate for mRNA



**Figure 11.** Quantity of mRNA over time with the initial conditions:  $k=2, \gamma = 0.1$



**Figure 12.** Quantity of mRNA over time with the initial conditions:  $k=3, \gamma = 0.4$

The figures 11 & 12 show the relation between quantity of mRNA and protein over time. Along the y-axis, we have the time axis and along x-axis we have abundance of both mRNA and protein. We observe that  $M$ (blue line) denotes quantity of mRNA in the above figures 11 & 12. If we do the stimulation for hundred times or thousand times it will show average steady state near about 20 for above figure 11. Since it is stochastic model the steady state is not real because it is fluctuating near the steady state.

If we change the initial parameters of  $k$  and  $\gamma$  greater than above figure 11, we came to know that fluctuation is more and hard to find the steady state in figure 12.

## 6 Deterministic vs Stochastic Modelings

Deterministic modeling allows us to calculate a future event exactly without the involvement of randomness. Stochastic modeling allows us to calculate a future event exactly with the involvement of randomness. In deterministic there is each set of initial condition so there is only one trajectory. Through this model we get smooth curve. The difference with stochastic model is that stochastic model involves some random component. Due to amalgamation of noise from different source, micro-array expression profiles becomes inherently noisy leading to significant impact on the Gene Regulatory Networks (GRN) reconstruction process. Micro-array replicates (both biological and technical), generated to increase the reliability of data obtained under noisy conditions, have limited influence in enhancing the accuracy of reconstruction. Therefore instead of the conventional GRN modeling approaches which are deterministic, stochastic techniques are becoming increasingly necessary<sup>18</sup>.

The trajectory of deterministic modeling is completely defined by the initial parameters and conditions due to no involvement of randomness but in stochastic modeling with the same parameters and same initial conditions we get different trajectories running at different times. In computational biology one motive for using stochastic modeling is to calculate random variation in system which is not possible through deterministic modelings. Here, we will discuss about oscillating gene network same as above for stochastic modeling calculation.

The coupled differential equations representing the model of Stochastic modelings are shown in equation 11 , 12 & 13.

$$\frac{dG_1}{dt} = \left[ \frac{c^n}{c^n + G_3^n} \right] k_1 - \gamma_1 G_1 \quad (11)$$

$$\frac{dG_2}{dt} = \left[ \frac{G_1^n}{c^n + G_1^n} \right] k_2 - \gamma_2 G_2 \quad (12)$$

$$\frac{dG_3}{dt} = \left[ \frac{G_2^n}{c^n + G_2^n} \right] k_3 - \gamma_3 G_3 \quad (13)$$

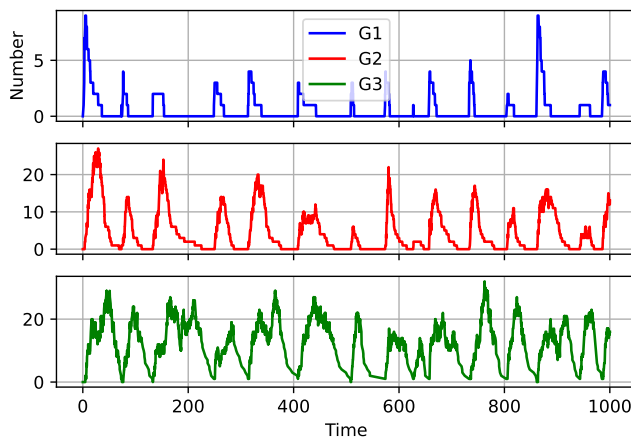
Events	Rates
$G_1 \rightarrow G_1 + 1$	$\left[ \frac{c^n}{c^n + G_3^n} \right] k_1$
$G_1 \rightarrow G_1 - 1$	$\gamma_1 G_1$
$G_2 \rightarrow G_2 + 1$	$\left[ \frac{G_1^n}{c^n + G_1^n} \right] k_2$
$G_2 \rightarrow G_2 - 1$	$\gamma_2 G_2$
$G_3 \rightarrow G_3 + 1$	$\left[ \frac{G_2^n}{c^n + G_2^n} \right] k_3$
$G_3 \rightarrow G_3 - 1$	$\gamma_3 G_3$

