Stochastic Modelling of Gene Regulation with microRNAs

Martina Marbà Maya

A Dissertation Submitted to the
University of Glasgow
for the degree of
Master of Science

Department of Statistics

May 2009

Abstract

The experimental and computational studies of microRNAs, a novel class of gene regulators, discovered relatively recently, is a rapidly growing field. In particular, researchers are focusing on identifying the targets of microRNAs and the roles of microRNAs in post-transcriptional gene regulation; the work presented in this thesis is a contribution to this field. Here, a range of kinetic models of gene regulation is studied computationally with a view to explore and predict the stochasticity in gene expression and to review the hypothesis that, in addition to reducing levels of target mRNA and proteins, microRNAs tune down the noise in protein output. Previously, it has been shown that other factors such as activation and deactivation rates of gene promoters have a direct effect on the variation of gene expression and the effect of microRNAs on protein output from different promoters is directly studied here. In addition, our methodology allows for a comparison of transcriptional and post-transcriptional modes of gene regulation. Finally, a model is proposed for the study of more realistic problems of many targets.

The challenging motivation of this thesis is the use of different statistical methods to explore gene expression and noise in protein output. Stochastic numerical simulations have been compared to theoretical analysis, such as the Probability Generating Function Approach and the method of matrices developed by Gadgil *et al*, showing similar results for the magnitude of noise in different systems. The Langevin Equation and Tau-Leaping methods (for which Matlab codes are developed here) are shown to be excellent approximations to the Gillespie Algorithm.

Acknowledgements

Firstly, I would like to thank my supervisor, Dr Raya Khanin, for having introduced me to bioinformatics, for all the knowledge she has transmitted me, and for the motivation she has always had in my work, as without her, this thesis would never have become what it is. Thank you for the patience and energy you granted me during all this time, and in spite of the difficult start which I had. I would also like to thank the University of Glasgow, and in particular the Statistics department, for having financed me and permitted me this grand opportunity to initiate myself with this research. Above all, I would thank Dr Agostino Nobile who sewed in my mind the idea of doing this Masters, and also Dr Ben Torsney who, with his sense of humour, made me smile at every encounter in the department.

Furthermore, I wish to thank Dr Douglas Macdonald for having supported my ups and downs throughout this last year, for the long discussions of mathematical-theoretical problems, and for his great influence in the planing of this work. I give thanks to my all friends: Sabela, Mia, Marta, Carme, Harry, Pau, David and Phil, for so many good moments in this city so rainy and so dear to me, for sharing and enjoying with me those times of diversion, and for showing me another side of living and learning.

Finally, I dedicate this paragraph to my mother, my father and my sister, to whom I owe everything, and without whom I would probably never have reached where I am now. Thank you for leaving me freedom of expression, for so many dialogues exchanged, and for listening to me and bestowing me with your confidence and trust. Thank you for being by my side!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

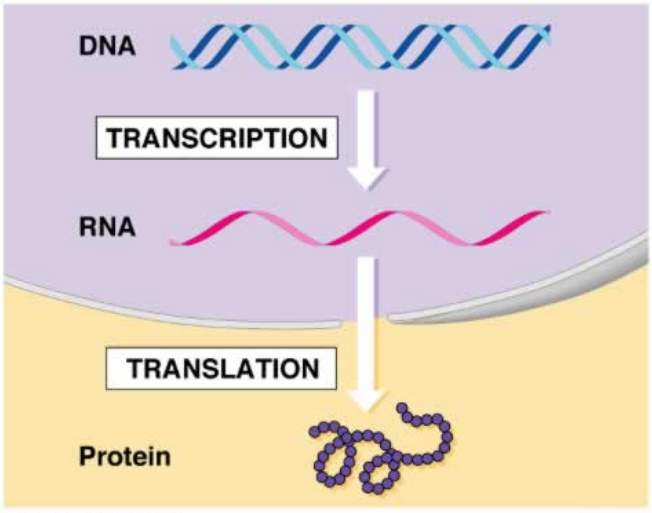
1.1 Biological Background

The most important element in all living organisms is the cell since it plays a main role in the organism's structure and functionality. In fact, the cell is the smallest unit considered in a living organism and is classified in two broad groups: prokaryotes and eukaryotes. Some organisms, such as most bacteria, are unicellular, consisting of a single cell and often referred to as prokaryotes while bigger organisms, such as humans, plants, trees and yeast are multi-cellular or eukaryotes, which are much more internally complex than prokaryotes. Both groups have certain features in common, such as DNA, genomic part of the cell, and RNA. The DNA carries the hereditary information via genes, while RNA, transcribed from DNA, contains the necessary information to synthesize proteins and other vital molecules.

Proteins are essential parts of organisms that participate in every process within cells, therefore protein synthesis (that starts with gene expression) plays a vital role in the living organism. This also depends on lots of factors and regulators that are themselves products of the gene expression. In fact, one single cell can have thousands of proteins, resulting in thousands of interconnections between proteins, and thousands of complex processes such as protein synthesis. So, the overall picture of the cell is very complex.

Despite all this general complexity, a two-step process can optimally characterise the action of protein production: **transcription** and **translation**. The processes of transcription and translation are carried out in the cell, to read each gene and produce the protein. In transcription a phase one strand of **DNA** molecule, the genetic information, is copied into a complementary RNA strand. This RNA strand is then processed to the short-lived **mRNA** (messenger RNA). When the transcription is finished, the portion of the DNA that coded for a **protein**, i.e. a gene, is then represented by a mRNA molecule that can be used as a template for translation. Translation is the second step process of making proteins. It is the process that decodes the transcribed mRNA to produce the relevant protein. (Figure 1.1 illustrates the two-step process that describes gene expression.)

Figure 1.1. Gene Expression: transcription-translation



This figure describes the two step process of gene expression (protein production). DNA molecules are translated into mRNA by transcription. Protein is synthesized from mRNA molecules by transcription.
`http://www.anselm.edu/homepage/jpitocch/genbio/transcrtransl.JPG`

This complex process of gene expression exists in every cell of an organism and allows the production of protein is often referred to as the Central Dogma of Biology.

1.2 MicroRNA

MicroRNAs (miRNAs) have emerged as a new class of gene regulatory elements in plants and animals. Their definitive functions are not yet clearly specified and experimental identification of miRNA targets is difficult. This is why over the past decade the focus of interest of many researchers has been into identifying the miRNAs ranges of various organisms tissues, developmental processes, and predicting the mRNA targets of these numerous miRNAs. It is known that miRNAs act post-transcriptionally, influencing the stability, compartmentalization, and translation of their target mRNAs (Carthew 2006) but the mechanisms of these processes have not yet been identified.

Mature miRNAs are small non-coding RNA molecules that regulate the expression of messenger RNAs (mRNAs). Made up of approximately 22 nucleotides, miRNAs are estimated to comprise 1% - 5% of animal genes, making them one of the most abundant classes of regulators, and their regulatory impact more prevalent than was previously suspected. Their widespread and important role in animals is highlighted by recent estimates that up to 30% of an organism's protein-coding genes are subject to miRNA-mediated control (Lewis et al. 2005, Stark et al. 2005) and is evidenced by their evolutionary conservation (Rajewsky 2006). Recent advances indicate that the miRNAs play a significant role in many biological processes such as developmental timing, cell proliferation, differentiation, apoptosis, metabolism, and morphogenesis (Ambros 2004, Carthew 2006).

MiRNAs function differently in plants than they do in animals; for example in plants, miRNAs inhibit a target mRNA by almost perfect base-pairing to complementary sequences (Standart & Jackson 2007); whereas in animals, they tend to make imperfect partial base pair contacts with their target transcript (Rajewsky 2006).

The miRNA can exert its regulating effect on the target post-transcriptionally that means on mRNA degradation and/or on protein translation. This mechanism is an efficient way of regulating the production of a diverse range of proteins.

1.3 Stochasticity/Noise in gene expression

Stochastical variations occur inside the cells. The stochasticity arises from fluctuations in transcription and translation of gene expression, despite constant environmental conditions, and it is measured by the amount of noise (variation) in the expression of mRNA, protein, or other involved reactants. This means that two identical cells can show different expressions. Furthermore, a unique cell is expressed differently from time to time because of the **intrinsic stochasticity**.

Intrinsic noise refers to the stochasticity that comes from genetically identical cells and organisms, with identical environmental exposures, which exhibit remarkable diversity because of the random nature of the biochemical reactants such as the finite number of molecules of the reactants of gene expression. In terms of noise of protein in a cell, this means the variance between levels of protein (or number of protein molecules), implying that sometimes the same protein will be expressed in larger amounts and other times in smaller amounts.

In order for everything to work properly and all the successive processes to take place (which depend on the amount of protein), the cells need to receive a ranged amount of protein. If a cell contains different levels of protein, consequently the noise is large, and the processes dependent on the protein might not occur.

The noise, variability among equal populations, can be a disadvantage and produces generally detrimental effects on cellular function with potential implications for diseases. For example diseases such as Cancer: in the theory of the origin of cancer, some mechanisms stop working and important genes get out of control producing lots of noise in the protein expression. However, noise can also be an advantage in, for example, cell flexibility: bacteria, when submitted to a certain change of environmental conditions (such as temperature) should die, but instead survive because of the noise production in their protein.

1.4 Aims

This project deals with the study of stochasticity in gene expression and its regulation with miRNA. The stochastic phenomenon in gene expression has attracted interest for several years because of its implications for cellular regulation and non-genetic individuality. Also, in the last few years, the novel mode of gene regulation by miRNAs has raised many interesting questions and speculations regarding their roles in the cellular regulatory networks. One of the roles that is ascribed to miRNAs is that they tune gene expression and protein production. That is why one of the important questions which will be addressed here is whether miRNAs affect (reduce) noise in gene expression and optimize the level of the protein.

Therefore, the goal of this thesis is to elucidate some of the general principles and mechanisms of microRNA action. To this end, models of the gene-regulatory circuits that involve microRNAs will be developed to predict and evaluate the dependence of the intrinsic noise of the protein output on system parameters and find out which factors regulate (minimise, or attenuate) this variation.

To answer these questions, several models will be proposed to describe some circuits of the biological systems and different analyses will be carried on to study with them. Stochastic numerical simulations, such as Gillespie Algorithm, and Langevin Equation (Higham 2007), will be used as well as more efficient numerical implementation, $\tau-$ leaping simulation (Gillespie 2001). These statistical techniques will be compared against other theoretical analysis such as the Probability Generating Function Approach (Takasu 2005) or the methodology used in Gadgil et al. (2005), to tackle in distinct ways through the nature of the problem.

1.5 Thesis Overview

Chapter 1 explains the biological problem of interest that is analysed in this thesis. This chapter introduces basic explanations about the concepts involved in this work such as gene expression, gene regulators such as microRNA (miRNA), and stochasticity of gene expression. The aims of the study are also stated.

In chapter 2, different models of gene expression used in this thesis are described. This chapter is divided in two blocks: models of transcriptional gene regulation that do not take into account miRNAs and models of post-transcriptional gene regulations where miRNA can regulate a target gene. The first part of this chapter introduces a standard model for gene expression, while the second part discusses the models with miRNA where the gene circuit structure depends on the number of targets that the regulator miRNA has.

Chapter 3 reports the methodology used to simulate the chemically reacting systems presented here. The methodology is divided in two types of analysis, numerical and theoretical analysis. Both types of analysis with various examples demonstrate that there exists a good approximation between the two methods that enable estimating expressions for the noise of the reactants.

Chapter 4 shows the results obtained from several *in-silico* experiments, where the noise and the effect of miRNA and other parameters of the models on protein output and target mRNA are explored. In this chapter, the models and the methodology introduced in the two previous chapters are used.

Finally, chapter 5 exposes and discusses the roles that this novel class of regulator has, highlighting the effect of miRNA on reducing noise in protein output in addition to reducing protein levels. This chapter concludes that statistical techniques used in this thesis are suitable for studying noise in gene expression, and can be extended to larger models with good approximated results.

Chapter 2

Gene Expression Models

"All models are wrong-but some models are useful"
(George P. E. Box)

This chapter introduces different models of gene expression used in this project to study gene regulation and stochastic fluctuations in this process. The models range from the basic model that considers only two major processes of transcription and translation up to more complex ones, ranging from, for example, studies which uses the *1 target mRNA model*, that studies only one target independently of anything else, to one that uses the model called *N target mRNA model*. But principally most of the work is centered on the study of the *one target model* as it is simpler, and can be used as an introduction to models with higher number of targets as it is done in this thesis.

This chapter introduces microRNAs and their effect into the models, in order to study the stochastic changes on the levels of mRNA, and proteins.

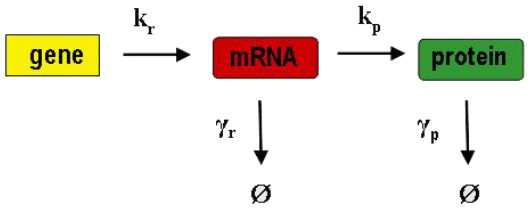
2.1 Models of transcriptional gene regulation

2.1.1 Simple Model (Prokaryotic Model)

The Simple Model gives a basic mathematical representation of gene expression (or transcription), taking the minimum number of reactions and variables needed to represent all of the essential features of transcription, translation and interactions between genes in a regulatory network (see model in Figure 2.1). This model was introduced by Thattai & van Oudenaarden (2001). Moreover, it is associated with prokaryotic expression (eg: cells in plants, or bacteria).

The model describes the fundamental two-step process of transcription and translation that a gene, continuous stretch of a genomic DNA molecule, executes to make the protein. To better understand this process of protein synthesis, the two stages have to be clear: transcription is the process in which DNA (gene) is transcribed into mRNA (messenger RNA) and translation is when mRNA is translated into protein. The degradations of both mRNA and protein are also included in this model. Figure 2.1 shows a graphic description of this model.

Figure 2.1. Simple model of gene expression



Simple model of gene expression. Transcription of the *gene* into mRNA , mRNA translation into proteins and both decays of mRNA and protein molecules are described with k_r, k_p, γ_r and γ_p, the respective rate constants.

Each of the different four reactions mentioned above have a rate constant associated with it: k_r, transcription; k_p, translation; γ_r, mRNA degradation and γ_p, protein degradation. The rate constants define the velocity or rate, at which such

molecules change from one type to another, or are produced or degraded. At the same time, these rates regulate each of the processes implied in this circuit.

Stochastic Reactions for a *Simple Model*

The stochastic *Simple Model* describes explicitly two stochastic birth and death processes of the species, or reactants, mRNA and protein:

$$1) \quad \varnothing \xrightarrow{k_r} mRNA \xrightarrow{\gamma_r \langle M \rangle} \varnothing$$

$$2) \quad \varnothing \xrightarrow{k_d \langle M \rangle} protein \xrightarrow{\gamma_p \langle P \rangle} \varnothing$$

Each of these processes are compounds of two reactions that correspond to the birth and death of the implied reactants successively and different associated rates that depend on the type of reactions. For example:

$$\varnothing \xrightarrow{k_r} mRNA$$

represents transcription (or mRNA production), where k_r is the rate at which mRNA is produced. Degradation of mRNA is represented by:

$$mRNA \xrightarrow{\gamma_r \langle M \rangle} \varnothing,$$

where γ_r is the rate of linear degradation (or decay). The total rate of mRNA decay is γ_r multiplied by $\langle M \rangle$, because it is the number of available mRNA molecules.

Precisely, the two processes are split up in four first-order reactions which describe each step as a single random event. The first-order reactions can be categorized in four types: production from a source, degradation, conversion, or catalytic production from a source. These categories with their correspondent rate constants can be found in the study by Gadgil et al. (2005, pg. 911).

The four reactions are the following:

$$\begin{array}{ll} 1) \quad \varnothing \xrightarrow{k_r} M \;\Big\}\; \text{Production from a source} \\ 2) \quad \varnothing \xrightarrow{k_p\langle M\rangle} P \;\Big\}\; \text{Catalytic production from a source} \\ \left.\begin{array}{l} 3) \quad M \xrightarrow{\gamma_r\langle M\rangle} \varnothing \\ 4) \quad P \xrightarrow{\gamma_p\langle P\rangle} \varnothing \end{array}\right\} \text{Degradation reactions} \end{array} \tag{2.1}$$

where mRNA, protein, mRNA mean, and mean of protein are encoded by M, P, $\langle M\rangle$ and $\langle P\rangle$ successively. In the *Simple Model*, mRNA and protein are the only two species of interest, and in fact, they are the only measurable species in this model.

Deterministic equations (ODE's) for a *Simple Model*

This system is described by the following rate equations:

$$\begin{cases} \dfrac{dM}{dt} &= \quad k_r - \gamma_r M \\ \dfrac{dP}{dt} &= k_p M - \gamma_p P \end{cases} \tag{2.2}$$

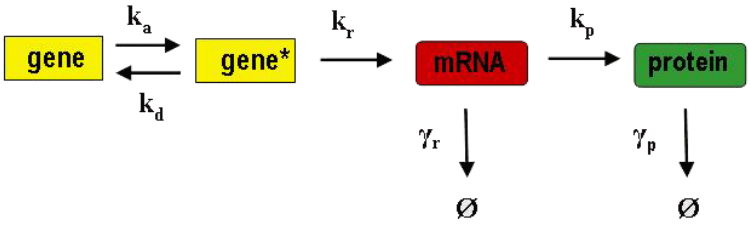
where M and P are concentrations of mRNA and protein, respectively. Other parameters are the same as in system 2.1.

2.1.2 General Model (Eukaryotic Model)

The General Model includes two states of the gene's promoter in addition to transcription and translation. Presented in Figure 2.2, it was originally constructed

by Raser & O'Shea (2004), and includes two states of the gene's promoter in addition to the two-step processes of transcription and translation, and the respective decays of mRNA and protein. This model is associated with the eukaryotic expression (eg: cells from higher organisms, such as the human organism.)

Figure 2.2. General model with two states of the gene's promoter



Two states of the gene's promoter (*gene*, inactive state; *gene**, active state). The *gene** is capable of transcription (or mRNA synthesis). Transition between these two states is reversible and the total number of promoters is conserved, i.e. $gene + gene^* = n_g$. Transcription takes place from *gene**, and all the other reactions and constants are equal to those in Figure 2.1.

The two states of the promoter considered are the inactive state of the gene, *gene*, where transcription can not take place, and the active state, *gene**, that permits the transcription activation. Transition between these two states is reversible and total number of promoters is conserved, i.e. $gene + gene^* = n_g$, where n_g is the total number of gene copies.

Promoter Types

Three kinetic mechanisms of promoter transcriptional activation are distinguished in this model Raser & O'Shea (2004).

- In case *1*, the activation step is infrequent relative to transcription and the active promoter state is **stable**. The rate constants of this first mechanism are: $k_a, k_d << k_r$, where k_a is the activation rate of the promoter, k_d the deactivation rate, and k_r the transcription rate.

- In case *2*, the activation is again infrequent, but a rapid reversion to the inactive state and the active promoter state is **unstable**. The rate constants in this second case satisfy: $k_a << k_r$ and $k_d >> k_r$, which implies a relatively infrequent to be in the active state of the promoter. The promoter is more often in the inactive state than in the active one.

- In case *3*, the activation step is frequent due to the rate constants: $k_a, k_d >> k_r$, but the promoter changes very quickly between different states. This promoter is called **prokaryotic**. Because of the rapid rate constants of activation/inactivation of the promoter, the transcription only occurs during a fraction of the events that change the promoter from one state to another. Moreover the model actually reduces to *Simple Model* considered before. (See the promoter types summarized in table 2.1)

Table 2.1. Promoter types

Promoters	Rate Relationships
Stable (1)	$k_a, k_d \ll k_r$
Unstable (2)	$k_a \ll k_r$, $k_d \gg k_r$
Prokaryotic (3)	$k_a, k_d \gg k_r$

Promoter types with the respective rate constants; (k_a is activation rate promoter constant, k_d is inactivation rate promoter constant, k_r transcription).

In terms of molecular biology, in case of the *stable* promoter, the probability to bind DNA* (active promoter state of the gene) and to begin the transcription is higher than for the *prokaryotic*, because once a molecule is in the active state, it takes more time until the reverse reaction is successful because the ratio for this reverse reaction is small. Instead, for the *prokaryotic* promoter, when the rates are larger there is more chance for DNA to go from one state to the other without stopping for an interval of time in one place. *Unstable* promoter however is the case where the activation step is much smaller than the inactivation, and consequently the promoter has smaller probability to be in the active state, DNA*, in comparison to two other promoters.

Stochastic Reactions for a *General Model*

This model has a total of six first-order reactions. The first four are equal to the reactions from the *Simple Model* (in system 2.1), and two remaining reactions corresponds to transitions between the two states of the gene.

These two new reactions are:

$$\left.\begin{array}{ll} 5) & D \xrightarrow{k_a\langle D\rangle} D^* \\ 6) & D^* \xrightarrow{k_d\langle D^*\rangle} D \end{array}\right\} \text{Conversion reactions} \tag{2.3}$$

DNA and DNA^* are encoded by D and D^*, and $\langle D\rangle$, $\langle D^*\rangle$ are their respective means. Then, the system of reactions that describes the *General Model* is a described by the systems 2.1 - 2.3, where the species of interest are: DNA (gene inactive), DNA^* (gene active), mRNA and protein.

Deterministic equations (ODE's) for a *General Model*

This system is described by the following rate equations:

$$\left\{\begin{array}{rcl} \dfrac{dD^*}{dt} &=& k_a D - k_d D^* \\ \dfrac{dM}{dt} &=& k_r D^* - \gamma_r M \\ \dfrac{dP}{dt} &=& k_p M - \gamma_p P \end{array}\right. \tag{2.4}$$

where D and D^* are the fractions of the available inactive and active gene copies respectively. M and P are concentrations of mRNA and protein, respectively. The other parameters are the same as in systems 2.2 and 2.3.

Because the total number of genes is constant:

$$D_T = D + D^* \qquad \Rightarrow \qquad D = D_T - D^* \tag{2.5}$$

we can compute D from this equation, where D_T is the total number of gene copies.

2.2 Models of post-transcriptional gene regulation by MicroRNAs

In a basic microRNA-mediated post-transcription circuit a gene is regulated by a microRNA. One miRNA has many (up to hundreds) target genes, but for simplicity only ***one* target** will be considered to start with and will be described by two models, the *Simple-miRNA Model* and the *General-miRNA Model*. Afterwards the models that consider ***two* targets** will also be described and can be easily extended to ***N* targets**. In fact, the models of *two* targets or N will be similar to the *one* target model, in terms of the types of reactions, the reactants of the system and parameters. But as the name indicates, the miRNA will have *two* or more targets, wich implies nearly double reactions, reactants and parameters for *two* targets, or much more of them for N targets, as well as interdependence between the targets.

The miRNA can exert its regulating effect post-transcriptionally on the target (or targets, but just one target will be considered for simplicity) on mRNA degradation and/or on protein translation (see Figure 2.3). These loci of miRNA action, allocated on the post-transcriptional steps of gene expression, can be divided into the three different modes of regulation described in Table 2.2 (see also Figure 2.3): *Case1*, there is no miRNA neither in mRNA degradation or protein production; *Case2*, there is miRNA but only in the degradation of mRNA; *Case3*, miRNA is either present in mRNA degradation and protein production.