where

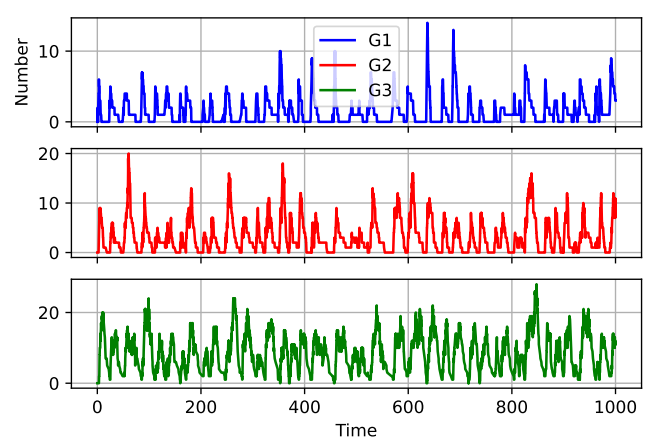
- $k_1$  is production rate of  $G_1$
- $\gamma_1$  is degradation rate of  $G_1$
- $k_2$  is production rate of  $G_2$
- $\gamma_2$  is degradation rate of  $G_2$
- $k_3$  is production rate of  $G_3$
- $\gamma_3$  is degradation rate of  $G_3$
- $c$  = constant
- $n$  = hill constant

The figures 13 & 14 show the relation between Gene first ( $G_1$ ), Gene second ( $G_2$ ) and Gene third ( $G_3$ ) over time. Along the x-axis, we have the time axis and along y-axis we have number of Gene first ( $G_1$ ), Gene second ( $G_2$ ) and Gene third ( $G_3$ ). We





**Figure 13.** Number of gene over time with the initial conditions:  $k_1 = 2$ ,  $\gamma_1 = 0.1$ ,  $k_2 = 2$ ,  $\gamma_2 = 0.1$ ,  $k_3 = 2$ ,  $\gamma_3 = 0.1$ ,  $n=9$ ,  $c=1$



**Figure 14.** Number of gene over time with the initial conditions:  $k_1 = 3$ ,  $\gamma_1 = 0.2$ ,  $k_2 = 2.5$ ,  $\gamma_2 = 0.2$ ,  $k_3 = 3$ ,  $\gamma_3 = 0.1$ ,  $n=10$ ,  $c=2$

observe that  $G_1$  (blue line) denotes Gene first,  $G_2$  (red line) denotes Gene second and  $G_3$  (green line) denotes Gene third in both figure 13 & 14. Each one have the oscillations but there is unevenly space because of the stochastic nature of this model. In compared to previous oscillator topic we did deterministic ODEs model which was smooth but this is stochastic so they are still oscillating but it is much more random.

If we change the initial parameters such as  $k_1$ ,  $\gamma_1$ ,  $k_2$ ,  $\gamma_2$ ,  $k_3$  and  $\gamma_3$  greater than the above figure 13, we came to know that the nature of oscillation of all Gene remain same but peak points of the all Gene varies from figure 13 and some what oscillates in rapid speed in less time in figure 14.

## 7 Discussion

As mentioned above, this manuscript has been engaged in the use of mathematical modelling in the Gene Regulatory System (GRN). Here, we underlined that the model should describes the continuous nature of the biochemical processes and reflect the non-linearities. We used the gene expression rates by ordinary differential equations (ODEs) where the expression rates were approximated by means of difference quotients. For the modeling, Python modules were used. Python has many excellent and well-maintained libraries that facilitate high-level scientific computing analyses. The modules such as numpy<sup>19</sup> which offers a numerical processing library that supports multi-dimensional arrays, scipy<sup>20</sup> which offers a scientific processing library providing advanced tools for data analysis, including regression, ODE solvers and integrator, linear algebra and statistical functions, matplotlib<sup>21</sup> which offers a plotting library with tools to display data in a variety of ways were used in Anaconda<sup>22</sup> software. The dynamical behavior of gene expressions which are represented by a system of ordinary differential equations were solved using "odeint" from scipy module. Further we also considered the different parameters to solve the different ODEs. Finally, the change of dependent variable with respect to time were plotted in different models under study.