Table 2.2. MiRNA cases by location

	Location**		
Cases	No	mRNA	Protein
1	✓		
2		✓	
3		✓	✓

The symbol ✓ indicates if a column is contained or not in the case.
Location**: *No* → There is no presence of miRNA; *mRNA* → miRNA present in mRNA decay; *Protein* → miRNA present in translation.

Therefore, when a transcript is a target of a specific miRNA, the rates of mRNA degradation and translation are not longer independent of miRNA level and they will be expressed by γ_r^* and k_p^*, respectively, where the symbol * denotes that they depend on miRNA. In the previous section, these two rates for gene transcription models were considered constant and defined as γ_r and k_p.

Explicitly, miRNA enhances the rate of mRNA degradation and/or inhibits the protein production rate. Plausible models for miRNA-mediated mRNA target degradation assume that the rate of mRNA degradation depends linearly on the miRNA level (Khanin & Vinciotti 2008),

$$\gamma_r^* = \gamma_r(1 + d \cdot m) \tag{2.6}$$

where $d \geq 0$ is the miRNA-mediated fold-change in the target mRNA degradation rate relative to the basal degradation rate, γ_r, which does not depend on miRNA. MiRNA levels are described by m, number of miRNA molecules, though for the models with one target different levels of miRNA are not considered (assumption that the miRNA level is constant), and then m indicates whether miRNA is present can be set to *one* if it is present without loss generality (and *zero* if it is absent). If miRNA does not affect this target degradation rate, then $d = 0$, and the mRNA degradation rate is γ_r.

The miRNA-mediated translational repression can be described by:

$$k_p^* = \frac{k_p}{K + a_m m} \tag{2.7}$$

where K is the half-saturation concentration or level of miRNA at which half of the maximum effect (downregulation) is achieved. The other variable a_m is Boolean and takes the values 0 or 1, depending if miRNA regulates mRNA degradation (then it would be $a_m = 0$) or it regulates both rate constants: mRNA decay and translation (being then $a_m = 1$). For convenience the translation rate, k_p, that is in the numerator will be converted to $k_p \cdot K$, as follows $k_p^* = k_p K/(K + a_m m)$, thus if miRNA is not present $m = 0$ the translation rate is k_p.

2.2.1 1 mRNA target Model: Simple-miRNA Model

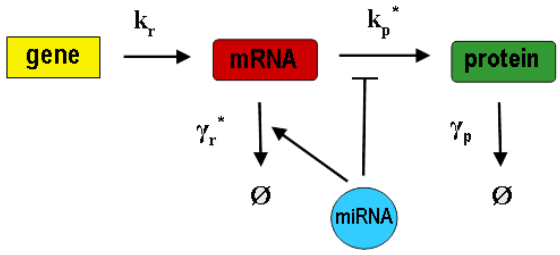
The Simple-miRNA Model is an implementation of the *Simple Model* with miRNA. This model only adds the miRNA molecule to the *Simple Model*, and it considers the changes of the relative rate constants affected for miRNA (see figure 2.3). In particular, the model describes the process of miRNA-mediated regulation of mRNA degradation and/or translation.

Stochastic Reactions for a *Simple-miRNA Model*

This model consists of four first-order reactions, the same ones as in the *Simple Model* (system 2.1) but with the rates of mRNA degradation and translation being dependent on miRNA in the model.

The four reactions are the following:

Figure 2.3. Simple-miRNA Model



This model is an implementation of the *Simple Model*, with *miRNA*. Transcription and protein decay have associated the rates k_r and γ_p respectively, as in the *Simple Model*. But the rate constants affected by miRNA are mRNA degradation and protein production with a_m=0,1 and the respective rates:

$$\gamma_r^* = \gamma_r(1 + dm), k_p^* = \frac{k_p K}{K + a_m m}.$$

$$\begin{aligned}
&1)\quad \varnothing \xrightarrow{k_r} M \quad \Big\} \text{ Production from a source} \\
&2)\quad \varnothing \xrightarrow{k_p^*\langle M\rangle} P \quad \Big\} \text{ Catalytic production from a source} \\
&3)\quad M \xrightarrow{\gamma_r^*\langle M\rangle} \varnothing \\
&4)\quad P \xrightarrow{\gamma_p\langle P\rangle} \varnothing \quad \Big\} \text{ Degradation reactions}
\end{aligned} \tag{2.8}$$

The only differences to the *Simple Model* in system (2.1) are the constants γ_r^*, and k_p^* of mRNA decay and protein production, because here they are dependent on miRNA and defined as in 2.6 -2.7.

Deterministic equations (ODE's) for a *Simple-miRNA Model*

The equations for this model are:

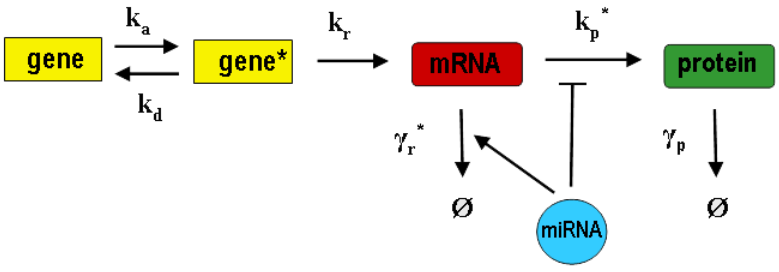
$$\begin{cases} \dfrac{dM}{dt} &= k_r - \gamma_r^* M \\ \\ \dfrac{dP}{dt} &= k_p^* M - \gamma_p P \end{cases} \tag{2.9}$$

where M and P are concentrations of mRNA and protein, respectively. The other parameters are the same as in system 2.8.

2.2.2 1 mRNA target Model: General-miRNA Model

The General-miRNA Model is the implementation of the *General Model* with additional miRNA effect. Or, it is an implementation of the *Simple-miRNA Model* but adding a promoter with two states. Therefore, this model describes the process of gene expression with two states of promoter and with miRNA-mediated regulation of mRNA degradation and/or translation. Again, miRNA enhances the rate of mRNA degradation and inhibits the protein production rate (see figure 2.4 for an ilustration of the model.)

Figure 2.4. General-miRNA Model



The two states of the gene's promoter (*gene*, inactive state; *gene**, active state), such as promoter from Figure 2.2. MiRNA actuates either in mRNA degradation and/or in translation, with the same rates as Figure 2.3 such as transcription and protein decay that are equal to the *Simple-miRNA model*.

Stochastic Reactions for a *General-miRNA Model*

This model splits up in six first-order reactions in the same way as the *General Model*. This system can be described by the combination of the reactions of the *Simple-miRNA Model* and the reactions for transition between gene's promoter states. Then, the *General-miRNA Model* is defined as the combination of the systems 2.8 -2.3.

Deterministic equations (ODE's) for a *General-miRNA Model*

This system is described by the following rate equations:

$$\begin{cases} \dfrac{dD^*}{dt} &= k_a(1-D^*) - k_d D^* \\ \dfrac{dM}{dt} &= k_r D^* - \gamma_r^* M \\ \dfrac{dP}{dt} &= k_p^* M - \gamma_p P \end{cases} \tag{2.10}$$

where D^* is the fraction of the available active gene copies, and M and P are concentrations of mRNA and protein, respectively. The parameters are the same as in systems 2.8 -2.3.
(Note: Here the symbol * does not represent the same in all the terms, for D^* represents gene in active state and for γ_r^*, k_p^*, constants dependent on miRNA.)

2.2.3 2 mRNA targets Model

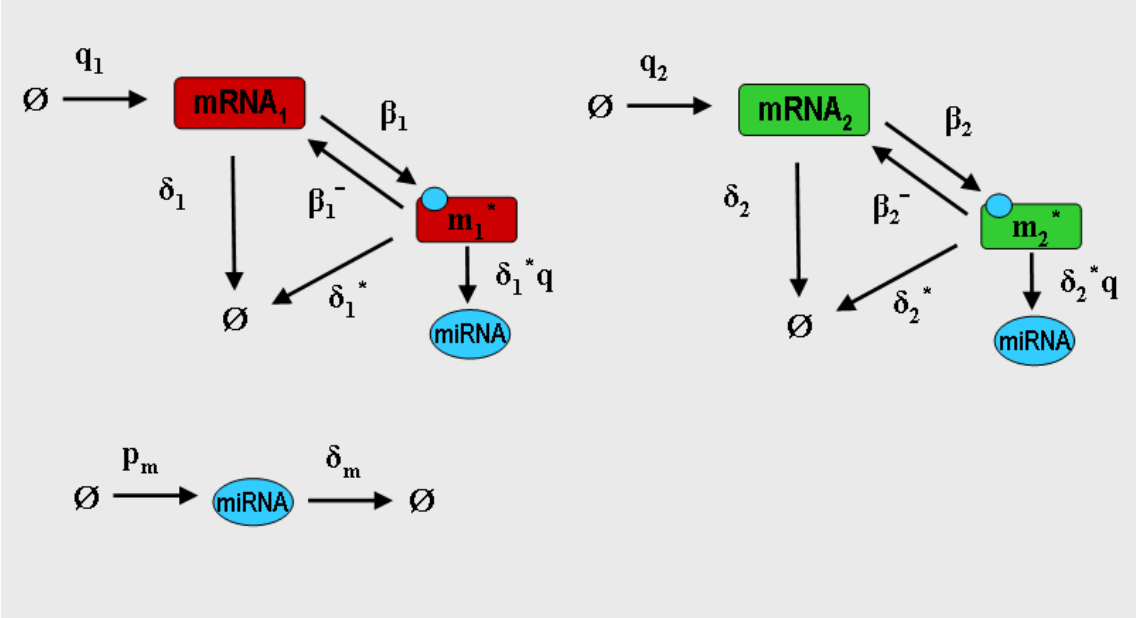
In this section, the two mRNA targets model introduced describes miRNA-mediated repression of two targets. Again for simplicity the models that will be studied in this section will be reduced to simple but representative models of one miRNA with two targets.

The *Two mRNA targets Model* illustrated in Figure 2.5 can be introduced as a simple-double version of the *Simple-miRNA model*, '*simple*' because now the

production of protein that was in the *Simple-miRNA model* will not be considered and '*double*' because miRNA has two targets. However, some more reactions such as the production and degradation of miRNA will be taken into account, and also the association and dissociation of the complex regulators of mRNA-miRNA. This is required to study interdependency between the targets via a common regulator, miRNA in this case, that controls both of them.

The *Two mRNA targets Model* from Figure 2.5 as we said before consid-

Figure 2.5. Two mRNA targets Model



MiRNA has two mRNA targets. This model considers miRNA production and degradation with the respective rate constants p_m, δ_m. Because there are two targets all the following reactions are double with the associated rates: transcription of mRNA (rate constants: q_1, q_2), mRNA decay (δ_1, δ_2), association and dissociation of mRNA-miRNA complex (denoted by m_1^*, m_2^* the complexes and with rate constants: $\beta_1, \beta_2, \beta_1^-, \beta_2^-$)), degradation of the mRNA-miRNA complexes (δ_1^*, δ_2^*), and the reaction from the mRNA-miRNA complex that miRNA returns to the pool ($\delta_1^* q, \delta_2^* q$).

ers production and degradation of miRNA with the respective rate constants

p_m, δ_m. Because miRNA has two targets, $mRNA_1$ and $mRNA_2$, two transcription rates are allocated for each production of mRNA with their respective rates q_1, q_2, and two mRNA decays for each target (with rate δ_1, δ_2). In this model the inclusion of miRNA binding to target mRNA, becoming a *mRNA-miRNA* complex, denoted by m_1^* and m_2^* in the figure 2.5 are also taken into account. Their rate constants for the association of these complexes are β_1, β_2, the rates for the dissociations are β_1^-, β_2^-, and the rates for their respective degradations are δ_1^*, δ_2^*. MiRNA exerts its downregulating effect on the targets by accelerating the degradation rate of the complexes having $\delta_1^*/\delta_1 > 1, \delta_2^*/\delta_2 > 1$ (Khanin & Higham 2009). In addition, miRNA returns to the cytoplasmic pool with the constant rate $q\ (\delta_1^* m_1^* + \delta_2^* m_2^*)$, where $q \leq 1$ is the miRNA turnover rate (Levine et al. 2007).

In the previous miRNA models the level of miRNA m was assumed to be either present (miRNA=1) or absent (miRNA=0). This is a step to make the model more realistic, and it considers positive real numbers instead of just two levels, for the values of miRNA molecules.

Stochastic Reactions for a *2 mRNA targets Model*

This model splits up in six first-order and second-order reactions. The reactions are described as follows:

$$
\begin{array}{lll}
1) & \varnothing \xrightarrow{q_1} M_1 & \left.\begin{array}{l}\\ \\ \end{array}\right\} \text{Production from a source} \\
2) & \varnothing \xrightarrow{q_2} M_2 & \\
3) & M_1 \xrightarrow{\delta_1\langle M_1\rangle} \varnothing & \left.\begin{array}{l}\\ \\ \end{array}\right\} \text{Degradation of mRNA} \\
4) & M_2 \xrightarrow{\delta_2\langle M_2\rangle} \varnothing & \\
3) & M_1 + miRNA \xrightarrow{\beta_1\langle M_1\rangle\langle miRNA\rangle} m_1^* & \left.\begin{array}{l}\\ \\ \end{array}\right\} \text{Formation of complex} \\
4) & M_2 + miRNA \xrightarrow{\beta_2\langle M_2\rangle\langle miRNA\rangle} m_2^* & \\
5) & m_1^* \xrightarrow{\beta_1^-\langle m_1^*\rangle} M_1 + miRNA & \left.\begin{array}{l}\\ \\ \end{array}\right\} \text{Dissociation of complex} \\
6) & m_2^* \xrightarrow{\beta_1^-\langle m_2^*\rangle} M_2 + miRNA & \\
7) & m_1^* \xrightarrow{\delta_1^*\langle m_1^*\rangle} \varnothing & \left.\begin{array}{l}\\ \\ \end{array}\right\} \text{Degradation of complex} \\
8) & m_2^* \xrightarrow{\delta_1^*\langle m_2^*\rangle} \varnothing & \\
9) & m_1^* \xrightarrow{\delta_1^* q\langle m_1^*\rangle} miRNA & \left.\begin{array}{l}\\ \\ \end{array}\right\} \text{MicroRNA turnover from complex} \\
10) & m_2^* \xrightarrow{\delta_1^* q\langle m_2^*\rangle} miRNA & \\
11) & \varnothing \xrightarrow{p_m} miRNA & \left.\begin{array}{l}\\ \end{array}\right\} \text{Production from a source} \\
12) & miRNA \xrightarrow{\delta_m} \varnothing & \left.\begin{array}{l}\\ \end{array}\right\} \text{Degradation of miRNA}
\end{array}
\tag{2.11}
$$

This large system represents the *Two mRNA targets Model* where the variables M_1, M_2 encode the first mRNA and the second respectively; all the other ones, plus the constants, have already been explained in the introduction of this section.

2.2.4 N mRNA targets Model

This model describes miRNA-mediated repression of many targets, and it is introduced under the name of a *Multi-step model of microRNA-mediated target regulation* in the recent work of Khanin & Higham (2009). In fact, the description for this model can be given taking the explanation for the *Two mRNA targets Model* but generalizing to the N case. This means that for example two reactions for production of each target, now will become N reactions. Similarly, for the number of reactions of mRNA decays, miRNA complex association and dissociation, and all the other reactions that take place in the models of two targets.

This generalization to the N case includes many more similar type of reactions, parameters and variables than before, increasing the degree of complexity and the difficulty for any study of more than 2 targets. In addition, substantial experimental data is required for a realistic simulation of N targets simple. Rate constants for each type of reaction should be sampled from empirical distribution estimated from experimental observations. This data is just becoming available. For this reason, the N targets model is only introduced in this thesis. It is shown that given sufficient experimental data one can easily extend the two targets model to the N targets case.

In particular, for a system of N targets the number of variables is $2N + 1$ given by N mRNA targets $(m_1, \ldots, m_N)$, N mRNA-miRNA complexes $(m_1^*, \ldots, m_N^*)$, and miRNA. The number of reactions that take place is $6N + 2$. To see more easily from where this number appears we decompose it in $2N + 2N + N + N + 2$, where the first *2N* comes from the production and decay of the targets, the next *2N* from the bindings and unbindings of miRNA-mRNA to form and dissociate the complexes, the next N for the complex decays, N more of miRNA returning to the pool, and the last *2* from the production and miRNA decay. All these reactions and variables will give a high computational cost for any simulation

carried on.

The *N mRNA targets Model* is closer to the idea of how the biological systems works. Modelling a realistic system, that includes a large amount of targets, one can study the effect of miRNA over the different targets, how many miRNA molecules bind mRNA and how many do not, whether groups of genes can be classified depending on these bindings, whether and how the miRNA levels decrease, how this is affected globally by the noise produced, and lots of other things that escape from the work presented in this thesis. But this can be a very interesting future work, with all the introductory explanations given here.

Chapter 3

The Methodology

3.1 Introduction & Aims

One accurate way to study the biological problem, and take into account all the stochastic reactions and changes that the biological system experiences over time, it is tracking each of the molecules of the different reactants over time. In this study we use both Numerical Simulations and Analytical Analysis, and we compare the results.

Principally for the Numerical Analysis, three stochastic simulations have been used: Stochastic Simulation Algorithm (or SSA for short) as described in the work by Higham (2007), simulating the system with randomly different reactions, one by one at a time, in the process of gene expression, and for different times; Langevin Equation, as used in Gillespie (2000), Khanin & Higham (2008), Higham & Khanin (2008), which replaces the massive ODE system used in the Chemical Master Equation (Higham 2007) with a small stochastic differential equation system that is more amenable to computation; and the Tau-Leaping Method (Gillespie 2001) that approximately advances the process by a pre-selected time τ, which may encompass more than one reaction event. Finally, an example of dimerization is simulated by the three methods to compare their results.

Two studies were carried out analytically. Both of these methods use the Chemical Master Equation (Higham 2007) or CME for short, which gives the probability of change from a state n (where the number of molecules for each reactant, each random variable, are specified for these state n), over time. The first method uses the Generating Function Approach to find the means and the variances for the variables in the study. The second method, for systems where the reactions are only unimolecular uses a general formula given in the recent study of (Gadgil et al. 2005), to easily calculate the moments of the Master Equation. Examples for a Simple model and a General model are also studied in the Analytical section and the two methods are compared.

The aim of this chapter is to accurately describe the different methods carried out for the study of the different models.

3.2 Numerical Analysis

Let X be a chemical reacting system consisting of N molecular species $S_1, \ldots, S_N$, and with M chemical reaction channels $R_1, \ldots, R_N$. Let $X_i(t)$ denote the number of molecules of the species S_i at time t, that describes the state of the system at time t. Let $I=1,\ldots, N$ and $J=1,\ldots, M$ be respectively, the sets of indexes over the species and the reactions channels sets. Our goal is to estimate the state vector $\boldsymbol{X}(t) \equiv (X_1(t), \ldots, X_N(t))$, given that the system was in state $\boldsymbol{X}(t_0) = \mathbf{x}_0$ at some initial time t_0.

Each reaction channel R_j represents an instantaneous physical event that changes the population of at least one reactant specie, and is characterized by its state change vector $\nu_j \equiv \nu_{1j}, \ldots, \nu_{Nj}$ and by its propensity function a_j; where ν_{ij} is called the stoichiometric matrix that denotes the change in X_i caused by one R_j event and $a_j(\mathbf{x})dt$, given the system is in state $\boldsymbol{X}(\mathrm{t}) = \mathbf{x}$, is the probability that a reaction R_j occurs in the infinitessimal interval $[t, t + dt)$.

The propensity function is defined as:

$$a_j(\mathbf{x}) = h_j(\mathbf{x})c_j, \tag{3.1}$$

where $h_j(\mathbf{x})$ is the number of distinct reactant molecules combinations, whose function is calculated differently depending of the type of reaction associated (i.e. first order, second order, or dimerization), and c_j is the rate constant of R_j.

Examples:

1. $S_1 \xrightarrow{c_j} S_2 : \quad \nu_j = (-1, 1, 0, \ldots, 0); \quad a_j(\mathbf{x})dt = (c_j dt) \times x_1 \quad \Rightarrow a_j(\mathbf{x}) = c_j x_1$

2. $S_1 + S_2 \xrightarrow{c_j} 2S_2 : \quad \nu_j = (-1, 1, 0, \ldots, 0); \quad a_j(\mathbf{x})dt = (c_j dt) \times x_1 x_2 \quad \Rightarrow a_j(\mathbf{x}) = c_j x_1 x_2$

3.2.1 Stochastic Simulation Algorithm (SSA)

Background

The Stochastic Simulation Algorithm, also known as the Gillespie Algorithm, has been employed extensively for numerically solving the chemical master equation (Gillespie 1992) and to study the evolution of chemically reacting systems. This method was developed by Gillespie (Gillespie 1976).

The SSA is a discrete and exact procedure. It is 'discrete' in the sense that every reaction is individually simulated, one molecule at a time, implying that $\boldsymbol{X}(\mathrm{t})$ is always represented by a non-negative integer random variable (number of molecules), and 'exact' because it is based on computing realisations of the state vector under the same microphysical hypothesis as the chemical master equation (CME). Thus, using the stochastic simulation algorithm for a system is equivalent to solving the Chemical Master Equation for the system, but the difference is that this algorithm computes single realisations of the state vector rather than an entire probability distribution from the CME, which is commonly impossible to solve for most practical problems.

The aim of this algorithm is to draw two random numbers at each time step,

one denoting the next reaction index and the other to determine the time of the next reaction. But because every reaction event is exactly simulated, this procedure exposes the problem that it is computationally very expensive for large populations of some chemical species and for fast reactions involved in the system.

The Method

We assume the system to be "well-stirred" (spatially uniform) and restricted to some constant volume Ω at a constant temperature.

The SSA simulates the system's trajectories of $\mathbf{X}$(t) by random reactions generating at each step two random numbers, one representing the number of the next reaction, and the second for the time at which the next reaction will appear [Gillespie (2007); Higham (2007)].

To generate $\mathbf{X}$(t), a starting point to develop the method of SSA is the probability function $p(\tau, j|\mathbf{x}, t)d\tau$, which denotes the probability that the next reaction j^{th} occurs over the infinitesimal interval $[t+\tau, t+\tau+d\tau)$, given $\mathbf{X}$(t)=$\mathbf{x}$. It is defined as follows:

$$p(\tau, j|\mathbf{x}, t) = P_0(\tau|x, t)a_j(\mathbf{x})d\tau, \tag{3.2}$$

where $a_j(\mathbf{x})d\tau$ is the probability that the j^{th} reaction takes place in the infinitesimal interval $d\tau$, and $P_0(\tau|x, t)$ is the probability that no reaction happens in $[t, t+\tau)$. Applying the laws of probability, P_0 is defined as

$$P_0(\tau|x, t) = exp(-a_0(\mathbf{x})\tau), \tag{3.3}$$

where

$$a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x}). \tag{3.4}$$

Thus, the initial probability derives to the joint density function for two random variables τ and j, given by

$$p(\tau, j|\mathbf{x}, t) = a_j(\mathbf{x})exp(-a_0(\mathbf{x})\tau). \tag{3.5}$$

It allows us to simulate independently the two random variables, implying that τ, **time until next reaction occurs**, is an exponential random variable with mean (and standard deviation) $1/a_0(\mathbf{x})$, while j, **next reaction index**, is a statistically independent integer random variable with point probabilities $a_j(\mathbf{x})/a_0(\mathbf{x})$. One of the Monte Carlo procedures for generating samples of these two variables is the so-called direct method, and it consists in: draw two random numbers r_1 and r_2 from the uniform distribution in the unit interval, and take

$$\tau = \frac{1}{a_0(\mathbf{x})} ln(\frac{1}{r_1}) \tag{3.6}$$

$$j = \text{the smallest integer satisfying } \sum_{k=1}^{j} (a_k(\mathbf{x}) > r_2 a_0(\mathbf{x}). \tag{3.7}$$

Finally, the jump to the next state in the system can be effectuated:

$$X(t + \tau) = X(t) + \nu_j. \tag{3.8}$$

Summary

The Gillespie's algorithm can be summarized as follows:

1. **Compute $a_j(\mathbf{x})$, $\forall$*j=1, . . . ,M*, and $a_0(\mathbf{x})$**
   Evaluate the M propensity functions $a_j(\mathbf{x})$ and their sum $a_0(\mathbf{x})$ from Eq. 3.4.

2. **Generate r_1 and r_2**
   Generate two numbers from the uniform distribution in the unit-interval.

3. **Compute τ**
   Set the time for the next reaction in Eq. 3.6, using the uniform sampled variable r_1.

4. **Compute j**
   Set the index of the next reaction satisfying Eq. 3.7, using r_2.

5. **Update the system**
   Update the system substituting $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \nu_j$ (equivalent to Eq. 3.8).

In practice, after the last step, we return to the first step to keep on updating the system, or stop in the case that t has passed a specified value related to the number of iterations that have been taken.

3.2.2 Chemical Langevin Equation

Background

Another stochastic method to describe the time-evolution of a chemical reacting system is the approximating Chemical Langevin Equation (CLE). It was introduced to accelerate the Gillespie Algorithm, replacing the massive ODE's system of the CME by stochastic differential equations (SDEs).

The CLE is a continous stochastic process, in the sense that the state-vectors that represent the molecular evolution of the system over time are given by real values in contrast to the non-negative integer values obtained with the SSA. The system X is represented by a set of nonlinear, autonomous SDEs, with one SDE for each chemical species. The solution of the j^{th} equation will represent in number of molecules the amount of the j^{th} specie at time t.

This representation by SDEs of the system reduces the number of components in the system allowing for gain in acceleration. However there is lost of accuracy in the results, because the discrete value variables obtained with the previous method are now given by real numbers.

The Method

In the CLE the chemical reacting system will be represented by $\boldsymbol{X}$(t). Also the amount of species i that the system has at time t is represented by a real-valued random variable, $\boldsymbol{X}_i$(t), in contrast to the non-negative integer variables that represented the SSA.

The Stochastic differential equation that describes the system evolution in CLE takes the form of an Ito's equation:

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} \nu_{ji} a_j(X(t)) + \sum_{j=1}^{M} \nu_{ji} \sqrt{a_j(X(t))} W_j(t), \quad (i = 1, \ldots, N) \quad (3.9)$$

where $W_j(t)$, for $j = 1, \ldots, M$ are independent scalar Brownian motions.

As explained in Gillespie (2000), an equivalent equation to this "white-noise form" of Langevin equation (eq.3.9) is the following expression

$$\begin{aligned} X_i(t+dt) &= X_i(t) + \sum_{j=1}^{M} \nu_{ji} a_j(X(t))dt + \sum_{j=1}^{M} \nu_{ji} \sqrt{a_j(X(t))dt} N_j(t) \\ &= X_i(t) + \sum_{j=1}^{M} \nu_{ji} \left(a_j(X(t))dt + \sqrt{a_j(X(t))dt} N_j(t) \right), (i = 1, \ldots, N) \end{aligned}$$

which is precisely a discretization of the continuous time problem (from eq.3.9). Where dt can be substituted by τ (just for a change on notation) and $N_j(t)$ is the "unit normal" random variable $N_j(0, 1)$. The τ is an arbitrary positive value that satisfies the following two conditions (Gillespie 2000):

- *Condition 1:* τ will be *small* enough that no propensity function $a_j(\mathbf{x})$ suffers an "appreciable" change in its value.
- *Condition 2:* τ will be *large* enough that the expected number of reactions $a_j(\mathbf{x})\tau$ for each type R_j firing in the interval $[t + t + \tau]$ is larger than 1.

Obtaining then the new state of the system:

$$X_i(t+\tau) = X_i(t) + \sum_{j=1}^{M} \nu_{ji} \left(a_j(X(t))\tau + \sqrt{a_j(X(t))\tau} N_j(t) \right), \quad (3.10)$$
$$(i = 1, \ldots, N).$$

In fact, the above equation is similar to the one obtained by the method of *τ-leaping* but with k_j, number of R_j reactions that occur in a duration τ if its propensity function remained constant at the value $a_j(x(t))$, being a normal random variable instead of a Poisson. So we are saying that

$$k_j = a_j(X(t))\tau + \sqrt{a_j(X(t))\tau} N_j(t), \quad (i = 1, \ldots, N) \quad (3.11)$$

is a variable from a Normal distribution with mean and variance $a_j(X(t))\tau$, when for the method *τ-leaping* this variable is drawn from Poisson distribution $P_j(a_j(x(t)), \tau)$. This is because different assumptions motivate these methods – for *τ-leaping* each $a_j(x(t))$ will have a relatively small change over $[t, t+\tau)$, when for *Langevin equation* the following assumption holds each $a_j(x(t))\tau$ is large. From the probability theory it is known that a Poisson random variable with large mean is well approximated by a normal random variable with equal mean and variance.

Finally the state-vector can be updated by the simplified expression of

$$X_i(t+\tau) = X_i(t) + \sum_{j=1}^{M} \nu_{ji} k_j \quad (i = 1, \ldots, N). \quad (3.12)$$

Summary

The steps to carry on with the method of CLE are summarised as follows:

1. **Compute $a_j(\mathbf{x})$, $\forall j=1, \ldots, M$, and $a_0(\mathbf{x})$**
   Evaluate the M propensity functions $a_j(\mathbf{x})$ and their sum $a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x})$.

2. **Compute** r_j, $\forall j=1, \ldots, M$
   Sample N random normal numbers from $N_j(0,1)$.

3. **Compute** $K_j(x(t), \tau)$
   Compute the N random variables K_j from Eq. 3.11 using the sample of the previous step.

4. **Update the system**
   The system is updated by incrementing the time by τ and evaluating the new state X(t + τ) with Eq. 3.12.

This algorithm repeats itself from step *1* to *4* while t does not overpass the pre-determined number of steps fixed by the user.

3.2.3 Tau-Leaping Method

Background

The τ-*leaping* method (or TL for short) was introduced by Gillespie (2001) to describe approximately the stochastic time evolution of chemical reacting systems. It was introduced in order to speed up the Gillespie's SSA, with the basic idea of advancing the system by a *pre-selected* time τ during which many reactions occur.

The computer times required to simulate numerically chemical reacting system over time with exact procedures such as SSA tend to be prohibitively long if the molecular populations of at least some of the reactant species are very large and/or some of the reactions are very fast.

The τ-*leaping* method chooses the time increment of the simulation steps in a manner that the propensity functions do not suffer any appreciable change in its value, during the entire interval $[t, t+\tau]$; this is called the *Leap Condition*. It determines how many times each reaction channel fires in this time and leaps from one subinterval to the next, instead of stepping along from one to other reaction. This approximation, as well as the Chemical Langevin Equation, can produce acceptable losses in accuracy but produces significant gains in simulation speed at the same time.

The Method

To set up the $\mathbf{X}(t) = \mathbf{x}$ states of the system over time, the intervals of time through which this method leaps over are chosen first. Then the number of different reactions k_j, for all *j=1,..,M* (where M is the number of reactions types) that will occur within the intervals are computed.

The choice of the interval consists of choosing a τ, for the interval $[t, t+\tau]$, that is neither 'too big' nor 'too small'. By not 'too big' this means that τ satisfies the *Leap Condition* (Gillespie (2001), page 1719), and 'too small' refers to duration of τ, that at least, many reaction events occur (Gillespie 2001); see also Higham (2007). As there are at least two and usually many reactions in a time interval, the TL has significant gains in acceleration over the SSA method, which moves one time step along reaction by reaction. If the acceleration is not significant – this is the case when TL does not include more than one reaction firing in the interval – and taking into account that TL is an approximate procedure, then it will be better to use the exact procedure of SSA, even if it fires only one reaction per τ.

This last condition, about the small size of τ, can be controlled by low bounding τ by $2/a_0(\mathbf{x})$ (Gillespie (2001), page 1721). Since the expected time to the next reaction in the SSA is $1/a_0(\mathbf{x})$, then if

$$\tau \leq \frac{2}{a_0(\mathbf{x})}, \tag{3.13}$$

where the numerator could arguably be replaced by anything between 1 and 10, it will be inefficient to use τ-*leaping*. Thus, the τ selected will be supplemented by the *tau-leaping* method for the exact SSA instead.

For each different state of the system $\mathbf{X}(t)$, τ will be chosen so that satisfies the *leap Condition* and the bounding by $2/a_0(\mathbf{x})$. Once τ is calculated, the number of reactions of each type R_j will be k_j, and approximated by $P_j(a_j(\mathbf{x})\tau)$, a statistically independent Poisson random variable with mean $a_j(\mathbf{x})\tau$ (where $a_j(\mathbf{x})\tau$ is interpreted as the probability that one R_j event will occur in the next τ time units).

Equations for TL following the method of Gillespie (2001)

Thus, the expression of the number of firings for each type of reaction R_j is given by

$$k_j = P_j(a_j(\mathbf{x})\tau), \quad \forall j=1, \ldots, M. \tag{3.14}$$

The next state of the system will be determined by λ, and replaced by

$$X(t+\tau) \rightarrow X(t) + \lambda,$$

where the expected net change in state in $[t,t+\tau)$ will be

$$\bar{\lambda} \equiv \bar{\lambda}(\mathbf{x}, \tau) = \Sigma_{j=1}^{M}[a_j(\mathbf{x})\tau]\nu_j = \tau\xi(\mathbf{x}), \tag{3.15}$$

thus $\xi(\mathbf{x})$ is defined as

$$\xi(x) \equiv \Sigma_{j=1}^{M}[a_j(\mathbf{x})\nu_j], \tag{3.16}$$

$\xi(\mathbf{x})$ can be interpreted as the mean or expected state change in a unit of time.

Once λ is computed, the problem of ensuring that λ will not exhaust any of the reactants driving them to a negative population appears. Different resolutions of this problem have been studied and published, one of them can be found in Cao et al. (2005), which introduces a control parameter that classifies as "critical" any reaction R_j that is in danger of exhausting any of its reactants. Besides these resolutions, a simple encoding version in Matlab (numerical computing environment and programming language) is shown in this project, and it consists in recalculating λ until the obtained value does not exhaust the reactant. Thus, λ satysfies:

$$\lambda(i) + \mathbf{X}(i) \geq 0, \quad \forall j=1, \ldots, N. \tag{3.17}$$

If this condition is not satisfied, λ and τ are set up to zero in the manner that there is no leap and neither a new state for the system. Everything is recalculated

again from the beginning.

Finally, the largest value of τ that is consistent with the Leap condition, and hence the optimal choice for τ given the value chosen for ϵ ($0 < \epsilon < 1$), is

$$\tau = Min_{j\epsilon[1,M]}\{\epsilon a_0(\mathbf{x})/|\sum_{i=1}^{N}\xi_i(\mathbf{x})b_{ji}(\mathbf{x})|\} \tag{3.18}$$

where $b_{ji}(\mathbf{x})$ and $a_0(\mathbf{x})$ are

$$b_{ji}(\mathbf{x}) = \frac{\partial a_j(\mathbf{x})}{\partial x_i} \qquad (j=1, \ldots,M;\ i=1,\ldots,N) \tag{3.19}$$

$$a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x}). \tag{3.20}$$

So finally the leap condition is executed, and with τ and k_j the new values of time and state are calculated updating the system as follows:

$$\begin{array}{rcl} t & \rightarrow & t+\tau \\ \mathbf{x} & \rightarrow & \mathbf{x}+\lambda. \end{array} \tag{3.21}$$

Summary

Overall, the steps of the tau-leaping method can be summarized as follows:

1. **Compute τ**

   (a) **Compute $a_j(\mathbf{x})$, $\forall$*j=1, . . . ,M*, and $a_0(\mathbf{x})$**
   Evaluate the M propensity functions $a_j(\mathbf{x})$ and their sum $a_0(\mathbf{x})$ (sum in Eq. 3.20)

   (b) **Compute $b_{ji}(\mathbf{x})$, $\forall$*j=1, . . . ,M*, $\forall$*i=1, . . . ,N***
   Compute the MN partial derivatives b_{ji} in Eq. 3.19

   (c) **Compute $\xi(\mathbf{x})$ and find τ**
   Compute the N dimension vector ξ from Eq. 3.16 to find the smallest

of the M ratios from Eq. 3.18

(d) **Evaluate τ boundary condition**
If τ satisfies Eq. 3.13 then reject it and execute instead SSA. Else if τ is larger than $2/a_0(\mathbf{x})$, then accept it and proceed to next step.

2. **Sample k_j and compute λ, $\forall j=1, \ldots, M$**
Generate the M sample value k_j of the Poisson random variable from Eq. 3.14 and compute the N dimension vector λ in Eq. 3.15.
If λ(i) does not satisfy Eq. 3.17 move backwards and come back to *step 1*, but if it satisfies the condition proceed to next step.

3. **Effectuate leap condition**.
Effectuate the leap condition in Eq. 3.21

The system can keep being updated by the leap condition repeating this algorithm several times until the number of steps of the iteration, fixed by the user, is overtaken.

3.2.4 Example of Dimerization: Comparison within Methods

The purpose of this section is to simulate an example of dimerization, already studied in the work of Khanin & Higham (2008) but with a slightly different objective here, which is to show that for three different numerical techniques – the methods of SSA, Langevin and Tau-leaping – the means and theirs confidence intervals (CIs) match.

Dimerization is the process where proteins produced from mRNA combine to form complexes. This process can be complex so to simplify things, the model is reduced to protein that are produced from a source and possible reversibility of dimerization is ignored.

The stochastic reactions that describe this simple dimer model are the following:

$$1) \quad \varnothing \xrightarrow{k_1} P$$

$$2) \quad P + P \xrightarrow{k_a} P_2$$

$$3) \quad P \xrightarrow{\gamma_p} \varnothing$$

$$4) \quad P_2 \xrightarrow{\gamma_{p_2}} \varnothing$$

where P encodes the protein monomer and P_2 the protein dimer, the respective propensity functions for each reaction are:

$$\begin{aligned} a_1 &= k_1 \\ a_2 &= k_a \\ a_3 &= \gamma_p P \\ a_4 &= \gamma_{p_2} P_2 \end{aligned}$$

The aim is to test whether the results match, or are approximate enough, when comparing between the three different numerical methods: SSA, CLE and Tau-Leaping frameworks. The implemented code for the Tau-Leaping in Matlab (for this model) introduced in this project is described in *Appendix* A.

Proceeding computationally the constants and initial values of P and P_2 are also taken from the example given in Khanin & Higham (2008). The rates are $k_1 = 5$, $k_a = 0.01$, $\gamma_p = 0.1$ and $\gamma_{p_2} = 0.01$, with the molecular initial data of P(0)=10 and $P_2(0) = 2$, and time interval of $0 \leq t \leq 20$. Computing the simulations by the three methods over $K = 10^4$ paths, we computed the sample means approximations and confidence intervals for P and P_2. The step size used in the CLE is $20/500 = 0.04$, and the ε used in Tau-Leaping is 0.04 because it is the value recommended in Gillespie (2007). The results are given in Table 3.1, which contains the 95% confidence intervals for each sample mean.

Table 3.1. 95% confidence intervals for the monomers and dimers by the different methods

	P	P_2	time
SSA	[17.8, 18.0]	[27.8, 28.1]	18.1sec
CLE	[17.8, 18.0]	[27.9, 28.1]	13.8sec
TAU	[18.0, 18.2]	[24.1, 24.4]	2.47min

The results of the 95% CIs of the means of P and P_2 at time t=20 are given by the methods of SSA, CLE and Tau-Leaping. The CLE uses stepsize $20/500 = 0.04$ and Tau-Leaping uses ε of 0.04.

From Table 3.1, we see that the SSA and CLE confidence intervals overlap for both monomer and dimer protein. Meanwhile with Tau-leaping they only overlap for the monomer. The explanation of the bias error given by the results of P_2 for Tau-leaping with the other two methods depends partly on the accuracy parameter of ε taken. Where if the error with which the user sets up the tau-leaping for the simulation is getting smaller (closer to zero) the approximation will be also closer to the results obtained by SSA. However, the fact that this limit is not exactly the same is not relevant, because the gain in time when the simulation of tau-leaping is applied is worthwhile when it allows reasonably large step sizes to be used. Even though in this example, the dimer model with the chosen initial number of molecules and other parameters does not accelerate the method of SSA.

Table 3.2 shows the accuracy of the Tau-leaping method for different values of ε for the same variables P and P_2. This is just a sample of how the variation of the accuracy error can change the different values of the results. The relevant thing is that in order to gain more accuracy (i.e. smaller error) the simulation becomes computationally expensive, but because one of the objectives of using Tau-leaping is to accelerate the SSA, practically some lose accuracy is allowed for in order to be able to spend less time in the simulation.

Table 3.2. 95% confidence intervals for the monomers and dimers by different ε using Tau-leaping

ε	0.9	0.5	0.09	0.05	0.009	0.005
P	[31.3 31.9]	[15.1, 15.4]	[17.9, 18.1]	[18.0, 18.2]	[17.9, 18.0]	[17.8, 18.0]
P_2	[27.4, 27.8]	[23.4, 23.7]	[20.3, 20.7]	[22.0, 22.4]	[27.2, 27.4]	[27.7, 27.9]

95% CIs of the means of P and P_2 at time t=20 for several values of ε using Tau-Leaping. And when the ε is getting smaller the approximation to the results of SSA (Table 3.1) the results are more similar and accurate.

Table 3.2 shows that the first two ε the results are very different of the results of SSA for the monomer P, even though they seem more accurate for the dimer. Thus it can be concluded that for this example to have some accurate results, the bounds of ε will have to be between *0* and *0.5* instead of between *0* and *1* as it is defined in the section of the Tau-Leaping method.

Finally the conclusions are:

1. The simulations of SSA, CLE and Tau-Leaping match the results closely for a very simple dimer model, with an allowance of accuracy error in Tau-Leaping of 0.04.

2. For the dimer model, the CLE accelerates the SSA, whereas Tau-Leaping is not under the best conditions in terms of number of molecules and/or parameters of the system to accelerate the method of Gillespie.

3. When the error of accuracy for Tau-leaping gets smaller, the limit of the protein levels becomes closer to the results given by SSA.

3.3 Analytical Analysis

In this section two analytical methods are described, the Probability Generating Function and the method developed by Gadgil *et al* (2005). Both methods attempt to solve the Chemical Master Equation (or CME for short (Higham 2007)) by alternative techniques. The Master Equation cannot be solved analytically except for a small number of specific simple systems, that is why other methods have to be applied.

The noise in the stochastic variables of interest q, **noise strength**, is usually measured by either the Fano factor or the coefficient of variation. Until recently the standard measure was the coefficient of variation (CV) defined as follows:

$$CV = \delta q / \langle q \rangle \tag{3.22}$$

where δq is standard deviation and $\langle q \rangle$ is the mean. The CV is used as a measure of noise in Elowitz et al. (2002), Swain et al. (2002) and Raser & O'Shea (2004). The second measure, the **Fano factor** ν, is defined as the ratio between the variance and the mean:

$$\nu = \delta q^2 / \langle q \rangle \tag{3.23}$$

introduced by Thattai & van Oudenaarden (2001), and used in Blake et al. (2003) and Ozbudak et al. (2002). As the definition shows, the Fano factor of an arbitrary stochastic system is related to the standard deviation and reveals deviations from Poissonian behavior. This measure of noise is sensitive and will be used principally in all the studies in this project.

It must be noted however that the use of different measures of noise may lead to different conclusions concerning the importance of noise in underlying process (Swain et al. 2002).

3.3.1 Probability Generating Function

Background

The Generating Function Approach (Takasu 2005) is used in this thesis to analyse the CME. The idea is to calculate the mean and the variance for all the reactant variables of the system. This will be carried out by deriving the equations for the first and second moments of their distributions. The advantage of the **probability generating function** is that it provides an analytical solution whereas generally the CME does not, thus enabling the calculation of the probability distribution function for the distribution of each reactant in such systems.

The probability generating function is associated with a probability distribution $P_n(t)$, defined as:

$$G(z,t) = \sum_n P_n(t) z^n$$

Some of the properties of this function are given as follows:

1. Substituting $z = 1$ in G:

$$G(z,t)|_{z=1} = \sum_n P_n(t) 1^n = 1. \tag{3.24}$$

2. Differentiating G respect to z, and substituting $z = 1$:

$$\frac{\partial}{\partial z} G(z,t)\Big|_{z=1} = \sum_n n P_n(t) 1^{n-1} = < n > \tag{3.25}$$

   where $< n >$ is the mean of the reactant variable n.

3. Differentiating G respect to z twice, and substituting $z = 1$:

$$\frac{\partial^2}{\partial z^2} G(z,t)\Big|_{z=1} = \sum_n n(n-1) P_n(t) 1^{n-2} = < n^2 > - < n > \tag{3.26}$$

At this point we observe that the property (2), first-order moment of the probability generating function G, gives already the mean of the reactant variable n. While the property (3), the second-order moment of G, gives an expression that is different but very close to the formula for the variance of n (Eq. 3.27), in the sense that contains also terms of $< n^2 >$ and $< n >$ like the variance formula.

So, recalling the formula for the variance:

$$Var[n] \;=\; < n^2 > - < n >^2 \tag{3.27}$$

we observe that property (3) given by Eq. 3.26, second-order moment of G, is quite close to be equal to the expected formula of the variance of the variable n. So, a quadratic function of $< n >$ differ from these two equations 3.27 and 3.26, which is the following:

$$< n > - < n >^2 \;=\; Var[n] - \left(\frac{\partial^2}{\partial z^2} G(z,t) \bigg|_{z=1} \right). \tag{3.28}$$

Therefore, we can rewrite now the formula of the variance just in terms of G:

$$Var[n] \;=\; \frac{\partial^2}{\partial z^2} G(z,t) \bigg|_{z=1} + < n > - < n >^2 \tag{3.29}$$

$$\;=\; \left\{ \frac{\partial^2}{\partial z^2} G(z,t) + \frac{\partial}{\partial z} G(z,t) + \left(\frac{\partial}{\partial z} G(z,t) \right)^2 \right\} \bigg|_{z=1} . \tag{3.30}$$

In fact, this is a result from standard statistical theory that links probability generating function to the moments of the random variable. Thus, the variance of the random variable n has also been expressed in terms of G, as we did previously with the mean.