In Central dogma, we observe that flow of genetic information in a biological system is steady after a certain time, and the abundance of both mRNA and protein depends on the production. Likewise in Hill function we observe that it is a parameters which increases or decreases the probability of RNA polymerise binding. In Oscillating gene network, we observe that it is a positive complex network also known as negative feed back. In Gillespie algorithm, we observe that it is an algorithm for predicting the time point of the reaction between the mRNA and proteins. In the Deterministic vs Stochastic modelings, we observe that deterministic model can calculate the future event with out randomness but stochastic can calculate future with randomness. We also discussed about the randomness of gene over a time which causes the significant impact on the Gene Regulatory Network (GRN). The Stochastic model is far better than Deterministic model to calculate future event exactly with low chance of error. We are able to predict the future event using different parameters. Thus, modeling is a mathematical tool, like a microscope, which allows consequences to logically follow from a set of assumptions. It may allow to understand importance of specific mechanisms/assumptions in biological processes. However, mathematical modeling can also be misleading if used inappropriately.

# References

1. Kremling, A. *Systems biology: mathematical modeling and model analysis* (CRC Press, 2013).
2. Ingalls, B. P. *Mathematical modeling in systems biology: an introduction* (MIT press, 2013).
3. Johannes Müller, C. K. *Mathematical models in biology* (Springer Link, 2015).
4. Crick, F. George gamov worked out the “central dogma,” explaining how dna functions to make protein. *Google Scholar* (1957).
5. Crick, F. Central dogma of molecular biology. *Nature* **227**, 561–563 (1970).
6. Santillán, M. On the use of the hill functions in mathematical models of gene regulatory networks. *Math. Model. Nat. Phenom.* **3**, 85–97 (2008).
7. Polynikis, A., Hogan, S. & di Bernardo, M. Comparing different ode modelling approaches for gene regulatory networks. *J. theoretical biology* **261**, 511–530 (2009).
8. Innocentini, G. C. & Hornos, J. E. Modeling stochastic gene expression under repression. *J. Math. Biol.* **55**, 413–431 (2007).
9. Moreno-Risueno, M. A. & Benfey, P. N. Time-based patterning in development: the role of oscillating gene expression. *Transcription* **2**, 124–129 (2011).
10. Tuttle, L. M., Salis, H., Tomshine, J. & Kaznessis, Y. N. Model-driven designs of an oscillating gene network. *Biophys. journal* **89**, 3873–3883 (2005).
11. Doob, J. L. Topics in the theory of markoff chains. *Transactions Am. Math. Soc.* **52**, 37–64 (1942).
12. Doob, J. L. Markoff chains–denumerable case. *Transactions Am. Math. Soc.* **58**, 455–473 (1945).
13. Kendall, D. G. An artificial realization of a simple" birth-and-death" process. *J. Royal Stat. Soc. Ser. B (Methodological)* **12**, 116–119 (1950).
14. Bartlett, M. S. Stochastic processes or the statistics of change. *J. Royal Stat. Soc. Ser. C (Applied Stat.* **2**, 44–64 (1953).
15. Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. computational physics* **22**, 403–434 (1976).
16. Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *The journal physical chemistry* **81**, 2340–2361 (1977).
17. Cai, X. Exact stochastic simulation of coupled chemical reactions with delays. *The J. chemical physics* **126**, 124108 (2007).
18. Chowdhury, A. R., Chetty, M. & Evans, R. Stochastic s-system modeling of gene regulatory network. *Cogn. Neurodynamics* **9**, 535–547 (2015).
19. Van Der Walt, S., Colbert, S. C. & Varoquaux, G. The numpy array: a structure for efficient numerical computation. *Comput. science & engineering* **13**, 22–30 (2011).
20. Jones, E., Oliphant, E., Peterson, P. *et al.* Scipy: Open source scientific tools for python [<http://www.scipy.org/>]. *Go to reference article* (2019).
21. Hunter, J. D. Matplotlib: A 2d graphics environment. *Comput. science & engineering* **9**, 90–95 (2007).
22. Analytics, C. Anaconda software distribution. version 2.4. 0 (2016).