This shows that if we can obtain the probability generating function, and solve its first and second-order moments, then we can calculate easily the mean and variance of n.

The Method

To obtain the probability generating function, G, we need to write down the CME, that involves all the different reactions (see Gadgil et al. (2005) for explanation about how to build the CME). The probability generating function is called on the second step and it has to be written down in concordance with the number of variables that the system has. The next step involves substitution of the CME in the previous expression. To obtain the term $\frac{dP}{dt}$ (of the CME expression) requires differentiating both sides of the equation with respect to time. After that, the expression have to be manipulated until $\frac{dG}{dt}$ depends only on terms of G, instead of P.

The aim is to study the properties of the system at steady-state that is when the system has no changes over time (or the minimum changes). Therefore, the expression of the variation of the function G over time is set up to *0* and written as follows, $\frac{dG}{dt} = 0$. From this point differentiating with respect to the different variables once and twice the last expression, the first and second moments for all the variables respectively can be found. Finally, using the properties (2) and (3) explained in the section 3.3.1, the means and variances of the different variables of the system are computed, and the Fano factor is obtained.

Summary

The Probability Generating Function method can be summarized as follows:

1. Write down the Master equation.
2. Call G (Generating Function) and differenciate w.r.t time.
3. Substitute the Master Equation in G.
4. Simplify the equation just in terms of G.

5. Put at steady-state $\Rightarrow \frac{\partial G}{\partial t} = 0$.
6. Find the partial derivatives 1^{st} and 2^{nd} order (for all z_i).
7. Find out the 1^{st} and 2^{nd} moments.
8. Set up: $z_i = 1, \forall$ i (where z_i are the species).
9. Give an expression for E$[z_i]$ and Var$[z_i]$.
10. Find the Fano factor.

3.3.2 Gadgil *et al*

Background

A recent paper published in the Bulletin of Mathematical Biology Gadgil et al. (2005) describes the general methodology for biological systems that contain only first-order reactions. Here their method is applied to the study of gene expression (basically in one mRNA target models, because all the reactions are of first-order).

The method of Gadgil et al. (2005) is based on using a general formula to easily calculate the moments of the Master Equation. Thus, this method is more simple than the method of the Generating Function Approach, because it only consists in writing correctly matrices for the rate constants and later doing algebraic operations instead of having to introduce the CME.

Table 3.3 classifies the first-order reactions in four groups corresponding to the following reactions: production from a constant source, degradation, conversion to another species, and production catalyzed by another species. The rate constants associated to each reaction type help to define later the incidence matrices for the different reaction types.

Method

The method is based on applying the formulas for the first and second-order moments matrices given in the paper of Gadgil et al. (2005). This requires defining

Table 3.3. The four classes of first-order reactions considered in the stochastic model

Label	Type of reaction	Reaction	Rate
I	Production from a source	$\varnothing \longrightarrow M_i$	k_i^s
II	Degradation	$M_i \longrightarrow \varnothing$	$k_i^d n_i$
III	Conversion	$M_j \longrightarrow M_i$	$k_{ij}^{con} n_j$
IV	Catalytic production from a source	$\varnothing \xrightarrow{M_j} M_i$	$k_{ij}^{cat} n_j$

This table shows the classification of the first-order reactions (Gadgil et al. 2005) with their associated rate constants.

a priori the matrices of the rate-constants that are also used to compute these formulas. With the first-order moments the means of the reactants are already given (see Eq. 3.25), and with the second-order moments the variances can be easily found, as the computation requires just another additional step, that is explained in the following paragraph.

Four matrices that correspond to different reaction types, one for each type, are given as follows: $K^s = diag\{k_i^s\}$, $K^d = diag\{k_i^d\}$, $K^{cat} = k_{ij}^{cat}$ and K^{con} that is defined as

$$K_{ij}^{con} = \begin{cases} k_{ij}^{con} & if \quad i \neq j \\ \\ -\sum_k{}' k_{ij}^{con} & if \quad i \neq j. \end{cases} \tag{3.31}$$

The first-order moments matrix is computed by the following system of equations:

$$\frac{d}{dt}M(t) = (K^{con} + K^{cat} - K^d)M(t) + K^s\mathbf{1} \tag{3.32}$$

$$= \mathcal{K}M(t) + K^s\mathbf{1}, \tag{3.33}$$

where $M(t) = [E[N_1(t)], \ldots, E[N_S(t)]]^T$ and $\mathcal{K}$ is defined by the second equality.

The matrix containing the second-order moments is:

$$\frac{d}{dt}V(t) = \mathcal{K}V(t) + (\mathcal{K}V(t))^T + \Gamma(t) + \Gamma^T(t), \tag{3.34}$$

where $\Gamma_{ij}(t) \equiv (K^{cat}_{ij} - K^s_{ii})M_j(t)$.

So, the second-order moments are already computed and the last step involves computation the variances, that follow from the second-order moments. This step requires adding the mean and subtracting the squared mean of each reactant to the corresponding second-order moments. Their variances are obtained as explained by Eq. 3.29.

Finally the Fano Factor for each reactant is calculated by dividing the variance by the mean.

Summary

The method of Gadgil et al. (2005) can be summarized as follows:

1. Set up the rate matrices: K^s, K^d, K^{cat} and K^{con}.
2. Compute $\mathcal{K}$ (where $\mathcal{K} = K^{con} + K^{cat} - K^d$, from Eq. 3.32 and 3.33).
3. Find out the means of the different reactants with the first-order moments matrix (Eq. 3.33).
4. Find out the second-order moments matrix (Eq. 3.34).
5. Compute the variances from Eq. 3.29.
6. Find the Fano factor.

3.3.3 Example of General Model: Gadgil *et al*

The General model can be separated in six first-order reactions, given by the systems 2.1 and 2.3 in *Chapter* 2. There are four species of interest: DNA,

DNA*, mRNA and protein (encoded by D, D^*, M and P, respectively). To apply Gadgil *et al*, first of all, we define the vector of the means, *M(t)*, that will be:

$$M(t) = [\mathbb{E}[D], \mathbb{E}[D^*], \mathbb{E}[M], \mathbb{E}[P]]$$

.

Furthermore, because we have four species (D, D^*, M and P) the matrices of the rate constants have dimensions $4x4$:

$$K^s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \qquad K^d = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma r & 0 \\ 0 & 0 & 0 & \gamma p \end{pmatrix};$$

$$K^{con} = \begin{pmatrix} -k_a & k_b & 0 & 0 \\ k_a & -k_b & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \qquad K^{cat} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & k_r & 0 & 0 \\ 0 & 0 & k_p & 0 \end{pmatrix}.$$

Then, $\mathcal{K}$ is defined as $\mathcal{K} = (K^{con} + K^{cat} - K^d)$:

$$\mathcal{K} = \begin{pmatrix} -k_a & k_b & 0 & 0 \\ k_a & -k_b & 0 & 0 \\ 0 & k_r & -\gamma r & 0 \\ 0 & 0 & k_p & -\gamma p \end{pmatrix}.$$

Thus, the first-order moment matrix is calculated by Eq. 3.33,

$$\begin{pmatrix} \frac{d}{dt}E[D] \\ \frac{d}{dt}E[D^*] \\ \frac{d}{dt}E[M] \\ \frac{d}{dt}E[P] \end{pmatrix} = \begin{pmatrix} -k_a & k_b & 0 & 0 \\ k_a & -k_b & 0 & 0 \\ 0 & k_r & -\gamma r & 0 \\ 0 & 0 & k_p & -\gamma p \end{pmatrix} \cdot \begin{pmatrix} E[D] \\ E[D^*] \\ E[M] \\ E[P] \end{pmatrix}$$

Set the system at steady-state by putting $\left(\frac{d}{dt}M(t) = \vec{0}\right)$:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -k_a & k_b & 0 & 0 \\ k_a & -k_b & 0 & 0 \\ 0 & k_r & -\gamma r & 0 \\ 0 & 0 & k_p & -\gamma p \end{pmatrix} \cdot \begin{pmatrix} E[D] \\ E[D^*] \\ E[M] \\ E[P] \end{pmatrix} \Rightarrow \begin{cases} E[D] &= \frac{k_d}{k_a} E[D^*] \\ E[D^*] &= \frac{k_a}{k_d} E[D] \\ E[M] &= \frac{k_r}{\gamma_r} E[D^*] \\ E[P] &= \frac{k_p}{\gamma_p} E[M] \end{cases}$$

The first and second equations, in the above system, are linearly dependent (i.e. they give the same results). Thus, removing the first equation and using Eq. 2.5 (from *Chapter* 2), the system of solutions is reduced to:

$$\begin{cases} E[D^*] &= \frac{k_a D_T}{k_a + k_d} \\ E[M] &= \frac{k_r}{\gamma_r} E[D^*] \\ E[P] &= \frac{k_p}{\gamma_p} E[M] \end{cases} \tag{3.35}$$

This system represents the solutions of the first-order moments (reactant means), where D_T is the total number of gene copies.

The next step is finding the second-order moments by the following formula (Eq. 3.34):
$\frac{d}{dt}V(t) = \mathcal{K}V(t) + (\mathcal{K}V(t))^T + \Gamma(t) + \Gamma^T(t)$, where $\Gamma_{ij}(t) \equiv (K_{ij}^{cat} - K_{ii}^{s})M_j(t)$.

First of all, the second-order moment matrix with the correlations is defined

is as follows,

$$V(t) := \begin{pmatrix} E[D^2 - D] & E[DD^*] & E[DM] & E[DP] \\ E[D^*D] & E[D^{*2} - D^*] & E[D^*M] & E[D^*P] \\ E[MD] & E[MD^*] & E[M^2 - M] & E[MP] \\ E[PD] & E[PD^*] & E[PM] & E[P^2 - P] \end{pmatrix} \quad (3.36)$$

It must be noted, however, that this matrix is symmetric. For example: $E[DD^*] = E[D^*D]$ or $E[DM] = E[MD]$. Thus, to make the notation simpler, the matrix $V(t)$ is rewritten as follows,

$$V(t) = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix} \text{ and } \xrightarrow{\text{by symmetry}} \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{12} & v_{22} & v_{23} & v_{24} \\ v_{13} & v_{23} & v_{33} & v_{34} \\ v_{14} & v_{24} & v_{34} & v_{44} \end{pmatrix} \quad (3.37)$$

The term $\mathcal{K}V(t)$ results in:

$$\mathcal{K}V(t) = \begin{pmatrix} -k_a v_{11} + k_d v_{12} & -k_a v_{12} + k_d v_{22} & -k_a v_{13} + k_d v_{23} & -k_a v_{14} + k_d v_{24} \\ k_a v_{11} - k_d v_{12} & k_a v_{12} + k_d v_{22} & k_a v_{13} - k_d v_{23} & k_a v_{14} + k_d v_{24} \\ k_r v_{21} - \gamma_r v_{13} & k_r v_{22} - \gamma_r v_{23} & k_r v_{23} - \gamma_r v_{33} & k_r v_{24} - \gamma_r v_{34} \\ k_p v_{13} - \gamma_p v_{14} & k_p v_{23} - \gamma_p v_{24} & k_p v_{33} - k_p v_{34} & k_p v_{34} - \gamma_p v_{44} \end{pmatrix}$$

The matrix $\Gamma_{ij}(t)$ is defined as,

$$\Gamma_{ij}(t) = (K_{ij}^{cat} - K_{ii}^{s}) M_j(t) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & k_r E[D^*] & 0 & 0 \\ 0 & 0 & k_p E[M] & 0 \end{pmatrix} \quad (3.38)$$

$$\Longrightarrow \quad \Gamma^t_{ij}(t) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & k_r E[D^*] & 0 \\ 0 & 0 & 0 & k_p E[M] \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.39}$$

Now, $\frac{d}{dt}V(t)$ can be computed (by Eq. 3.34). To simplify the next expression we use the notation $a_{ij} = \frac{d}{dt}v_{ij}(t)$, and the result is the following,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ \vdots & a_{22} & a_{23} & a_{24} \\ \vdots & \ddots & a_{33} & a_{34} \\ \dots & \dots & \dots & a_{44} \end{pmatrix} \Rightarrow \begin{cases} a_{11} &= -2k_a v_{11} + 2k_d v_{12} \\ a_{12} &= -k_a v_{12} + k_d v_{22} + k_a v_{11} - k_d v_{12} \\ a_{13} &= -k_a v_{13} + k_d v_{23} + k_r v_{21} - \gamma_r v_{13} \\ a_{14} &= -k_a v_{14} + k_d v_{24} + k_p v_{13} - \gamma_p v_{14} \\ a_{22} &= 2k_a v_{12} + 2k_d v_{22} \\ a_{23} &= k_a v_{13} - k_d v_{23} + k_r v_{22} - \gamma_r v_{23} + k_r E[D^*] \\ a_{24} &= k_a v_{14} + k_d v_{24} + k_p v_{23} - \gamma_p v_{24} \\ a_{33} &= 2k_r v_{23} - 2\gamma_r v_{33} \\ a_{34} &= k_r v_{24} - \gamma_r v_{34} + k_p v_{33} - k_p v_{34} + k_p E[M] \\ a_{44} &= 2k_p v_{34} - 2\gamma_p v_{44} \end{cases}$$

The matrix is completed by dots because it is symmetric: the lower triangle of the matrix is the symmetrical to the upper triangle.

Here, to find out the solutions of the second-order moment matrix, we put the system at the steady-state ($\frac{d}{dt}V(t) = \vec{0}$):

$$\begin{cases} 0 = -2k_a v_{11} + 2k_d v_{12} & (\Rightarrow \frac{d}{dt} v_{11} = 0) \\ 0 = -k_a v_{12} + k_d v_{22} + k_a v_{11} - k_d v_{12} & (\Rightarrow \frac{d}{dt} v_{12} = 0) \\ 0 = -k_a v_{13} + k_d v_{23} + k_r v_{21} - \gamma_r v_{13} & (\Rightarrow \frac{d}{dt} v_{13} = 0) \\ 0 = -k_a v_{14} + k_d v_{24} + k_p v_{13} - \gamma_p v_{14} & (\Rightarrow \frac{d}{dt} v_{14} = 0) \\ 0 = 2k_a v_{12} + 2k_d v_{22} & (\Rightarrow \frac{d}{dt} v_{22} = 0) \\ 0 = k_a v_{13} - k_d v_{23} + k_r v_{22} - \gamma_r v_{23} + k_r E[D^*] & (\Rightarrow \frac{d}{dt} v_{23} = 0) \\ 0 = k_a v_{14} + k_d v_{24} + k_p v_{23} - \gamma_p v_{24} & (\Rightarrow \frac{d}{dt} v_{24} = 0) \\ 0 = 2k_r v_{23} - 2\gamma_r v_{33} & (\Rightarrow \frac{d}{dt} v_{33} = 0) \\ 0 = k_r v_{24} - \gamma_r v_{34} + k_p v_{33} - k_p v_{34} + k_p E[M] & (\Rightarrow \frac{d}{dt} v_{34} = 0) \\ 0 = 2k_p v_{34} - 2\gamma_p v_{44} & (\Rightarrow \frac{d}{dt} v_{44} = 0) \end{cases} \tag{3.40}$$

An attempt to solve the system at this point will result in having all the variables depending on one, because the system is linearly dependent. So, the second equation ($\frac{d}{dt} v_{12} = 0$) that depends on the first and the fifth equations, is removed. Using Eq. 2.5, the variable v_{12} is substituted by the next expression:

$$v_{12} = E[DD^*] = E[(D_T - D^*)D^*] = D_T E[D^*] - E[D^{*2}],$$

where $E[D^{*2}]$ can be found by,

$$\begin{aligned} v_{22} = E[D^{*2} - D^*] = E[D^{*2}] - E[D^*] \\ \Rightarrow E[D^{*2}] = v_{22} - E[D^*]. \end{aligned}$$

Finally, we arrive at

$$v_{12} = (D_T + 2)E[D^*] - v_{22}.$$

In this particular example, $D_T = 1$, reducing the last expression to

$$v_{12} = 3E[D^*] - v_{22}. \tag{3.41}$$

So, by removing the second equation and substituting the new expression for v_{12}, results in the system 3.40 that is solved using the mathematical solver software *Maple 9.5*) yielding the corresponding values for v_{ij} for all $i, j = 1, .., 4$.

If $i \neq j$, v_{ij}'s are the correlations: $E[DD^*]$, $E[DM]$, $E[DP]$, $E[D^*M]$, $E[D^*P]$, $E[MP]$; If $i = j$, v_{ij}'s are defined by the second-order moments:

$$\begin{aligned} v_{11} &= E[D^2 - D] = E[D^2] - E[D] \\ v_{22} &= E[D^{*2} - D^*] = E[D^{*2}] - E[-D^*] \\ v_{33} &= E[M^2 - M] = E[M^2] - E[-M] \\ v_{44} &= E[P^2 - P] = E[P^2] - E[-P] \end{aligned}$$

From these equations, the variances of the reactants are easy to find using Eq. 3.29 and the solutions given in the system 3.67. The results are very large expressions, for some of them, that is why here only the expressions for variance of mRNA (M) and noise of mRNA and protein(P) are expressed.

The variance of mRNA is

$$Var_{\langle M \rangle} = \frac{k_a k_r \left(k_d k_r + k_a^2 + 2k_d k_a + k_d^2 + \gamma_r k_a + k_d \gamma_r\right)}{(\gamma_r (k_a + k_d + \gamma_r)(k_a + k_d)^2)} \tag{3.42}$$

and the Fano factor of mRNA is given by

$$Fano_{\langle M \rangle} = \frac{Var_{\langle M \rangle}}{\langle M \rangle} = 1 + \frac{k_d k_r}{(k_a + k_d)(k_a + k_d + \gamma_r)}. \tag{3.43}$$

The Fano factor of protein is expressed by

$$Fano_{\langle P \rangle} = 1 + \frac{kp}{\gamma_r + \gamma_p} \left(1 + \frac{k_d k_r}{(k_a + k_d)(k_a + k_d + \gamma_r)} (1 + \frac{\gamma_r}{k_a + k_d + \gamma_p})\right) \tag{3.44}$$

3.3.4 Example of Simple Model: Comparison within Methods

Example of Simple Model: Probability Generating Function Approach

The CME for the Simple model (model from Figure 2.1, in *Chapter 2*) is given by the four following reactions:

$$1) \quad \varnothing \xrightarrow{k_r} M \quad : \text{production from a source}$$

$$2) \quad M \xrightarrow{\gamma_r \langle M \rangle} \varnothing \quad : \text{degradation}$$

$$3) \quad M \xrightarrow{k_p \langle M \rangle} P \quad : \text{catalytic reaction}$$

$$4) \quad P \xrightarrow{\gamma_p \langle P \rangle} \varnothing \quad : \text{degradation}$$

where M and P represent mRNA and protein, respectively.

Before writing down the CME, the following notation is introduced: the number of molecules of mRNA $\langle M \rangle$ is denoted by n_1 , and the number of molecules of protein $\langle P \rangle$ is given by n_2. To understand how the construction of the CME works, and how each reaction contributes to the CME, we introduce details in Table 3.4.
Thus, the CME is built by summing the equation terms of each reaction (from Table 3.4), and it is written as:

$$\begin{aligned} \frac{d}{dt}P(n_1,n_2,t) &= k_r\left(P(n_1-1,n_2,t)-P(n_1,n_2,t)\right) \\ &\quad +k_p\left(n_1P(n_1,n_2-1,t)-n_1P(n_1,n_2,t)\right) \\ &\quad +\gamma_r\left((n_1+1)P(n_1+1,n_2,t)-n_1P(n_1,n_2,t)\right) \\ &\quad +\gamma_p\left((n_2+1)P(n_1,n_2+1,t)-n_2P(n_1,n_2,t)\right) \end{aligned} \tag{3.45}$$

Table 3.4. Contributions to the CME for a Simple Model

Reaction	Rate	terms in equation	Reaction type
$\varnothing \xrightarrow{k_r} M$	k_r	$k_r P(n_1-1,n_2,t) - k_r P(n_1,n_2,t)$	production from a source
$M \xrightarrow{\gamma_r n_1} \varnothing$	γ_r	$\gamma_r(n_1+1)P(n_1+1,n_2,t) - \gamma_r n_1 P(n_1,n_2,t)$	degradation
$M \xrightarrow{k_p n_1} P$	k_p	$k_p P(n_1,n_2-1,t) - k_p P(n_1,n_2,t)$	catalytic reaction
$P \xrightarrow{\gamma_p n_2} \varnothing$	γ_p	$\gamma_p(n_2+1)P(n_1,n_2+1,t) - \gamma_p n_2 P(n_1,n_2,t)$	degradation

This table shows the different parts from each reaction that contribute to the CME. Note that $P(n_i,t)$ can be written as $P_{n_i}(t)$, but here we use the first notation for convenience.

Now, to determine the average numbers $\langle n_1 \rangle$ and $\langle n_2 \rangle$ in the steady-state and the variances of n_1 and n_2, we will write the CME in terms of G.

Recall the generating function approach:

$$G(z_1,z_2,t) = \sum_{n_1,n_2} z_1^{n_1} z_2^{n_2} P(n_1,n_2,t) \tag{3.46}$$

Consider first:

$$\frac{\partial G}{\partial z_1}(z_1,z_2,t) = \sum_{n_1,n_2} n_1 z_1^{n_1-1} z_2^{n_2} P(n_1,n_2,t) \tag{3.47}$$

$$\frac{\partial G}{\partial z_2}(z_1,z_2,t) = \sum_{n_1,n_2} n_2 z_1^{n_1} z_2^{n_2-1} P(n_1,n_2,t) \tag{3.48}$$

From equation 3.46, differentiate both sides with respect to time to easily obtain in the right hand side the CME:

$$\frac{\partial}{\partial t}G(z_1,z_2,t)=\sum_{n_1,n_2} z_1^{n_1}z_2^{n_2}\frac{d}{dt}P(n_1,n_2,t) \tag{3.49}$$

Take equation 3.49, and substitute it in the right hand side the CME given by equation 3.45:

$$\begin{aligned}\frac{\partial}{\partial t}G(z_1,z_2,t) &= k_r\left(\sum_{n_1,n_2} z_1^{n_1}z_2^{n_2}P(n_1-1,n_2,t)-\sum_{n_1,n_2} z_1^{n_1}z_2^{n_2}P(n_1,n_2,t)\right)\\ &+ k_p\left(\sum_{n_1,n_2} n_1 z_1^{n_1}z_2^{n_2}P(n_1,n_2-1,t)-\sum_{n_1,n_2} n_1 z_1^{n_1}z_2^{n_2}P(n_1,n_2,t)\right)\\ &+ \gamma_r\left(\sum_{n_1,n_2} (n_1+1) z_1^{n_1}z_2^{n_2}P(n_1+1,n_2,t)-\sum_{n_1,n_2} n_1 z_1^{n_1}z_2^{n_2}P(n_1,n_2,t)\right)\\ &+ \gamma_p\left(\sum_{n_1,n_2} (n_2+1) z_1^{n_1}z_2^{n_2}P(n_1,n_2+1,t)-\sum_{n_1,n_2} n_2 z_1^{n_1}z_2^{n_2}P(n_1,n_2,t)\right)\end{aligned} \tag{3.50}$$

The right hand side of equation (3.50), expressed by four main terms, shows that each of these terms has similarities with the expressions given by (3.47) and (3.48). So, playing around with these equations and trying to give a final expression only on terms of G instead of P, we obtain that the above equation can be expressed as follows:

$$\begin{aligned}\frac{\partial}{\partial t}G(z_1,z_2,t) &= k_r(z_1-1)G(z_1,z_2,t)\\ &\quad +k_p z_2(z_1-1)\frac{\partial G}{\partial z_1}(z_1,z_2,t)\\ &\quad +\gamma_r(1-z_1)\frac{\partial G}{\partial z_1}(z_1,z_2,t)\\ &\quad +\gamma_p(1-z_2)\frac{\partial G}{\partial z_2}(z_1,z_2,t)\end{aligned} \tag{3.51}$$

After that, the system is set up at steady-state implying the following:

$$\frac{\partial}{\partial t}G(z_1, z_2, t) = 0 \tag{3.52}$$

therefore equation 3.51 is rewritten as:

$$\begin{aligned} 0 &= k_r(z_1-1)G(z_1,z_2,t) + k_p z_2(z_1-1)\frac{\partial G}{\partial z_1}(z_1,z_2,t) + \gamma_r(1-z_1)\frac{\partial G}{\partial z_1}(z_1,z_2,t) \\ &\quad +\gamma_p(1-z_2)\frac{\partial G}{\partial z_2}(z_1,z_2,t) \end{aligned} \tag{3.53}$$

The last equation expressed only in terms of G allows us to find the first and second-order moments. So, using properties (2) and (3) from the Probability Generating Function, differentiate equation 3.53 with respect to z_1 once and to z_2, and later differentiate the same equation again with respect to z_1 twice and the same for z_2. The first and second-order moments of G for each reactant (n_1 and n_2) will be obtained.

Finding the first-order moments:

For example, to find out the mean of the mRNA, that is the mean of n_1, property (2) is applied. Again, the notation is changed to make simpler the equations carried on, and we will write just G for what before was $G(z_1, z_2, t)$. So, we just need to differentiate equation 3.53 with respect to z_1, and later substitute z_1 and z_2 equal to 1:

$$\begin{aligned} 0 &= \left(k_r G + k_r(z_1-1)\frac{\partial G}{\partial z_1} + k_p(z_2-1)\frac{\partial G}{\partial z_1} + k_p z_1(z_2-1)\frac{\partial^2 G}{\partial z_1^2} + (-\gamma_r)\frac{\partial G}{\partial z_1} \right. \\ &\quad \left. +(1-z_1)\frac{\partial^2 G}{\partial z_1^2} + \gamma_p(1-z_2)\frac{\partial^2 G}{\partial z_1 \partial z_2} \right)\Bigg|_{z_1=z_2=1} \end{aligned} \tag{3.54}$$

substituting $z_1 = z_1 = 1$, and $\frac{\partial G}{\partial z_1} = \langle n_1 \rangle$ and simplifying:

$$0 = k_r - \gamma_r \langle n_1 \rangle \quad \Rightarrow \quad \langle n_1 \rangle = \frac{k_r}{\gamma_r} \quad \Rightarrow \quad \boxed{\langle M \rangle = \frac{k_r}{\gamma_r}} \tag{3.55}$$

To find out the mean of the protein, or n_2, we apply the property (2), as before. The difference here is that we differentiate with respect to z_2 instead of z_1. So the differentiation with respect to z_2 yields:

$$\begin{aligned} 0 = & \left(k_r(z_1 - 1)\frac{\partial G}{\partial z_2} + k_p z_1 \frac{\partial G}{\partial z_1} + k_p z_1 (z_2 - 1)\frac{\partial^2 G}{\partial z_1^2} + \gamma_r (1 - z_1)\frac{\partial^2 G}{\partial z_1^2} + (-\gamma_p)\frac{\partial G}{\partial z_2} \right. \\ & \left. + \gamma_p (1 - z_2)\frac{\partial^2 G}{\partial z_2^2} \right) \Big|_{z_1 = z_2 = 1} \end{aligned} \tag{3.56}$$

Substituting and simplifying results in:

$$0 = k_p \langle n_1 \rangle - \gamma_p \langle n_2 \rangle \quad \Rightarrow \quad \langle n_2 \rangle = \frac{k_p \langle n_1 \rangle}{\gamma_p} \quad \Rightarrow \quad \boxed{\langle P \rangle = \frac{k_p \langle n_1 \rangle}{\gamma_p}} \tag{3.57}$$

Finally, as the property (2) indicates, the first-order moments of G give the means of the two reactant variables of the system, mean (M) and protein (P).

Finding the second-order moments:

To find out the second-order moments, the property (3) is applied computing partial derivatives of the second-order, and substituting $z_1 = z_2 = 1$. In this subsection all the intermediate steps are not shown as the computation of involves many steps and lengthy formulas calculations. The steps to follow to obtain the results are described as follows:

- Differentiate twice the equation 3.53 with respect to z_1, and later substitute z_1

and z_2 equal to 1. The results are given by:

$$\frac{\partial^2 G}{\partial z_1^2} = \frac{k_r}{\gamma_r}\frac{\partial G}{\partial z_1} = \left(\frac{k_r}{\gamma_r}\right)^2 \tag{3.58}$$

- Differentiate twice the equation 3.53 with respect to z_2, and later substitute z_1 and z_2 equal to 1. Here the solution is more complex, because ends up in terms of $\frac{\partial^2 G}{\partial z_2 \partial z_1}$. And it gives the following:

$$\frac{\partial^2 G}{\partial z_2^2} = \frac{k_p}{\gamma_p}\frac{\partial^2 G}{\partial z_2 \partial z_1}. \tag{3.59}$$

Thus, to find out $\frac{\partial^2 G}{\partial z_2 \partial z_1}$, expression 3.54 has to be differentiated with respect to z_2. The results are as follows:

$$\frac{\partial^2 G}{\partial z_2 \partial z_1} = \frac{\partial^2 G}{\partial z_2^2}\frac{\gamma_p}{k_p} \tag{3.60}$$

and substituting the above result in equation 3.61, to obtain:

$$\frac{\partial^2 G}{\partial z_2^2} = \frac{k_p}{\gamma_p(\gamma_r + \gamma_p)} k_r \left(\frac{k_p}{\gamma_r}\right)\left(\frac{k_r}{\gamma_p} + \frac{k_r}{\gamma_r} + 1\right). \tag{3.61}$$

It has been shown that the second-order moments of G for the two reactant variables of the system, mRNA and protein, have been obtained by property (3). Now, the only property left to compute is the variance of the reactants.

Finding the variance:

Once we have the first and second-order moments for the two variables, we can find

their variances just applying equation 3.30.

The variance for mRNA is:

$$Var_{\langle M\rangle} = \frac{\partial^2 G}{\partial z_1^2} + \langle M\rangle - \langle M\rangle^2 = \langle M\rangle \tag{3.62}$$

The variance for protein is:

$$\begin{aligned} Var_{\langle P\rangle} &= \frac{\partial^2 G}{\partial z_2^2} + \langle P\rangle - \langle P\rangle^2 \\ &= \frac{k_p}{\gamma_p(\gamma_r+\gamma_p)} k_r \left(\frac{k_p}{\gamma_r}\right)\left(\frac{k_r}{\gamma_p}+\frac{k_r}{\gamma_r}+1\right) + \langle P\rangle - \langle P\rangle^2 \\ &= \frac{k_p}{\gamma_p}\frac{k_r}{\gamma_r} + \left(\frac{k_p}{\gamma_r+\gamma_p}\left(\frac{k_r}{\gamma_p}+\frac{k_r}{\gamma_r}+1\right)+1-\langle P\rangle\right) \\ &= \langle P\rangle + \left(\frac{k_p}{\gamma_r+\gamma_p}\left(\frac{k_r}{\gamma_p}+\frac{k_r}{\gamma_r}+1\right)+1-\langle P\rangle\right) \end{aligned} \tag{3.63}$$

So, without the need to solve the CME and making use of the Probability Generating Function, we have found out the mean and the variance for the mRNA and the protein. So the noise can be computed using the formula for the Fano factor, and the theoretical noise expressions for mRNA and protein are given by:

$$Fano_{\langle M\rangle} = \frac{Var_{\langle M\rangle}}{\langle M\rangle} = \frac{\langle M\rangle}{\langle M\rangle} = 1. \tag{3.64}$$

$$\begin{aligned} Fano_{\langle P\rangle} &= \frac{Var_{\langle P\rangle}}{\langle P\rangle} = \frac{\langle P\rangle + \left(\frac{k_p}{\gamma_r+\gamma_p}\left(\frac{k_r}{\gamma_p}+\frac{k_r}{\gamma_r}+1\right)+1-\langle P\rangle\right)}{\langle P\rangle} \\ &= \frac{k_p}{\gamma_r+\gamma_p}\left(\frac{k_r}{\gamma_p}+\frac{k_r}{\gamma_r}+1\right)+1-\frac{k_r k_p}{\gamma r \gamma p} = \ldots = 1+\frac{k_p}{\gamma r+\gamma p}. \end{aligned} \tag{3.65}$$

These formulas are the same as in the work of Thattai & van Oudenaarden (2001). The only difference is in the notations. Thattai & van Oudenaarden (2001) used the notation $b = \frac{k_p}{\gamma_r}$ and $\eta = \frac{\gamma_p}{\gamma_r}$, resulting in:

$$Fano_{\langle P \rangle} \quad = \quad \frac{Var_{\langle P \rangle}}{\langle P \rangle} \quad = \quad 1 + \frac{b}{1+\eta}. \tag{3.66}$$

Example of Simple Model: Gadgil *et al*

To apply the method of Gadgil *et al* to the *Simple Model* is much easier and quicker than for the *General Model*, because there are only two species (mRNA and protein) to study, instead of four, and then the dimensions of all the matrices involved are smaller.

Here some tips are given for the different matrices computed for finding out the Fano factor for the two reactants involved. The vector of the means is defined only for the two species:
$M(t) = [\mathbb{E}[M], \mathbb{E}[P]]$.

The K's matrices of rate constants are:

$$K^s = \begin{pmatrix} k_r & 0 \\ 0 & 0 \end{pmatrix}; \quad K^d = \begin{pmatrix} \gamma r & 0 \\ 0 & \gamma p \end{pmatrix}; \quad K^{con} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}; K^{cat} = \begin{pmatrix} 0 & 0 \\ k_p & 0 \end{pmatrix}.$$

Then $\mathcal{K}$ is defined as $\mathcal{K} = (K^{con} + K^{cat} - K^d) \quad \Longrightarrow \quad \mathcal{K} = \begin{pmatrix} -\gamma r & 0 \\ k_p & -\gamma p \end{pmatrix}$.

Thus, the first-order moments can be calculated by Eq. 3.33, and the results are

the following:

$$\begin{cases} E[M] &= \dfrac{k_r}{\gamma_r} \\ \\ E[P] &= \dfrac{k_p}{\gamma_p} E[M] \end{cases} \tag{3.67}$$

To find the second-order moments (by Eq. 3.34), the involved matrices are the following:

$$V(t) := \begin{pmatrix} E[M^2 - M] & E[MP] \\ E[PM] & E[P^2 - P] \end{pmatrix} = \begin{pmatrix} E[M^2 - M] & E[MP] \\ E[MP] & E[P^2 - P] \end{pmatrix}.$$

The term $\mathcal{K}V(t)$ is defined as,

$$\mathcal{K}V(t) = \begin{pmatrix} -\gamma r E[M^2 - M] & -\gamma r E[MP] \\ k_p E[M^2 - M] - \gamma p E[MP] & k_p E[MP] - \gamma p E[P^2 - P] \end{pmatrix}$$

and $\Gamma_{ij}(t)$ as,

$$\Gamma_{ij}(t) = \begin{pmatrix} k_r E[M] & k_r E[P] \\ k_p E[M] & 0 \end{pmatrix} \implies \Gamma_{ij}^t(t) = \begin{pmatrix} k_r E[M] & k_p E[M] \\ k_r E[P] & 0 \end{pmatrix}.$$

The computation of the second-order moment matrix, $\frac{d}{dt}V(t)$, gives

$$\begin{pmatrix} -2\gamma r E[M^2-M]+2k_r E[M] & k_p E[M^2-M]-(\gamma p+\gamma r)E[MP]+k_p E[M]+k_r E[P] \\ k_p E[M^2-M]-(\gamma p+\gamma r)E[MP]+k_p E[M]+k_r E[P] & 2(k_p E[MP]-\gamma p E[P^2-P]) \end{pmatrix}$$

Setting the system at steady-state (= 0) results in the second-order moments from

which the variances and Fano factors are readily written. The variances are:

$$Var_{\langle M\rangle} = \frac{k_r}{\gamma_r} \tag{3.68}$$

$$Var_{\langle P\rangle} = E[P]\left[\frac{k_p}{\gamma_r+\gamma p}\left(E[M]+1+\frac{k_r}{\gamma_p}\right)+1-E[P]\right]. \tag{3.69}$$

The Fano factors are given by

$$Fano_{\langle M\rangle} = \frac{Var_{\langle M\rangle}}{E[M]} = \frac{E[M]}{E[M]} = 1. \tag{3.70}$$

$$Fano_{\langle P\rangle} = \frac{Var_{\langle P\rangle}}{E[P]} = 1+\frac{k_p}{\gamma r+\gamma p}. \tag{3.71}$$

Finding out that the expressions for the noise of mRNA and protein, that have been obtained by the method of Gadgil et al. (2005) (Eq. 3.70 and 3.71), are the same as the results obtained by the Probability Generating Function Approach for the *Simple Model* (results given by Eq. 3.64 and 3.65 or 3.66).

3.4 Conclusions of the methodology

The first analytical method studied in this thesis, the Probability Generating Function, is a standard method in Probability Theory (see the book of Van Kampen (2007)). This method uses the explicit formula of a probability generating function associated to a probability distribution and their properties of the first and second order moments to explain the expectation and variance at the same time. This method has become one of the main components to develop the formulae for the means and the variances

of the stochastic reactants involved in the differential chemical master equation that described the *Simple Model*. With the two first moments for the mRNA and protein, we could also find out the formulae for the Fano factor and all the results compared to the expressions obtained by the method of Gadgil et al. (2005). Thus, both theoretical methods ended in the same results for the *Simple Model*. Finally, we have assessed the credibility of the findings with also equal results (also see Higham & Khanin (2008)).

These results have evolved in the development of the formulae for a more complex model, the *General Model*, where the second analytical method by *Gadgil et al.* was applied. It has appeared to be shorter than the probability generating approach, in terms of computations, and it allowed us one more time to develop the second stage of formulae that defined the means and variances for the mRNA and protein of this model, and consequently to find out the formula for the noise (or Fano factor). This analytical evaluation for the noise was then used to compare with numerical methodology in Chapter 4, specifically in section 4.2.

In addition, three numerical methods were compared and first evaluated on a Dimer example in this section, and in further analysis (it will be seen in 4). Numerical methods have been coded to simulate several systems and scenarios, elucidating that all three: the SSA, the Langevin Equation and the Tau-Leaping methods are good approximations to simulate the behaviour of our stochastic models.

Chapter 4

Noise Analysis

4.1 Introduction

The principal goal of this thesis is to estimate the intrinsic noise through the simulation of stochastic systems for different gene expression models (introduced in *Chapter* 2). Specifically, the aim is to analyse the behaviour of the system at a steady-state and study how their properties vary with the number of molecules. Furthermore, the aim is to find out the factors that affect this variance and, if possible, ways of regulating the genes to reduce this variance.

Steady state, in terms of stochastic systems, is when the system conserves a 'particular form' in the behaviour for all the variables (number of molecules of reactants) relative to it. Because of the intrinsic variation always generated in the cell, this steady state will never be constant, but the means of the variables and the variations produced by the noise will remain bounded because the variables describe a constant probability. Although, if the system is very noisy, the boundaries will be quite large and the steady state will not be too precise. Even so, steady state implies that the probability that various different states will be repeated will remain constant and this stable situation will enable the prediction of the system in the future.

Once the stochastic system reaches the steady state, and from the range of variability obtained in this situation, the mean and variance of the variables of interest can be computed and analysed.

4.2 Analysis of the model with *One Target*

4.2.1 In-silico experiment 1: Steady-state & noise by promoters

This study focuses on exploring the mean, $\langle q \rangle$, and the variance, δq^2, of the number of molecules of each species q in the steady state for several gene regulatory modules. The modules are three different simulated cases of gene regulation by miRNA and three different promoters, and is the first part of a larger study concerning the effect of miRNA on gene transcription.

The aim is to study whether miRNA reduces noise in protein production as has been suggested by several authors (Hornstein & Shomron 2006). In addition, it is informative to study whether the effect of miRNA reduction of noise is the same for different promoter types.

The model used to describe the process for this experiment is the *General-miRNA model*, explained in *Chapter* 2 (Figure 2.4).

The various miRNA cases (Table 2.2) depend on a 'signal' which only affects the presence/activation of them. This means that if there is no signal there are no miRNAs, but if there is a signal, miRNAs can exert its downregulating effect on either mRNA degradation or translation and mRNA degradation at the same time. The frequency of promoters being in an active state largely depends on the activation/desactivation

rates. The three different structures of the model, depending on the location and presence of miRNA, are studied through these three different promoter cases, implying that the parameters of the system are fixed for the three miRNA cases, while the rate constants of the activation/deactivation of the promoters are changed according to the three different types.

Three promoters simulated in this study (Table 4.1), only differ in the relative magnitude of the rate constants of two reactions. Table 4.1 specifies the rates used in this study for the three types of promoter: the first promoter, *stable*, has slow but equal rates; the second, *unstable*, has a slow activation step and much faster inactivation; and the third, *prokaryotic*, has larger, and equal, rates than the other promoters. Consequently in the global structure of the model this third promoter has more chance of changing from active to inactive state, and vice versa, than the other two promoters.

Table 4.1. Promoter Rates

	Rate constants	
Promoters	k_a	k_d
Stable (1)	0.001	0.001
Unstable (2)	0.001	1
Prokaryotic (3)	1	1

Promoter rates for 3 promoter types (see Table 2.1); (k_a goes from inactive state to active of the promoter, k_d from active state to inactive). The transcription rate used was $k_r = 0.1$.

To have a further idea about how a small or large reaction rate value can affect one of these genetic circuits, one may think for a moment of the following situation: *If the rate constant to form 'reactant X' is a small value, it means that the rate at which 'reactant X' forms is slow, thus with less production or a low probability to form 'reactant X' than if it was a larger value.*

In summary, with the three different modes of regulation by miRNA and the three promoter types considered, nine cases are studied.

Methods & Results

In this study, two methods are used, one numerical and the other theoretical. The numerical method is the Gillespie Algorithm (Higham 2007) that simulates the system of the *General Model* at steady state for the nine cases, obtained to combine three miRNA modes of regulation and three promoter types described in the previous section, and to estimate an expression for the noise. The analytical procedure is derived from the formulae by Gadgil et al. (2005) to study the noise theoretically and to compare results obtained by these two methods.

The Gillespie Algorithm gives an approximation of the system's variable values, such as the number of molecules of mRNA or protein, and simulates one reaction at a time over an interval fixed by the user. The simulation is carried out over a large period of time to make sure that it takes the system to a steady-state. From this steady-state we compute an expression for the noise.

The parameters used for this model are the rate constants used in (Bundschuh et al. 2003). The initial values for the reactants of the system (*D, D*, mRNA, protein*) are obtained from the steady-states values computed by the deterministic approach. This approach is defined for a non-linear ordinary differential equations (ODEs) in the system 2.10 (defined in *Chapter* 2), which produces concentrations of the chemical species instead of counts of molecules, as does the stochastic approach. The initial values, therefore, come from setting up of the rate equations of ODEs to zero:

$$\begin{aligned}
(0 =) \quad \frac{dD}{dt} &= k_d D^* - k_a D \\
(0 =) \quad \frac{dD^*}{dt} &= k_a D - k_d D^* \\
(0 =) \quad \frac{dM}{dt} &= k_r D^* - \gamma_r M \\
(0 =) \quad \frac{dP}{dt} &= k_p M - \gamma_p P,
\end{aligned} \tag{4.1}$$

which results in the mean values at steady-state:

$$\begin{aligned}
D &= \frac{k_d}{k_a + k_d} D_{total} \\
D^* &= \frac{k_a}{k_a + k_d} D_{total} \\
M &= (k_r/\gamma_r) D^* \\
P &= (k_p/\gamma_p) M.
\end{aligned} \tag{4.2}$$

These mean values, before the start of the simulation, have to be introduced as integers, i.e. we round them in the algorithm, because the Gillespie Algorithm works for integers numbers of the reactant variables, and with the correspondent parameters.

For the analytical procedure, because all the reactions are unimolecular, the method of (Gadgil et al. 2005) is applied using the formula of the Fano factor obtained theoretically (see *Section* 3.3.2), and substituting the relevant parameters for each of the cases of miRNA modes of regulation and promoter types. This yields the following formulas (the same as formulas 3.43 and 3.44 in *Section* 3.3.2) for mRNA and protein

respectively:

$$Fano_{\langle M\rangle} = 1 + \frac{k_d k_r}{(k_a + k_d)(k_a + k_d + \gamma_r)}$$

$$Fano_{\langle P\rangle} = 1 + \frac{k_p}{\gamma_r + \gamma_p}\left(1 + \frac{k_d k_r}{(k_a + k_d)(k_a + k_d + \gamma_r)}(1 + \frac{\gamma_r}{k_a + k_d + \gamma_p})\right)$$

In the prokaryotic promoter, the main source of stochasticity is translational bursting (Thattai & van Oudenaarden 2001), or the number of proteins produced per lifetime of a transcript. This is reflected in the parameter called translational efficiency ($b = k_p/\gamma_r$).

The general formula for the protein Fano factor (Eq. 3.44) for the case of prokaryotic promoter reduces to:

$$Fano_{\langle P\rangle} = 1 + \frac{k_p/\gamma_r}{1 + \gamma_p/\gamma_r}$$

Indeed, because $k_a, k_d, \gg k_r$, the term

$$\frac{k_d k_r}{(k_a + k_d)(k_a + k_d + \gamma_r)} \ll 1.$$

The mRNA molecules have Poisson distribution with Var(M) =< M > and $Fano_{\langle M\rangle} = 1$.

The eukaryotic model of gene regulation has additional sources of noise due to promoter switching from inactive to active states (Raser & O'Shea 2004). Infrequent (relative to transcription) promoter activation rate contributes to fluctuations in mRNA and protein levels (*transcriptional bursting*). The size of bursts in transcription depends on the average number of mRNAs produced between promoter activation and deactivation; the ratio k_r/k_d is referred as *transcriptional efficiency* (Kærn et al. 2005). If the

transition rates of promoter activation and deactivation are very slow, the mRNAs and proteins track the states of the promoter. In this case, miRNA can be very efficient in reducing the noise, as it reduces translational efficiency, by either enhancing degradation rate of target mRNA, γ_r, and/or decreasing translation rate, k_p. See parameters k_p/γ_r and γ_p/γ_r above.

It is therefore tempting to speculate that noise in protein production is minimized for essential, highly-connected, genes, wherein there is an additional burden on translational rate, that for eukaryotes can be exerted via miRNAs.