# Appendix

The code that was used in order to study different types of modeling in biological systems are shown here. This will help to reproduce the solutions to the models discussed above.

## Central Dogma

The code for the Central Dogma problem that is used in order to generate the plots is shown here:

```

1  # We need some modules (Python libraries) in the following.
2  # Usually, such modules are loaded in the first cell of a notebook.
3  # The modules that we need concern numerical calculations,
4  # solving the ordinary differential equations, and plotting them later.
5
6  # modules necessary for plotting
7  import matplotlib.pyplot as plt
8  # integrate a system of ordinary differential equations
9  # initial value problem
10 from scipy.integrate import odeint
11 # module to make available data structures and routines
12 # for numerics
13 import numpy as np
14
15 y0 = [0,0] # 0 protein and 0 mRNA
16 t = np.linspace(0,200,num=100) # 100 means hundred different point
17
18 k_m = 0.2
19 gamma_m = 0.05 #degradation rate should be less than production rate
20 k_p = 0.4
21 gamma_p = 0.1
22
23 #putting the params to array which will be passed in this offer
24 params = [k_m, gamma_m, k_p, gamma_p]
25
26 def dogma(variables, t, params):
27     m = variables[0]
28     p = variables[1]
29     #they are the first, second and so on element of the params
30     #arraygamma_m = params[1]
31     k_m = params[0]
32     k_p = params[2]
33     gamma_p = params[3]
34     dmdt = k_m - gamma_m * m
35     dpdt = k_p * m - gamma_p * p
36     return ([dmdt, dpdt])
37
38 #, is used for storing args and we get y matrix with two column one for p
39 #another for m and row for every time point
40 y = odeint(dogma, y0, t, args=(params,))
41 # to plot and label the plot
42 plt.plot(t,y[:,0], color="b", label="M")
43 plt.plot(t,y[:,1], color="r", label="P")
44 plt.xlabel("Abundance")
45 plt.ylabel("Time")
46 plt.legend()
47 plt.grid()
48 plt.show()

```

## Hill function

The code for the Hill function problem that is used in order to generate the plots is shown here:

### Activation Hill Function

```

1  import matplotlib.pyplot as plt
2  from scipy.integrate import odeint
3  import numpy as np
4
5  # 0 for G1 and 0 for G2
6  y0 = [0,0]
7  # 100 means hundred different point
8  t = np.linspace(0,200,num=100)
9
10 # degradation rate should be less than production rate
11 k_1 = 0.5
12 gamma_1 = 0.1
13 k_2 = 0.5
14 gamma_2 = 0.05
15 n = 5
16 c = 5
17
18 # putting the params to array which will be passed in this offer
19 params = [k_1, gamma_1, k_2, gamma_2, n, c]
20
21 def activation(variables, t, params):
22     #they are the first, second and so on element of the params
23     # and arraygamma_m = params[1]
24     G1 = variables[0]
25     G2 = variables[1]
26     k_1 = params[0]
27     gamma_1 = params[1]
28     k_2 = params[2]
29     gamma_2 = params[3]
30     n = params[4]
31     c = params[5]
32     dG1dt = k_1 - gamma_1 * G1
33     dG2dt = (G1**n / (c**n + G1**n)) * k_2 - gamma_2 * G2
34     return ([dG1dt,dG2dt])
35
36 #,is used for storing args and we get y matrix with two column one
37 # for G1 another for G2 and row for every time point
38
39 y = odeint(activation, y0, t, args=(params,))
40
41 plt.plot(t , y[:,0], color="b", label="G1")
42 plt.plot(t , y[:,1], color="r", label="G2")
43 plt.xlabel("Time")
44 plt.ylabel("Number")
45 plt.legend()
46 plt.grid()
47 plt.show()