Fraser et al. (2004) estimated the noise in protein production for nearly every yeast gene. These authors found that noise in protein production is minimized in genes for which it is likely to be most harmful, specifically essential genes and genes encoding protein complex subunits. The noise is minimal for genes with high (frequent) transcription and low (inefficient) translation rates. Noise minimization is not without a cost as the high transcription and high mRNA decay rates that are needed to minimize noise are energetically expensive and are thus expected to be advantageous only when the benefit of reducing noise in a particular gene's expression outweighs this cost.

The role of miRNA in reducing noise in protein output might also explain the seemingly counterintuitive prevalence of positive expression correlation in miRNA-target pairs found in human and mouse genomes (Tsang et al. 2007), as the role of miRNAs might be to decrease the translational efficiency and to reduce the noise in protein output, rather than eliminate it completely.

The results obtained from the stochastic simulation of a system with the *General-miRNA model* are displayed in Table 4.2, which contains the mean and standard deviation of the mRNA and protein levels at steady-state. The results for the noise, given by the Fano factor measure of mRNA and protein are shown in Table 4.3, for the

numerical study (using Eq. 3.23) and the analytical method (formulas given by Eq. 3.44).

Table 4.2. Steady-states for mRNA and protein

miRNA mode of regulation	Promoter case	mRNA		protein	
		mean	std	mean	std
1	Stable (1)	30.1	16.4	1614.6	667.0
	Unstable (2)	1.5	1.4	48.6	26.1
	Prokaryotic (3)	26.5	5.2	1598.5	124.0
2	Stable (1)	12.1	6.2	627.5	234.8
	Unstable (2)	1.1	1.3	19.0	10.5
	Prokaryotic (3)	8.7	2.9	527.7	38.6
3	Stable (1)	12.3	6.5	212.4	81.4
	Unstable (2)	1.2	1.5	5.5	3.6
	Prokaryotic (3)	8.6	2.9	174.7	21.4

The means and standard deviations for the mRNA and protein at steady-states, for each different miRNA and promoter cases. Simulation run-time 80.000 time units.

Looking at the results of Table 4.2 the largest levels of molecules for means of protein are given by the *stable* promoter followed by the *prokaryotic*. They are the ones that allow more production of protein, while the *unstable* promoter, produces a low amount of protein. In part, this was expected from the definition of this second promoter (in Table 4.1), where the activation rate of the gene is much slower than its deactivation, being less probable for the gene to be in active state and consequently lower protein production. It is also shown that the presence of miRNA in the system represses the production of protein, in particular when it exerts its regulation on mRNA degradation and translation (case *3*), more so than when it does it only on mRNA degradation (case *2*). In this case *3*, all promoters exhibit a reduction in the level of protein in comparison to the others. In contrast, the levels of mRNA that are also reduced with the presence of miRNA are not affected differently for being regulated by miRNA mode *2* or

Table 4.3. Noise at steady-states for mRNA and protein

		NUMERICAL		ANALYTICAL	
miRNA mode of regulation	Promoter case	Fano factor		Fano factor	
		mRNA	protein	mRNA	protein
1	Stable (1)	8.89	275.57	9.47	203.52
	Unstable (2)	1.33	14.02	1.94	19.47
	Prokaryotic (3)	1.02	9.62	1.02	10.56
2	Stable (1)	3.20	87.84	4.65	71.80
	Unstable (2)	1.49	5.82	1.88	7.85
	Prokaryotic (3)	0.97	2.82	1.02	4.55
3	Stable (1)	3.47	31.19	4.65	24.60
	Unstable (2)	1.80	2.38	1.88	3.28
	Prokaryotic (3)	0.97	2.63	1.02	2.18

Noise in mRNA and protein levels at steady-state computed by numerical and analytical methods (using the Fano factor).

3 because what matters here is that miRNA acts on (increasing) the mRNA decay rate.

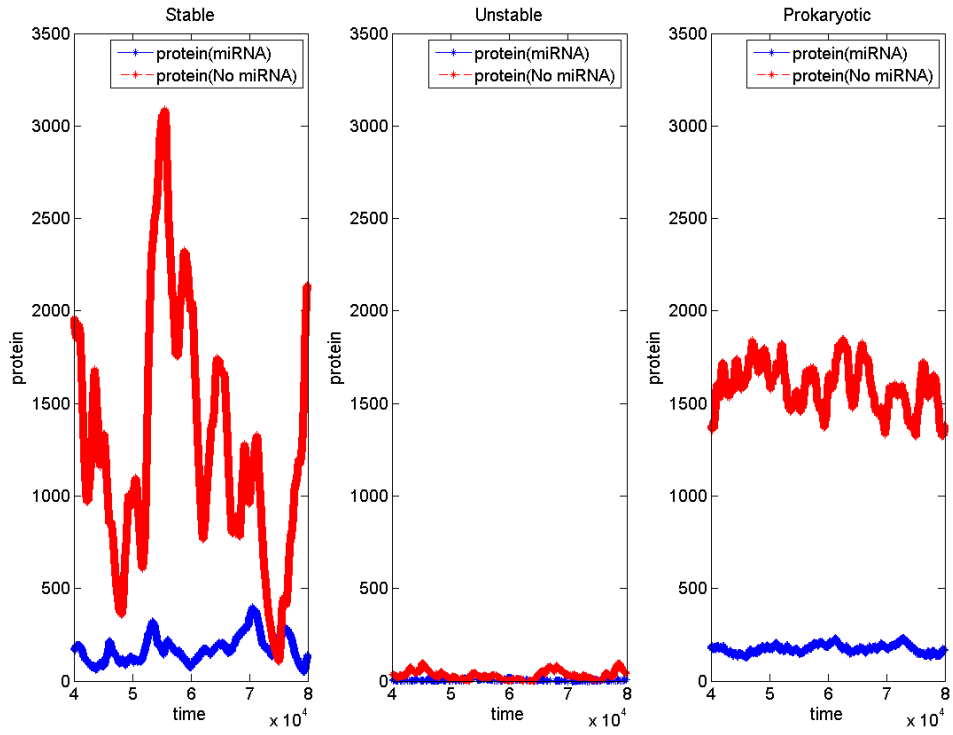
Table 4.3 presents results for the noise. This table shows how similar are the results comparing the two methods. Analytical and Numerical methods show the same pattern for the Fano factors in the different miRNA modes of regulation and for all three promoters. The values of the Fano factor for each case just differ by a small amount between the two methods. The *stable* promoter seems to be the noisiest among three cases, while the *prokaryotic* the least noisy. Both methods demonstrate that the system without miRNA (case *1*) has larger values of noise for all three promoters. Instead, when miRNA is present in the system (cases *2* and *3*) the noise is reduced, in particular the noise is small when miRNA acts on both, mRNA degradation and translation.

Figure 4.1 shows a plot illustrating the protein behaviour at a steady-state for the

three different promoters. Here the three plots represent the system for the three promoters, each of them displaying the steady-states of protein for the systems with and without miRNA. The plots are in the same scale and the noise is represented by jagged lines, some more noisy than other ones, and with more or less variability. The noisiest of the promoters is the *stable* and the least one is the *prokaryotic*. Visually it is not clear that the *unstable* promoter is not the least noisy because of the scale of the plots and that is why afterwards Figure 4.2 is introduced to illustrate its instability. Besides, it is clear that the *stable* promoter is the noisiest because the variance is wide, equally so for the system with miRNA (red lines) as for that without (blue lines), in concordance with the numerical results obtained in Table 4.3. Furthermore, from these plots (in exception of the plot for the *unstable* promoter that is clarified in Figure 4.2), the height and width of the lines representing the system without miRNA, show that miRNA represses the production of protein, thereby reducing the noise since its protein production is expressed under the system without miRNA.

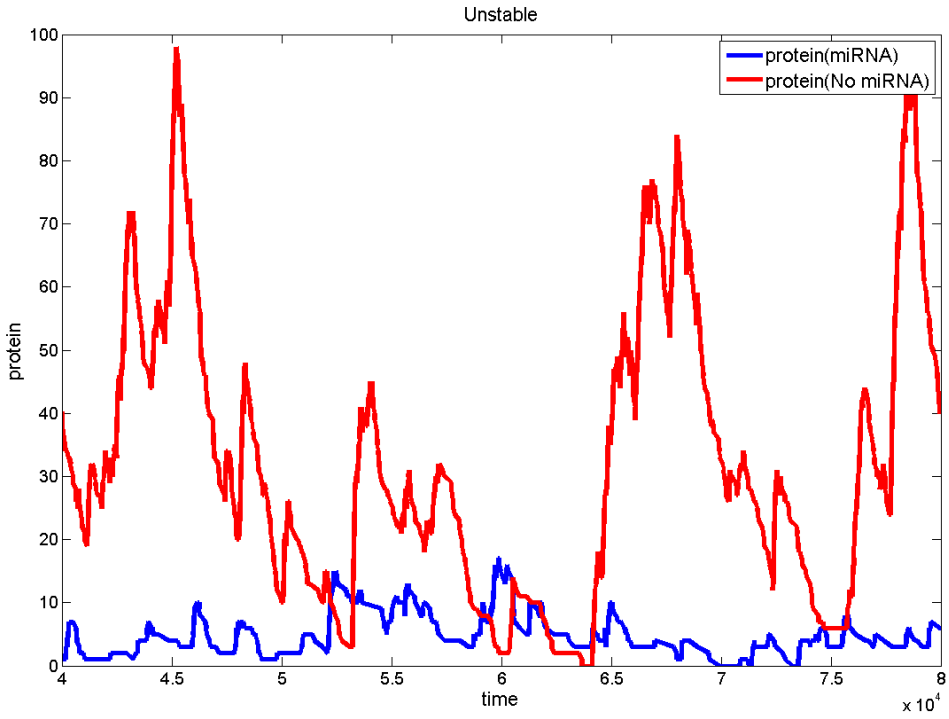
The plot in Figure 4.2 is a zoom in to the middle (second) plot in Figure 4.1 illustrating the noise produced by the second (*unstable*) promoter. The increased scale of the y-axis allows to see more clearly how noisy it is. For example, the two systems cross at some points in time, where the protein production of the system without miRNA is below the levels produced for the system with miRNA. This happens just for a few periods of time, and could be caused by the huge noise produced in the first system.

Figure 4.1. Noise of protein at steady state for three different promoters.



The fluctuations in the protein levels at steady state for three promoters (stable, unstable, and prokaryotic respectively): without miRNA (red line) and with miRNA present (blue line). Protein levels (molecules number) are plotted against time (only the steady-state interval).

Figure 4.2. Noise of protein at steady state for the *unstable* promoter.



The fluctuations in the protein levels for the unstable promoter at steady state: without miRNA (red line) and with miRNA present (blue line). Protein levels (molecules number) plotted against time (only the steady-state intervals) have several scattered peaks at diff erent values of protein, demonstrating that the *unstable* promoter is very noisy.

Conclusions

In this in-silico experiment we have tested three different modes of gene regulation by miRNA for three different promoters. The main observations are:

1. Protein and mRNA levels at steady-state are much higher for the system with a *stable* promoter than an *unstable*, while the *prokaryotic* produces medium levels of proteins (Table 4.2).

2. *Stable* promoter is the noisiest, the higher Fano factor values in the columns of Table 4.3 are always in this promoter. Thus, comparing *stable* and *prokaryotic* promoters that have similar mean protein output (Table 4.2), their Fano factors differ 20-fold.

3. The production of protein is significantly decreased with the third miRNA mode of regulation (both miRNA decay and protein translation are down), followed by the second mode (mRNA decay only) which shows a smaller but also measurable decrease. Meanwhile, for the mRNA, the presence of miRNA in the system (*modes 2* and *3*) decreases levels of mRNA as well, but between both modes there is no difference (Table 4.2).

4. In the presence of miRNA the noise decreases for all three promoters. The largest decrease is for the *stable* promoter. (See Table 4.3, that with *mode 2* and *3* the values of Fano factor are smaller than with *mode 1*).

5. *Stable* promoter is highly sensitive to the decrease of noise of protein with the appearance of miRNA, as we can see from the results with a ratio of 8.8 ($\sim$ 275.57 / 31.19), from the numerical results, and ratio of 8.2 ($\sim$ 203.52 / 24.60), from the analytical results.

6. Noise in *prokaryotic* promoter is the least affected by the presence of miRNA, with a ratio of 3.6 ($\sim$ 9.62 / 2.63), from the numerical results, and with a ratio of 4.8 ($\sim$ 10.56 / 2.18), from the analytical results.

7. The effect of miRNA is less significant in the noise of mRNA than protein. Moreover, from Table 4.3, it is trivial that the Fano factor of mRNA, for the *prokaryotic* promoter, defines a Poissonian distribution of mRNA because all the values are approximately 1.

8. The Fano factors show similar values for the two methods applied, verifying and matching the results for the numerical study as for the analytical.

Finally, it can be concluded that the presence of miRNA decreases protein production; the presence of miRNA decreases the noise as being measured here by the Fano factor for all promoters; and the noisiest promoter, in terms of the Fano factor measure of protein levels, is the *stable* promoter and it shows the largest decrease caused by miRNA.

4.2.2 In-silico experiment 2: Transition Times

Biological systems change over time. This can be the result of different external factors and specific signals, e.g. high osmosis or stress caused by physical or chemical parameters, such as pressure or harmful molecules. The system response can turn a regulation of specific genes and pathways on or off. To understand biological systems it is important to know how quickly and reliably different genetic circuits respond to external stimulus. For example, what type of gene regulatory circuit responds quicker (and with less variability) when regulation is turned on or off. Problems of this type have been studied by several groups, including Shimoni et al. (2007) for small RNAs.

Two types of regulation will be contrasted here, post-transcriptional target regulation by miRNA and transcriptional target regulation by a transcription factor (TF). It is assumed that the "signal" a sudden change in the external conditions, turns on the regulation. In the first scenario the signal activates the presence of miRNA: the miRNA molecules bind the transcripts of the target gene, accelerate target mRNA degradation

rate and inhibit the protein translation step. In the transcriptional regulation, signal activates production of TF: the regulatory TF binds to the promoter of the target gene and represses its transcription.

Therefore, the aim of this study is to describe quantitatively the transition time that a specific system takes to change from one state to another. Usually, transition time is measured as half-time, or the time it takes the system to move half-way between two steady states that correspond to different levels of signal. For example, looking at protein levels and considering two states depending on signal (*signal*=$\{0,1\}$), the transition time that goes from *signal=0* to *1* is $t_{1/2}$, where:

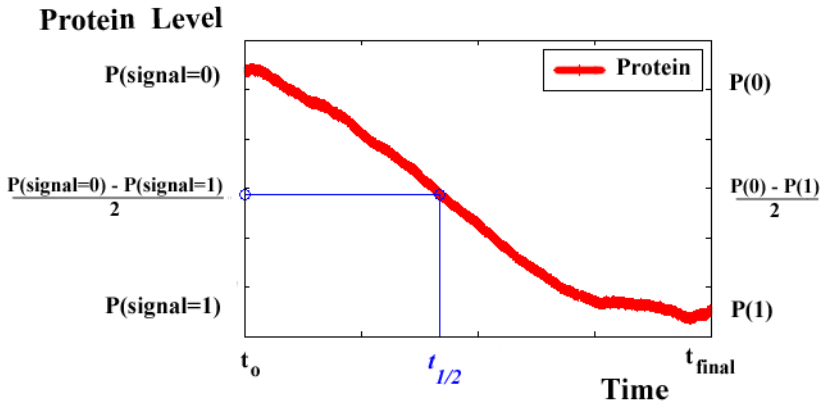
$$t_{1/2} = \quad \text{time to go from } P(signal=0) \text{ to } \quad \frac{|P(signal=0) - P(signal=1)|}{2} \tag{4.3}$$

Figure 4.3 illustrates the transition time for a protein from a state *0* to a state *1* (of the signal). The level of the target protein (number of molecules) is presented versus time, starting from the moment of time at which the regulation is turned on. At time *t=0*, a "signal" arrives turning on the regulation.

The properties of post-transcriptional regulation by miRNA to transcriptional regulation by TF will be also compared.

For the first scenario of post-transcriptional regulation the system is modeled by the *General-miRNA Model*. The study of this circuit is compared for each of three different promoters and for each of three different miRNA location cases. The different modes of regulation by miRNA are dependent on the activation of the "signal" over time, which means that without a signal there is no miRNA, and if the signal is on, the miRNA is present. MiRNA acts on the system either in the step of mRNA degradation or in two steps at a time: mRNA degradation and translation (see Table 2.2).

Figure 4.3. Transition time for protein from state *0* to *1*



The transition time from the state=0 (no signal, Protein(signal=0)) to the state 1 (signal present, Protein(signal=1)). The transition time, t1/2, is computed as time it takes the system to go from the original state (state=0) to the state that is half-way between state=0 and state=1. The x-axis is time, the left and right y-axis show the protein levels (same scales).

Combining three promoter types with the three miRNA modes of regulation gives a total study of nine different cases of the circuit. The promoters are dependent on the activation/deactivation rates of the gene (see Table 2.1), whilst the miRNA modes of regulation are determined by the location of miRNA in the system (Table 2.2). If miRNA is not present (signal is off) then the model is reduced to the *General Model*. In the second scenario, transcriptional regulation as described by the *General model* has been considered, wherein the TF only affects the transcription rate, k_r.

To compute transition time from the *in-silico* experiment, an initial and a final state of the system have to be computed *a priori*. The initial state is the one at which the system starts (at time *t=0*), and the final state is reached by the system when the signal is permanently on. Protein profiles change between two steady-states: one of the system that is not subjected to regulation, and the other when the regulation is turned on. For each of these cases the transition time is calculated as the time it takes the protein to go half-way between two steady-states calculated as means. For instance,

when a single gene is repressed, the transition time denotes the half-way between a starting point (steady-state) where the system is not regulated and a final state where the system is repressed. The transition time for the recovery of the target gene indicates the half-way between the starting point of the system repressed at steady-state and the final state where the regulation is off (system is de-repressed).

Experimental Results

The Gillespie's stochastic simulation (Higham 2007) is used in this study for two main simulations. Firstly, system's steady state at the initial and final states of interest, *P(0)* and *P(1)* are computed. Secondly, the transition times between these two states are computed. Here transition time is defined as the time it takes for a protein to go from its initial state (*P(0)*) at $t = 0$ to half level between *P(0)* and *P(1)* (see Eq. 4.3). For each protein level *P(0)* and *P(1)* and all types and different modes of regulation by TF and miRNA, a confidence interval (CI) of transition time is computed as *mean* $\pm$ *std*.

The transition times for transcriptional and post-transcriptional types of regulation are compared using the constraint that the initial and final states for both regulatory systems are fixed at the similar values. When the regulation is off, protein levels are the same in both types of regulatory circuits: regulatory circuits without miRNA or TF reduce simply to the *General Model*. When the regulation is switched on, the transcription rate, k_r for TF model, is adjusted to result in the same value as the repression caused by miRNA to insure similar levels in both systems. The following steps set up this type of equivalence between states:

1. Compute steady-states for the system regulated by miRNA (6 cases = 6 steady-states). Store only the protein levels (P).

2. Find out k_r, the transcription rate constant for the system regulated by TF.

Using ODEs system 4.2 and the values from step *1*, compute:

$$\langle P\rangle = \frac{k_p}{\gamma_p}\langle M\rangle \Longrightarrow \langle P\rangle = \frac{k_p k_r}{\gamma_p \gamma_r}\langle D^*\rangle \Longrightarrow k_r = \frac{\gamma_p \gamma_r}{k_p \langle D^*\rangle}\langle P\rangle$$

where k_p, γ_p, γ_r, $\langle P\rangle$ and $\langle D^*\rangle$ are known and fixed at the same level for both scenarios.

(Obtaining then the k_r rates for all 6 protein levels stored previously).

One has to note that the k_r values computed above are used for stochastic simulations of the circuit regulated by a TF. This sometimes result in protein values that are different from the ones obtained from deterministic approach.

Trials of 100 runs are computed for each scenario. The computation of the system's steady state is run over 80,000 time units to make sure that the system reaches steady-state. To determine the transition times, the time chosen for the simulation is shorter, just 4,000 time units, because all the transition time values are always under 2,000 units. The mean and standard deviation for 100 runs, confidence intervals and noise (Fano factor) of the response times are computed and recorded in Tables 4.4 -4.5.

Table 4.4 shows the transition times obtained from the study of the system repression by two types of regulators. The means of the transition times in the case of miRNA and TF regulation are different (looking at rows) only for the *stable* and the *prokaryotic* promoters. The CIs, with a confidence of 95%, also show this difference because the results do not overlap at all. Instead, the *unstable* promoter does not show a significant difference between the two types. The noise is very small for the *stable* and *prokaryotic* promoters, when the system is repressed by miRNA acting on both mRNA degradation and translation (system repressed to case3). Moreover, in the same cases the system regulated by miRNA shows a faster response than regulated by TF. This study of TF-regulation could have been restricted by considering three cases only: three different promoters. However, six cases are considered, as in miRNA regulation,

because the transcription rate k_r of this system is set up depending on the system regulated by miRNA.

In Table 4.5, the transition times for the recovery (de-repression) of the system are computed after the repression has been switched off. Looking at the noise of all the cases, the *prokaryotic* promoter is the only one that shows lower Fano factors, i.e. *prokaryotic* promoter is the least noisy. In general for the recovery of the system it can be seen that the means and the 95% CIs of the transition times for both types of regulations (by miRNA and TF) show similar responses of the system. In fact, the CIs overlap. Morover, the noisiest transition times are given by the *unstable* promoter because has the largest noise. This is probably because the steady-states for a system with or without miRNA are very close.

Now, let us compare the transition times for different modes of miRNA regulation for each promoter type with the results obtained in the Table 4.4. It is clear that when miRNA only affects degradation of target mRNA, the transition times in protein are extremely noisy for all three types of promoters (Table 4.4, column of Fano factor for the system regulated by miRNA). So, for targets that are repressed purely on the level of mRNA, it is difficult to predict the time it takes to reach a required protein level. It is therefore tempting to speculate that genes that switch on vital pathways via miRNA upon the arrival of a signal are regulated on protein translation level. This ensures a more predictable time for switching pathways *on* and *off*, in particular for the *stable* and *prokaryotic* type of promoters, because they are less noisy than the *unstable*.

In addition, achieving similar level of protein repression, less time is required for miRNA-regulated genes, when miRNA affects mRNA degradation and translation, than for TF-regulation. Importantly, the transition times for these modes of miRNA-regulation are less noisy than TF-regulated genes (for all three types of promoters). Similar results hold for de-repression.

Table 4.4. Transition times for the Repression of the target gene

Promoters	INITIAL *state:*	FINAL *state:*	Regulation by miRNA				Regulation by TF			
			mean	*std*	*95% CI*	*Fano*	*mean*	*std*	*95% CI*	*Fano*
Stable	*no repression*	*case2*	1224.4	423.2	(1141.4, 1307.3)	146.3	819.6	66.7	(806.5, 832.7)	5.4
		case3	875.7	34.8	(868.9, 882.5)	1.4	1106.8	70.3	(1093.0, 1120.6)	4.5
Unstable	*no repression*	*case2*	1080.5	445.3	(993.2, 1167.8)	183.5	944.6	421.7	(861.9, 1027.2)	188.3
		case3	987.3	200.9	(947.9, 1026.7)	40.9	1168.5	434.1	(1083.4, 1253.6)	161.3
Prokaryotic	*no repression*	*case2*	1112.6	129.9	(1087.1, 1138.5)	15.2	874	71	(860.1, 887.9)	5.8
		case3	895.4	37.9	(888.0, 902.8)	1.6	1087.5	71.9	(1073.4, 1101.6)	4.7

The mean, std, CI and Fano factor of the transition times for the repression of the target gene for two types of regulation obtained from 100 simulations. Three promoter types (table 4.1), and two repressed cases of final states (table 2.2). The transcription rates, k_r, in the regulation by TF, for the 6 cases, are respectively: 0.0334, 0.0133, 0.0013, 0.0002, 0.0322, 0.0114 (to have equivalent final states for both regulation types).

Table 4.5. Transition times for the Recovery of the target gene

Promoters	INITIAL	FINAL	Regulation by miRNA				Regulation by TF			
	state:	*state:*	*mean*	*std*	*95% CI*	*Fano*	*mean*	*std*	*95% CI*	*Fano*
Stable	*case2*	*no*	1527.9	940.1	(1343.6, 1712.1)	578.4	1319	903.7	(1141.9, 1496.1)	619.2
	case3	*repression*	1258.3	770.2	(1107.3, 1409.2)	471.4	1396.8	920.7	(1216.3, 1577.2)	606.9
Unstable	*case2*	*no*	658.8	900.4	(482.3, 835.3)	1230.6	779.3	1019.7	(579.4, 979.2)	1334.3
	case3	*repression*	743.5	912.9	(564.6, 922.4)	1120.9	899.7	959.9	(711.5, 1087.8)	1024.2
Prokaryotic	*case2*	*no*	1406.1	299.8	(1347.3, 1464.9)	63.9	1393.2	303.1	(1333.8, 1452.6)	65.9
	case3	*repression*	1271.7	199.4	(1232.6, 1310.8)	31.3	1316.9	217.6	(1274.2, 1359.5)	35.9

The transition times for the recovery of the target gene. The parameters and table elements are the same as in Table 4.4. Initial states are repressed (see table 2.2) and there is only one final no repressed state.

It can therefore be concluded that repression by miRNAs for *stable* and *prokaryotic* promoters when miRNA exerts its effect on protein translation (in addition to mRNA degradation) is clearly faster than the one mediated by TF. In this study, it is assumed that both TF and miRNA appear instantaneously as soon as the signal is switched on. In reality, production of TF follows the signal with some delay, reinforcing our conclusion that repression by miRNA happens on the faster time-scale than the repression on transcriptional level. This has indeed been proposed by Hobert (2008) and shown to be the case for small RNAs in bacteria using ODE model by Shimoni et al. (2007).

The simulated levels of proteins at steady-states are compared between the two methods: the stochastic simulation and the deterministic approach (ODEs). It is important to see if the values for steady-states, computed by these approaches are far away or close. Recall that to simulate the system stochastically, the levels of the different species were initialised using the ODE system (4.2). The results obtained by the Gillespie algorithm are in Table 4.6 and by the deterministic approach in Table 4.7.

From Table 4.6 it can be seen that the system regulated by TF represses the mRNA transcript more than the regulation by miRNA (mode *3* of miRNA regulation), however, for protein this is inversed (the system regulated by miRNA exerts repression on level of protein more than the regulatory circuit). Besides, the results of the deterministic approach (Table 4.7) do not show that regulation by miRNA represses protein more than regulation by TF for all three promoters. This is because the deterministic approach does not take into account the intrinsic stochasticity of this regulatory circuit. The steady-states for the first three rows in both tables are the same values (between regulation types) because the mode of miRNA regulation *1* denotes that the regulation is *off*, then means that the regulation by TF is also *off*, and then the system is the same in both cases. Finally, the most consistent promoter between the stochastic and deterministic results is the *prokaryotic*.

Table 4.6. Steady-states by Stochastic Simulation

miRNA mode of regulation	Promoters	By miRNA		By TF		
		mRNA	protein	mRNA	protein	k_r
1	Stable	31.2	1679.9	31.2	1679.9	0.1
	Unstable	1.6	57.2	1.6	57.2	0.1
	Prokaryotic	26.8	1630.4	26.8	1630.4	0.1
2	Stable	10.8	525.4	12.4	678.2	0.0334
	Unstable	1.2	19.7	0.9	29.1	0.0013
	Prokaryotic	8.5	505.5	9.8	520.8	0.0322
3	Stable	12.4	209.6	5.1	279.7	0.0133
	Unstable	1.2	4.1	0.6	14.9	0.0003
	Prokaryotic	8.7	179.2	4.7	234.7	0.0114

Steady-states of mRNA and protein computed by the Stochastic Approach, for two types of regulation (post-transcriptional by miRNA and transcriptional by TF). The rate k_r is the constant used in regulation by TF, which sets up equal final levels with the system regulated by miRNA.

Another study on the system post-transcriptionally regulated by miRNA has been carried out in this thesis to compare the protein levels at the different steady-states of the system with miRNA, and of the system without. (It should be noted that when miRNA is present, it is considered to act on mRNA degradation and translation). After computing the results for the transition times of the system while it was recovering from the repression of the regulation (Table 4.5), the Fano factors for $t_{1/2}$ in the case of the *unstable* promoter were very large, and we therefore investigate whether two steady-states separate from each other.

Figure 4.4 illustrates the protein levels for a system with and without miRNA for three different promoters. The distribution of the protein levels (number of molecules) is given by histograms. There are three plots, one for each promoter type. Each plot has two sets of protein data, representing initial and final states (with and without miRNA). In the cases wherein the histograms for the initial and final states overlap or

Table 4.7. Steady-states by Deterministic Approach

miRNA mode of regulation	Promoters	By miRNA		By TF		
		mRNA	protein	mRNA	protein	k_r
1	Stable	25.6	1571.4	25.6	1571.4	0.1
	Unstable	0.5	31.1	0.5	31.1	0.1
	Prokaryotic	25.6	1571.4	25.6	1571.4	0.1
2	Stable	8.5	523.8	11.6	525.4	0.0334
	Unstable	0.2	10.4	0.0	0.4	0.0013
	Prokaryotic	8.5	523.8	8.4	515.8	0.0322
3	Stable	8.5	174.6	3.0	184.2	0.0133
	Unstable	0.2	3.5	0.0	0.1	0.0003
	Prokaryotic	8.5	174.6	2.8	169.7	0.0114

Steady-states of mRNA and protein computed by the Deterministic Approach, for two types of regulation (post-transcriptional by miRNA and transcriptional by TF). The rate k_r has the same value that in the previous Table 4.6.

are very close, the estimation of the transition time ($t_{1/2}$) will not be right. Because by the definition transition time denotes the time to reach half-way between two different states of the system. However, if the initial and final states might not be different at all, the definition of transition time would not make sense here (for initial and final states not well distinguished).

In Figure 4.4, it is easy to see that the presence of miRNA optimises the production of protein (see Table 4.6). The histograms for the system with miRNA occupy the lower ranges on the graphs (situated more in the left side) than the system without miRNA which reaches much larger values in the protein synthesis. The means are given in the plots. For the first two types of the promoters, *stable* and *unstable*, the two steady-states overlap or almost overlap, with the specific parameters used to simulate the system in this study; this means that there is no significant difference between initial and final states. This is contrasted with the *prokaryotic* promoter case, where

the histograms do not overlap, and miRNA shifts protein range to lower values. In addition, miRNA sharpens protein range, or tunes protein levels, as it has been proposed by many experimentalists.

The parameters used in the simulations can be one of the causes that affects this overlap between two states (regulated and not by miRNA). That is why here, we propose for further work to investigate whether the increase of parameters from the model, such as d or K, will help to define a large differentiation between the two states. Recall, section 2.2, that d is the miRNA-mediated fold-change in the target mRNA degradation rate, and K is the half-saturation concentration or level of miRNA at which half of the maximum downregulating effect is achieved.

So, recapitulating, the aim of this section was to study the transition times based on the implication that the two states had to be different. However, the histograms for these particular systems at steady-states, show that only for the *prokaryotic* promoter there is a significance difference between protein levels of a system with miRNA and a system without. Therefore, it makes no sense to consider the other two cases in terms of studying the transition times.

Following these results, the most reasonable study of transition times for the particular models exposed in this project, will be to carry out a stochastic simulation for the transition states for regulatory systems that contain the *prokaryotic* promoter. (Figure 4.5 shows the results.)

Figure 4.5 represents the level of the target protein (number of molecules) versus time: transcriptional regulation (TF, blue line) and post-transcriptional (miRNA, red line), for the repression and recovery of systems with a prokaryotic promoter. The plot on the left represents repression of the system, and the plot on the right recovery from repression (or de-repression) for three representative simulations. An external stimulus

switches the regulation and the protein goes to a repressed (or recovered) state respectively. The post-transcriptional repression by miRNA shows a faster response of the system compared to the transcriptional repression of the same magnitude, principally in the repression of the system. However, the recovery of the system displays more noise than the repression.

Conclusions

The purpose of this study was to investigate the time taken for a protein to go from not repressed state to repressed one (and vice versa). In particular, we wanted to investigate whether transition time is different for different promoter types, and between transcriptional and post-transcriptional modes of regulation.

Transition times have been described by the mean, standard deviation and confidence intervals of 100 runs, and computed between two states of the system (or a protein): a state without repression and a repressed state. Different promoter types and different regulation modes have been considered, contrasting at the same time two types of regulations: the post-transcriptional regulation by miRNA and the transcription regulation by TF.

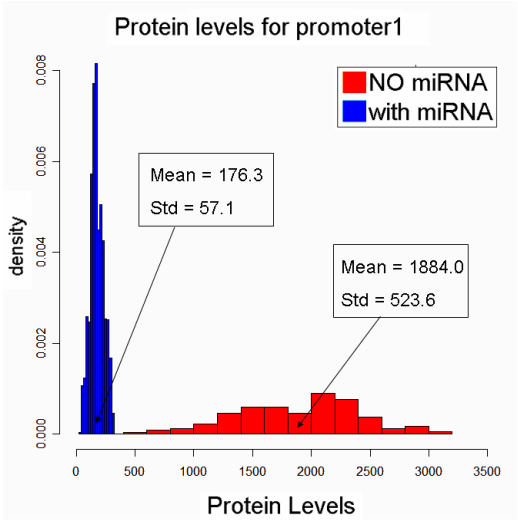
When the signal switches on, the repressor, the protein level decreases. The least noisy promoter is *prokaryotic*, but only when the system is repressed by miRNA that exerts mRNA degradation and inhibits protein translation (see Table 4.4). Therefore, the conclusions are just centered for the case of the *prokaryotic* promoter which is the only valid case for the further analysis. In this case We can conclude that the system regulated by miRNA shows a faster response to a signal than regulated by TF.

In the recovery of the system, when it changes from a repressed state to a state free of regulation (de-repressed), the prokaryotic promoter is again the least noisy (see Table 4.5). The system response is not particularly different between transcriptional and

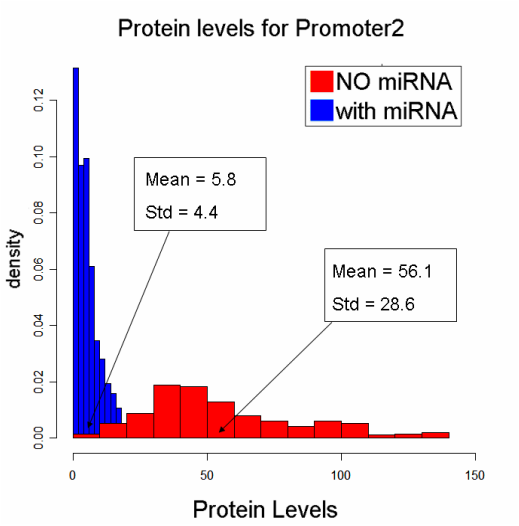
post-transcriptional modes of regulation.

Gene circuits, like the one with a *prokaryotic* promoter and miRNA repressing the system (or de-repressing) acting on both target mRNA degradation and protein translation is the least noisy. So, perhaps constructions like this have evolved to have very precise systems response in terms of transition time between different states.

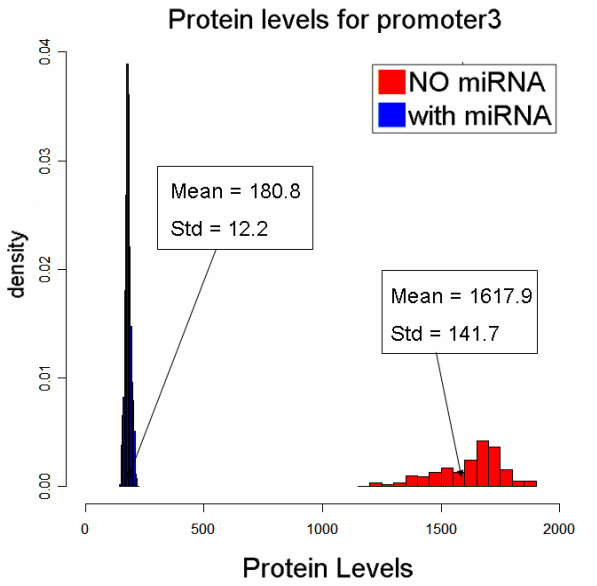
Figure 4.4. Histograms for the system with and without miRNA



(a)



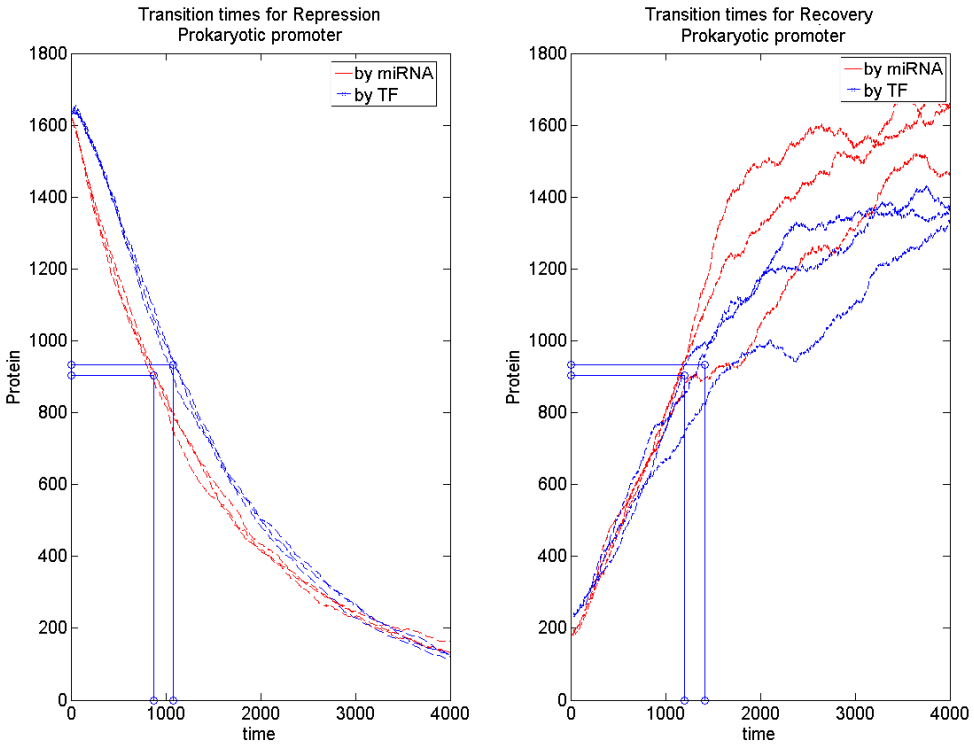
(b)



(c)

The Probability Distributions of the protein level at two steady-states of the system: one without miRNA (red histograms) and the other with miRNA being present (blue) for different types of promoters (*stable* (a), *unstable* (b) and *prokaryotic* (c)). The *prokaryotic* promoter is the only one that displays a good differentiation of the two states. NOTE: The means are not exactly equivalent to those in Table 4.6 because these plots have been obtained in a simulation run for a longer period of time, to have more data of the protein at steady-state.

Figure 4.5. Repression and Recovery of a single gene with a prokaryotic promoter



Response of the gene circuit with a prokaryotic promoter to the repression (plot on the left) and for the recovery from it (right plot). The level of protein of 3 simulations versus time is shown for two types of regulation: transcriptional by TF (blue) and post-transcriptional by miRNA (red).

4.2.3 In-silico experiment 3: Fluctuations in the signal

The following in-silico experiment is a study of the effect of the presence of miRNAs in the system, in particular whether it helps to filter out fluctuations in the signal.

Biological systems are affected by different external and internal factors including molecular signals, which results in the activation or deactivation of Transcription Factors (TFs) (various proteins that bind DNA and play a role in the regulation of gene expression by promoting transcription), and miRNAs. In our systems, the mRNA and miRNA production rates can be either increased or decreased by activator/de-activator signals (the cases considered here). However, the signals received by a biological system are not always real. Sometimes, the signals are just some noise, and even when signals are real, they can be affected by random fluctuations. These small effects can cause some unexpected changes in the sensitive protein synthesis and can be propagated further in the network.

In this study, fluctuation of the signal is considered to be switching on for a very short period of time and then turned off again. This switching on and off is repeated several times. The result of these fluctuations in activator signal is followed by short activation in mRNA and miRNA production. Later (after a short activator period of time) their rate constants are deactivated, i.e. return to zero (or to a initial very low values). In the case of a deactivator signal, the opposite effect is observed: the deactivation followed by activation of these rates (when the deactivator signal switches on and off respectively).

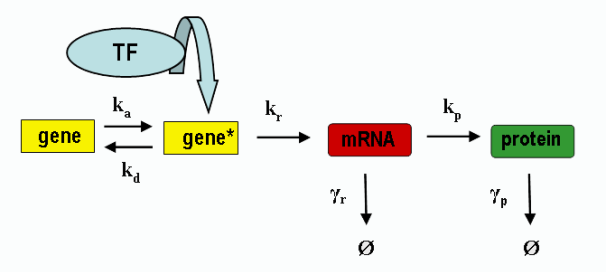
The goal of this simulation is to see whether activation/deactivation of a miRNA together with its mRNA target will filter out the noise in the signal. In fact, over 90 intronic miRNAs have been identified using bioinformatic approaches to date, but the function of the vast majority of these molecules remains to be determined (Rodriguez et al. 2004). We here investigate whether one of the roles of these intronic miRNAs is

to filter out the effect of the signal on the host mRNA.

Experiment

In this experiment two models with one target are compared: one model includes a miRNA and another does not. These are the models of *General Model* type and *General-miRNA Model* (with two variations). Firstly, production and decay of miRNA are considered when miRNA is present. Secondly, both models include a TF, that affects the transcription step binding to the promoter of the target gene and activating its transcription. The first model does not have miRNA, only TF, and the second model includes a miRNA which affects mRNA stabilization and protein translation. The signal regulates (activates or deactivates) both TF and miRNA. Figures 4.6 and 4.7 show these two models.

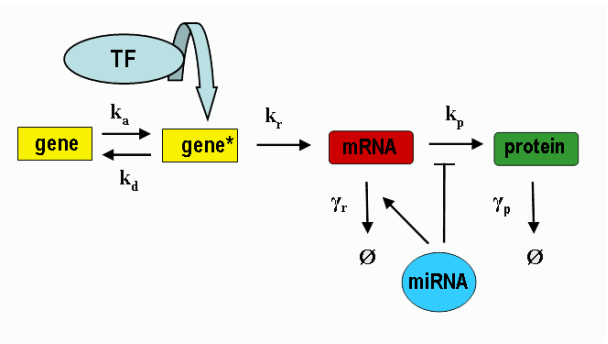
Figure 4.6. General Model with TF



General Model with a transcription factor. This model is like the General Model (Figure 2.2, explained in *Chapter 2*) but considering a TF, which regulates transcriptionally the gene affecting only the transcription rate, k_r.

Two types of signal are being considered, one being an activator and the other being a deactivator. In the case of a simulation with an activator signal, the gene regulatory circuit produces very little mRNA at rest (low k_r). When the signal arrives, the TF (for both models) and miRNA (considered only in the second model) become activated. In the case of a deactivator signal, TF and miRNA are active but become deactivated

Figure 4.7. General Model with TF and miRNA



General-miRNA model with a transcription factor. This model is like the General-miRNA model (Figure 2.4, in *Chapter 2*) but considering a TF. Model with two regulations: the transcriptional one from the TF, and the post-transcriptional from miRNA.

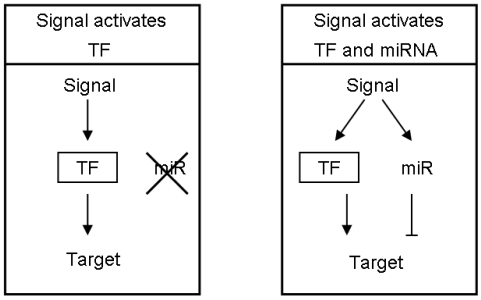
upon the arrival of the signal.

This study is not focused on the effect of different promoters, therefore only two models with an unstable promoter are simulated (check Table 2.1 for promoters).

Figure 4.8 and 4.9 illustrate the structure of the regulatory circuit for an activator signal and deactivator.

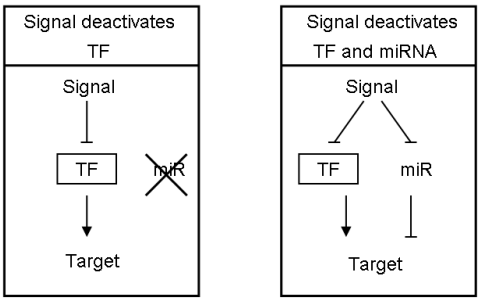
The appearance of the signal can be represented by a threshold function, which is switched on-off several times, over the time interval of interest. Figure 4.10 shows a graphical example of how the fluctuations of an active signal (top plot) affects the behaviour of the system (bottom plot), making the protein production from the second graph increases immediately after the several activations of the signal (peaks from the first plot).

Figure 4.8. Activator signal



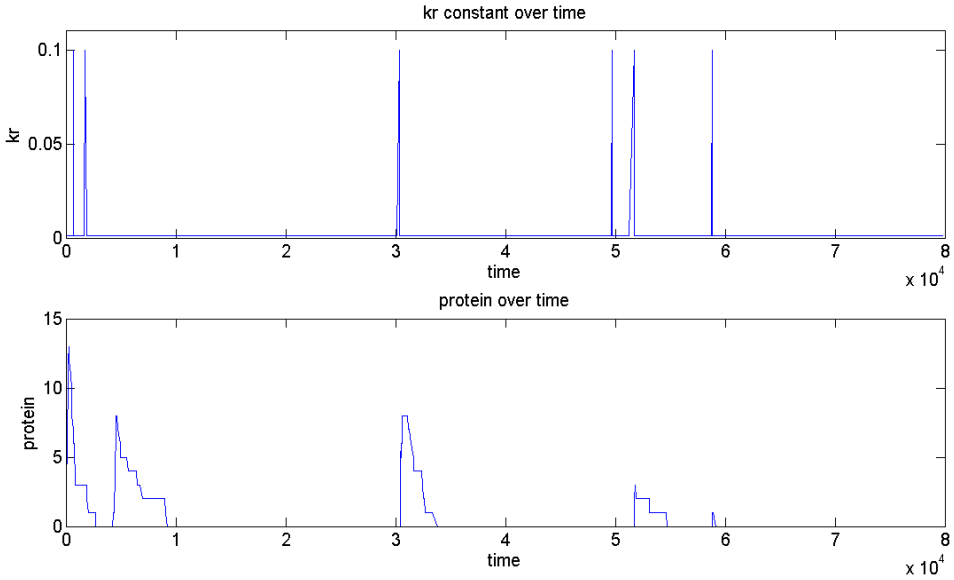
The architecture of the Activator Signal over the system. In the first diagram (on the left), the signal activates the cellular transcription factor (TF), which at the same time activates a downstream target gene. Simultaneously, in the second plot (on the right), signal activates both TF and miRNA (miR), where the transcription of a miRNA creates a safeguarding post-transcriptional channel that represses leaky transcript (mRNA).