```

### Repression Hill Function

```

1  import matplotlib.pyplot as plt
2  from scipy.integrate import odeint

```

```

3 import numpy as np
4
5 # 0 for G1 and 0 for G2
6 y0 = [0,0]
7 # 100 means hundred different point
8 t = np.linspace(0,200,num=100)
9
10 #degradation rate should be less than production rate
11 k_1 = 0.5
12 gamma_1 = 0.1
13 k_2 = 0.5
14 gamma_2 = 0.05
15 n = 5
16 c = 5
17
18 #putting the params to array which will be passed in this offer
19 params = [k_1, gamma_1, k_2, gamma_2, n, c]
20
21 def repression(variables, t, params):
22     #they are the first, second and so on element of the params
23     #and arraygamma_m = params[1]
24     G1 = variables[0]
25     G2 = variables[1]
26     k_1 = params[0]
27     gamma_1 = params[1]
28     k_2 = params[2]
29     gamma_2 = params[3]
30     n = params[4]
31     c = params[5]
32     dG1dt = k_1 - gamma_1 * G1
33     dG2dt = (c*n / (c*n + G1*n)) * k_2 - gamma_2 * G2
34     return ([dG1dt,dG2dt])
35
36 #, is used for storing args and we get y matrix with two column one for G1
37 # and another for G2 and row for every time point
38 y = odeint(repression, y0, t, args=(params,))
39
40 plt.plot(t , y[:,0], color="b", label="G1")
41 plt.plot(t , y[:,1], color="r", label="G2")
42 plt.xlabel("Time")
43 plt.ylabel("Number")
44 plt.legend()
45 plt.grid()
46 plt.show()

```

## Oscillating Gene Network

The code for the Oscillating gene network problem that is used in order to generate the plots is shown here:

```

1 import matplotlib
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4 import numpy as np
5
6 y0 = [0,0,0]
7 t = np.linspace(0,200, num=100)
8

```

```

9   k_1 = 0.5
10  gamma_1 = 0.1
11  k_2 = 0.5
12  gamma_2 = 0.1
13  k_3 = 0.5
14  gamma_3 = 0.1
15  n = 9
16  c = 1
17
18  params = [k_1, gamma_1, k_2, gamma_2, k_3, gamma_3, n, c]
19
20  def sim(variables, t, params):
21      G1 = variables[0]
22      G2 = variables[1]
23      G3 = variables[2]
24      k_1 = params[0]
25      gamma_1 = params[1]
26      k_2 = params[2]
27      gamma_2 = params[3]
28      k_3 = params[4]
29      gamma_3 = params[5]
30      n = params[6]
31      c = params[7]
32      dG1dt = (c**n / (c**n + G3**n)) * k_1 - gamma_1 * G1
33      dG2dt = (G1**n / (c**n + G1**n)) * k_2 - gamma_2 * G2
34      dG3dt = (G2**n / (c**n + G2**n)) * k_3 - gamma_3 * G3
35      return ([dG1dt, dG2dt, dG3dt])
36
37  y = odeint(sim, y0, t, args=(params,))
38
39  f, (ax1, ax2, ax3) = plt.subplots(3, sharex=True, sharey=False)
40  line1, = ax1.plot(t, y[:,0], color="b", label="G1")
41  line2, = ax2.plot(t, y[:,1], color="r", label="G2")
42  line3, = ax3.plot(t, y[:,2], color="g", label="G3")
43  ax1.set_ylabel('Number')
44  ax1.set_xlabel('Time')
45  ax1.legend(handles=[line1, line2, line3])
46  plt.grid()
47  plt.show()

```

## Gillespie Algorithm

The code for the Gillespie algorithm that is used in order to generate the plots is shown here:

```

1  import matplotlib.pyplot as plt
2  from scipy.integrate import odeint
3  import numpy as np
4  import random
5
6  x = [0] # x represent mRNA
7  t = [0] # t represnt time point
8
9  tend = 1000 # simulation end time
10
11  k = 2 # production rate
12  gamma = 0.1 # degradation rate
13

```

```

14 while t[-1] < tend:
15     # gives latest value of x
16     current_x = x[-1]
17     rates = [k, gamma * current_x]
18     rate_sum = sum(rates)
19     # tau is the time points for next event
20     tau = np.random.exponential(scale=1/rate_sum)
21     t.append(t[-1] + tau)
22     rand = random.uniform(0,1)
23     # production event
24     if rand * rate_sum > 0 and rand * rate_sum < rates[0]:
25         x.append(x[-1] + 1)
26     # production event
27     elif rand * rate_sum > rates[0] and rand * rate_sum < rates[0] + rates[1]:
28         x.append(x[-1] - 1)plt.plot(t,x)
29
30 plt.xlabel("time")
31 plt.ylabel("mRNA quantity")
32 plt.grid()
33 plt.show()

```

## Deterministic vs Stochastic modelings

The code for the Deterministic vs Stochastic modelings problem that is used in order to generate the plots is shown here:

```

1 import matplotlib.pyplot as plt
2 from scipy.integrate import odeint
3 import numpy as np
4 import random
5
6 G1 = [0] # Gene first time interval
7 G2 = [0] # Gene second time interval
8 G3 = [0] # Gene third time interval
9 t = [0] # time point of new event
10
11 tend = 1000
12
13 k_1 = 2
14 gamma_1 = 0.1
15 k_2 = 2
16 gamma_2 = 0.1
17 k_3 = 2
18 gamma_3 = 0.1
19 n = 9
20 c = 1
21
22 # -1 means last value of t array
23 while t[-1] < tend:
24     current_G1 = G1[-1]
25     current_G2 = G2[-1]
26     current_G3 = G3[-1]
27     rates = [(c**n / (c**n + current_G3**n)) * k_1, gamma_1 * current_G1,\
28             (current_G1**n / (c**n + current_G1**n)) * k_2, gamma_2 * current_G2,\
29             (current_G2**n / (c**n + current_G2**n)) * k_3, gamma_3 * current_G3]
30     rate_sum = sum(rates)
31     # tau is the next point of time for event
32     # the scale is the mean of distribution

```

```

33     tau = np.random.exponential(scale=1/rate_sum)
34     t.append(t[-1] + tau)
35     rand = random.uniform(0,1)
36     # G1 production event
37     if rand * rate_sum < rates[0]:
38         G1.append(G1[-1] + 1)
39         G2.append(G2[-1])
40         G3.append(G3[-1])
41     # G1 decay event
42     elif rand * rate_sum > rates[0] and rand * rate_sum < sum(rates[:2]):
43         G1.append(G1[-1] - 1)
44         G2.append(G2[-1])
45         G3.append(G3[-1])
46     # G2 production event
47     elif rand * rate_sum > sum(rates[:2]) and rand * rate_sum < sum(rates[:3]):
48         G1.append(G1[-1])
49         G2.append(G2[-1] + 1)
50         G3.append(G3[-1])
51     # G2 decay event
52     elif rand * rate_sum > sum(rates[:3]) and rand * rate_sum < sum(rates[:4]):
53         G1.append(G1[-1])
54         G2.append(G2[-1] - 1)
55         G3.append(G3[-1])
56     # G3 production event
57     elif rand * rate_sum > sum(rates[:4]) and rand * rate_sum < sum(rates[:5]):
58         G1.append(G1[-1])
59         G2.append(G2[-1] )
60         G3.append(G3[-1] + 1)
61     # G3 decay event
62     elif rand * rate_sum > sum(rates[:5]) and rand * rate_sum < sum(rates[:6]):
63         G1.append(G1[-1])
64         G2.append(G2[-1] )
65         G3.append(G3[-1] - 1)
66
67 f, (ax1, ax2, ax3) = plt.subplots(3, sharex=True, sharey=False)
68 line1, = ax1.plot(t , G1, color="b",label="G1")
69 line2, = ax2.plot(t , G2, color="r",label="G2")
70 line3, = ax3.plot(t , G3, color="g",label="G3")
71 ax1.set_ylabel("Number")
72 ax3.set_xlabel("Time")
73 ax1.legend(handles=[line1,line2,line3])
74 ax1.grid()
75 ax2.grid()
76 ax3.grid()
77 plt.show()

```