Figure 4.9. Deactivator signal



The architecture of the Deactivator Signal over the system. The main structure over the system is the same as before with the Activator Signal (Figure 4.8), but instead of activating both TF and miRNA, the signal deactivates them.

Figure 4.10. Signal fluctuations and system response.



Representation of an Activation Signal. Top plot: the signal switching on-off several times over a period of time of 80,000 time units. Bottom: protein production for the system under the affect of this activation signal. The protein levels are increased very soon after the signal is activated but take a while to go to the background level.

Methodology

To carry on this study, numerical simulations using the Gillespie Algorithm for two models with one target (one with miRNA, Fig, 4.7, and the other without, Fig. 4.6) are carried out.

This algorithm runs over the interval of time of 80,000 time units with the parameters:
$k_a = 0.001$, $k_d = 0.1$, $\gamma_p = 0.0007$ (protein degradation), $\delta_m = 0.01$ (miRNA degradation). Then if the Transcription Factor is regulating the system $k_r = 0.1$ (mRNA production), if not, $k_r = 0.001$. The next three constants depend on miRNA, if miRNA is not present: $\gamma_r = 0.0039$ (miRNA degradation), $k_p = 0.0429$, $p_m = 0$ and if miRNA is present $\gamma_r = 0.0117$, $k_p = 0.0286$ and $p_m = 0.5$. A hundred samples for each system are simulated.

The signal appears at time zero ($t = 0$), that is when the mRNA production (k_r, rate constant) and the miRNA production (p_m, rate constant) are activated or deactivated. When the signal is off (or disappears), the parameters are set up to the original values. So, over the time line the signal is switched on and off several times (in Figure 4.10 where the signal is switched on-off 6 times) always being on for a very short time. The signal is computed independently in each simulation, being the number of on-off switches a random number. The time vector that denotes the times of the switches on-off of the signal is described by a piece of Matlab code in the *Appendix* B.

Results

Let us first consider the activator signal:

In this case (for both models), the mRNA and protein levels are low (or zero) as TF is not activated and, until the arrival of the signal, the system is repressed. Rates

of mRNA production and miRNA production in the second model increase upon arrival of the signal. This increase remains activate until the signal is switched off again. The production of the protein is still working but for a short while more (i.e. for the rest of molecules of mRNA that were already produced, and allowed the translation from mRNA into protein molecules). Several on-off of the signal happen along the time interval.

For the model with miRNA, the signal also triggers a production of miRNA that downregulates protein production. Once the signal is switched on, transcription of both mRNA and miRNA are activated simultaneously: the TF promotes the transcription of the target, while miRNAs exerts downregulating effect. Thereby the transcription of the target produced by the TF is counteracted by the miRNA repression of the target, as Figure 4.11 indicates by the relative frequencies of the protein levels of 100 simulations.

Let us now consider the deactivator signal:

For both models, the protein production is controled by TF and the signal deactivates it. This means that before the signal appears, the system is in a certain state producing mRNA and proteins, but the signal deactivates TF, resulting in a very low mRNA production, that consequently causes a decrease in protein production just for the time when the signal is on. Later, the signal is switched off again, and the system returns to its initial state. This is repeated several times depending on the random fluctuation generated by the piece of code introduced previously.

In the second model, at $t = 0$ the system is producing mRNA proteins and miRNAs at a certain level, until the arrival of the signal, wich is switched on, then both TF and miRNAs are deactivated. So, if a target is controlled by both TF and miRNA,

TF promotes the transcription of the target while miRNA represses it directly. The signal stops both regulations, i.e. the k_r transcription rate decreases to a very low rate and p_m miRNA production goes to zero. The protein level in the presence of miRNA is again much lower than in the model without miRNA, as it is shown in Figure 4.12 with the representation of the relative frequencies of the protein levels, for 100 simulations.

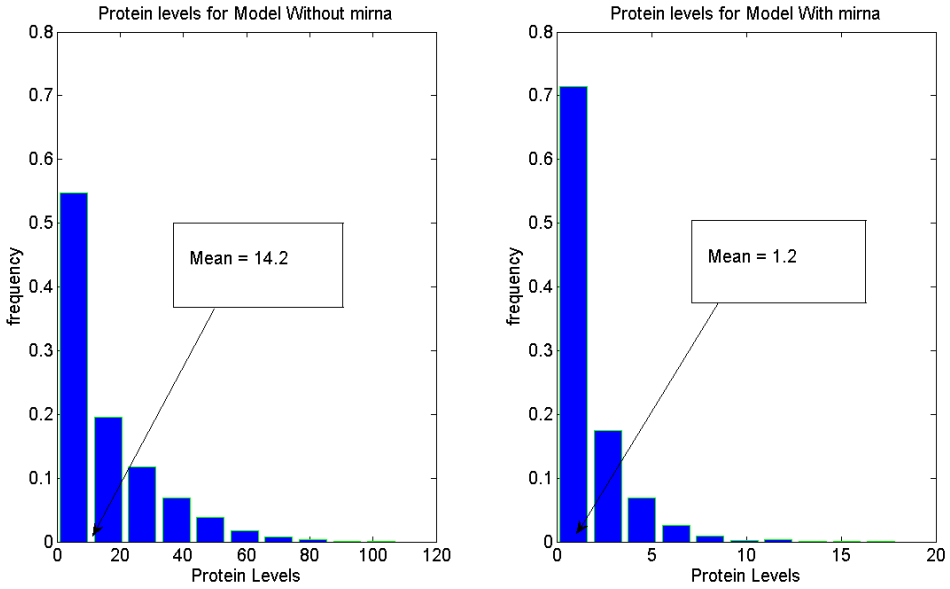
Conclusions

In this in-silico experiment, two basic regulatory circuits are tested in their ability to filter out external noise. The first model is a basic gene regulation model, while the second model also includes miRNA. Noise is presented as fluctuations in the fixed level of a signal. The signals considered are two, activator and deactivator, which are the responsible for the activation or deactivation of the TF and miRNA production rates.

We considered a first model, wherein a TF activates the process of transcription, and with the appearance of the signal the TF is activated or deactivated, depending of the type of signal. The second model, also includes a miRNA: the TF is an activator of the target, while the miRNA a repressor. Both regulators are activated or deactivated with the arrival of the signal. The signal is expressed by fluctuations always actuating in a short time interval, such as noisy signals or false activations in the system.

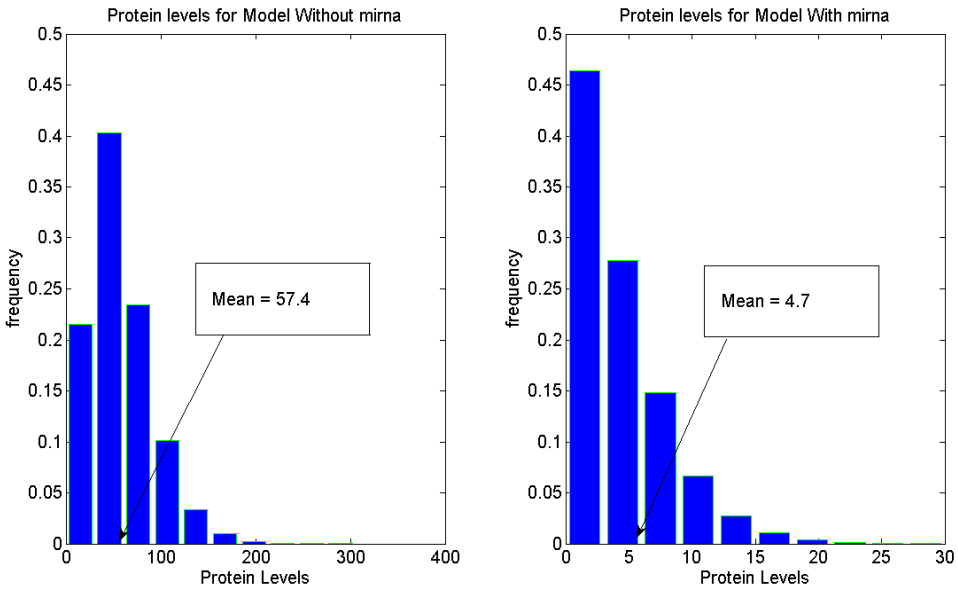
Finally, it has been shown that the system with miRNA is less susceptible to the presence of noisy signals than the system without miRNA. This is because the presence of miRNA helps to prevent the system of expressing protein when the signal is not real. This is due to the fact that miRNA represses the target.

Figure 4.11. Levels of protein for an activator signal.



The histograms of the protein levels for an activator signal switched on-off over an interval of time of 80,000 time units averaged over 100 simulations. Left plot is for a system regulated by TF only; right plot is for a system regulated by TF and miRNA. When the signal is on, only two rate constants are affected: $k_r = 0.1$, and $p_m = 0.5$; the signal is off, $k_r = 0.001$ and $p_m = 0$. The common rate constants for the two systems are: $k_a = 0.001$, $k_d = 0.1$, $\gamma_p = 0.0007$ and $\delta_m = 0.01$. Protein production is higher when miRNA is not present (left: mean = 14.2, γ_r= 0.0039, $k_p = 0.0429$), compared to the case wherein miRNA is present (right: mean = 1.2, $\gamma_r = 0.0117$ and $k_p = 0.0286$).

Figure 4.12. Levels of protein for a deactivator signal.



The histograms of the protein levels for an activator signal switched on-off over an interval of time of 80,000 time units averaged over 100 simulations. Left plot is for a system regulated by TF only; right plot is for a system regulated by TF and miRNA. When the signal is on, only two rate constants are affected: $k_r = 0.0011$, and $p_m = 0$; the signal is off, $k_r = 0.1$ and $p_m = 0.5$. All the other rate constants are the same than in figure 4.11. Protein production is higher when miRNA is not present (left: mean = 57.4), compared to the case wherein miRNA is present (right: mean = 4.7).

4.3 Analysis of the model with *Two Targets*

To carry on the different studies of this section, two methods that supposedly speed up the simulation of the Gillespie algotithm have been codified in this thesis, particularly, for the specific model of two targets. These methods are Langevin Equation and Tau-Leaping.

The code for Langevin (see *Appendix* D) was implemented from a previous code from Higham and Khanin (2007) for the case of a simplified protein monomer-dimer system. The code for Tau-Leaping (see *Appendix* C) has been codified in this project for this model.

4.3.1 In-silico experiment 1: Comparison within 3 numerical methods

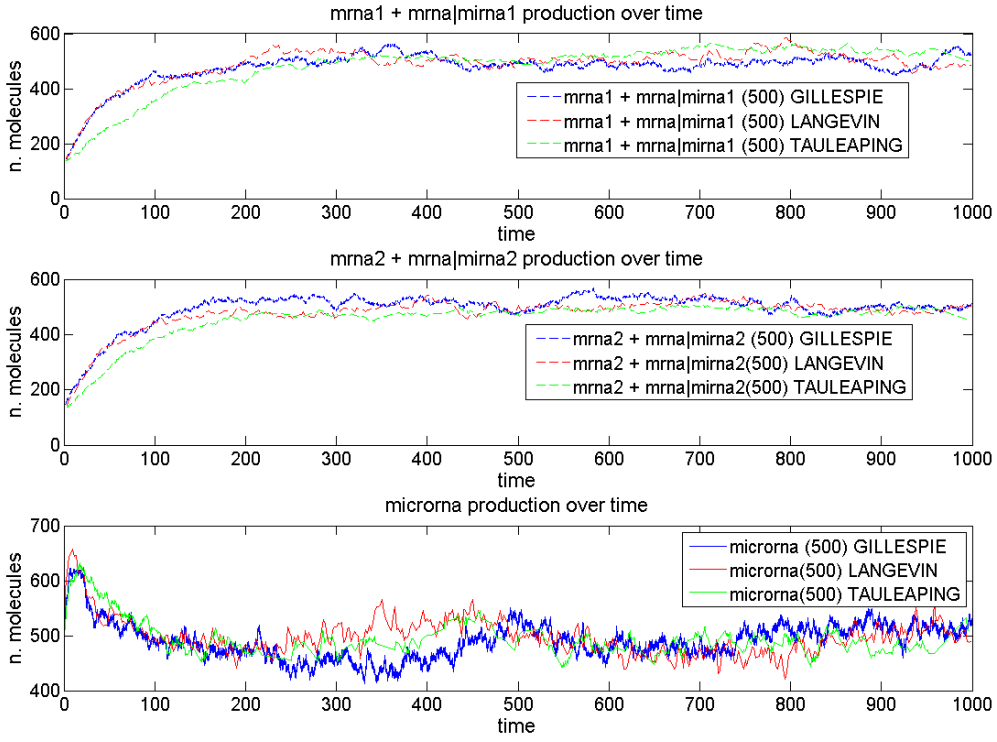
This section considers a model that studies the regulatory system with miRNA and two targets, the *two mRNA targets model* (see Figure 2.5 in section 2.2.3). Because two targets imply more variables and more reactions the first important thing to do, before carrying any other study, is to compare three methods (Gillespie Algorithm, SSA; Langevin Equation, CLE; Tau-Leaping) in terms of good approximations as well as in terms of low computationally costs. Thus, the aim here is to see whether the CLE and Tau-Leaping speed up the simulation of SSA, and also they are good approximations of the SSA.

About the mRNA targets, consider first that they are expressed at the same level in the absence and in the presence of miRNA. This implies that for the rate constants for two targets are the same (transcription rates: $q_1 = q_2$; mRNA degradation rates: $\delta_1 = \delta_2$; complex formation/dissociation rates: $\beta_1 = \beta_2$, $\beta_1^- = \beta_2^-$; complex degradation rates: $\delta_1^* = \delta_2^*$; and rates for miRNA turnover from complex to the pool rates: $\delta_1^* q = \delta_2^* q$).

Experimental Results

The simulations of *two mRNA targets model* are run by three stochastic methods (SSA, CLE and Tau-Leaping) and the results are compared and illustrated in Figure 4.13, which represents the levels of the two targets and the miRNA. Here mRNA is present in either free (unbounded) form or in a complex with miRNA (bounded), i.e. mRNA|miRNA complex. The levels of the target mRNAs are computed by the sum of both levels (free mRNA and bound mRNA). The CLE uses a step size of 1000/500=2, and the ϵ used in Tau-Leaping is 0.04. The plots show that the three methods give good approximated results because all the lines are very close over the time interval.

Figure 4.13. Two target model behavior simulated by three methods



The first plot shows the levels of free mRNA1 and the complex mRNA1|miRNA versus time ($tfinal = 1000$) by the numerical methods SSA (blue), CLE (red) and Tau-Leaping (green). Second plot is the same but it shows the levels of the second target. The third plot shows the levels of miRNA itself. The two targets are expressed at the same level, where the parameters are $q_i = 10$, $\delta_i = 0.01$, $\beta_i = 0.0001$, $\beta_i^- = 0.1$, $\delta_i^* = 0.05$, for i = {1, 2}, $q = 1$.

From Figure 4.13, it seems that both CLE and Tau-Leaping hold a good approximation to the SSA. A second numerical study on the means and the CIs of the means of the different reactants will confirm this conclusion.

Table 4.8 contains the numerical values of the CIs of the level means of the reactants: mRNA1, mRNA2 and miRNA. In fact, the reactants mRNA1 and mRNA2 are considered as the sum of both mRNAs, free and bound. It also contains the elapsed time of each method to run the simulation until steady-state (set up at the final time of 1000). As the two targets of the system have been set up at the same levels, the steady states for each target do not show a big difference between them. This means that miRNA can bind with the same probability to the first target and to the second one. The table also shows the time it takes for each method to simulate results, wherein it can be seen that Langevin is the fastest one.

Table 4.8. 95% CI for the reactant level means by three numerical simulations

	mRNA1 + mRNA1|miRNA	mRNA2 + mRNA2|miRNA	miRNA	Time
Gillespie	(491.6, 492.0)	(511.9, 512.3)	(498.1, 498.4)	9.10 min
Langevin	(509.2, 515.2)	(499.1, 503.1)	(480.7, 486.6)	0.16 sec
Tau-Leaping	(511.4, 517.2)	(478.6, 481.3)	(484.7, 487.9)	3.36 sec

This Table contains the CIs of the mRNA target levels and miRNA computed by three methods, and the time elapsed to run each of them. See Figure 4.13 for graphical representation.

Conclusions

A model with two targets has been simulated and with it, it has been shown that the Langevin Equation and Tau-Leaping are good approximations for the Gillespie algorithm (see Figure 4.13 and Table 4.8).

Moreover, the simulation of Langevin Equation accelerates Gillespie as does Tau-Leaping, but Langevin Equation is the fastest method (see Table 4.8). It should be noted and taken into account for this last assumption that the CLE and Tau-Leaping depend on the number of the step size used in the CLE and the ϵ used in Tau-Leaping.

The simulations also demonstrate that the probability that miRNA binds two targets with the same levels are equal (see Table 4.8 where the results for two targets at steady-state are the same).

4.3.2 In-silico experiment 2: Steady-states and noise by increase of miRNA

As it has already been discussed in this thesis, the expressions of the reactants of a gene regulatory circuit are noisy for nature. The factors that help to reduce this noise are investigated in this thesis, with the main focus on the effect that miRNA has over them. So the aim of this test on steady-states is to investigate whether different initial levels of miRNA have any effect on the amount of noise in the levels of target proteins.

This study is a continuation of the previous one (in section 4.3.1). The model studied is the *two mRNA targets model* and the rate constants and variables are also taken from the previous section (see caption in Figure 4.13 for the rate constants). The only thing that changes here is the initial number of miRNA, because it is now the focus of the study. Although, here we have the advantage that we can start running the simulations with Langevin or Tau-Leaping instead of the Gillespie Algorithm, because it has been proven in the previous study that these methods accelerate the SSA.

Experimental Results

Three miRNA levels are considered: 500, 1000 and 2000 (copies per cell), to simulate the two mRNA targets model the simulation used in this project is Tau-Leaping (with $\epsilon = 0.04$). This simulation is given by plots that contain the evolution of two mRNA targets over time and for three different miRNA levels. After that, the numerical values obtained for the noise of each reactant, in terms of Fano factors, are also given for three miRNA levels.

Figure 4.14 graphically shows the effect that miRNA levels causes on each of the two targets (targets with same levels). At low level of miRNA (500 copies per cell, top curve) the degree of downregulation of the targets is not as large as at higher levels of miRNA (for instance at 2000 copies per cell, bottom curve). This illustrates that the level of downregulation is determined by the level of miRNA itself.

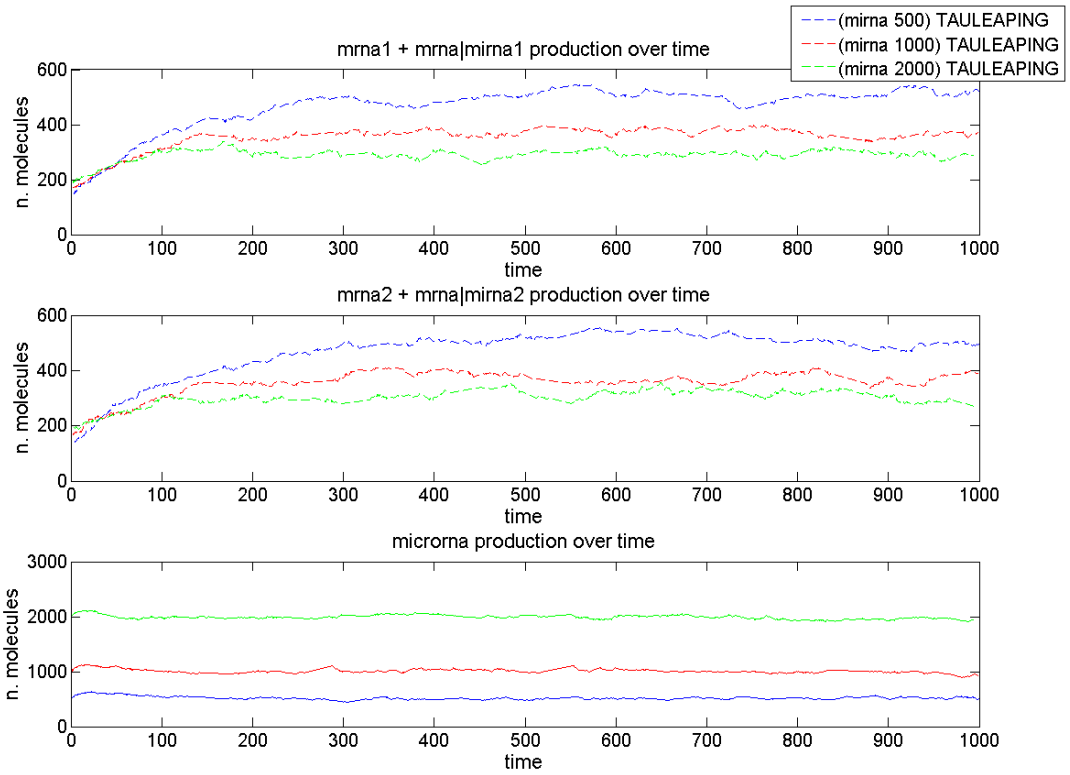
In addition, higher levels of miRNA reduce the noise in protein output (Table 4.9).

Table 4.9. Fano factor of the two targets and miRNA by levels of miRNA

	FANO FACTOR		
	500 miRNAs	1000 miRNAs	2000 miRNAs
mRNA1+mRNA1|miRNA	58.3	17.8	3.9
mRNA2+mRNA1|miRNA	61.4	19.1	5.6
miRNA	3.9	2.0	0.9

Fano factors of the levels of two targets and miRNA for three miRNA levels. The Fano factors of the two targets decrease with increase of miRNA levels: 500, 1000 and 2000 copies per cell. (Results by Tau-Leaping method, $\epsilon = 0.04$.)

Figure 4.14. Two targets of miRNA simulated for three different miRNA levels



The three miRNA levels are represented on the targets by different colours: 500 (blue), 1000 (red) and 2000 (green) and versus time. The first two plots represent the targets. Other parameters are the same as in Figure 4.13.

Conclusions

In this example of two targets of one miRNA, we have demonstrated that the effect of target downregulation on the targets depends on the level of miRNA. So, for high miRNA levels the system experiences a larger degree of target downregulation.

The presence of miRNA, and even more the increase in miRNA levels causes a decrease in the noise for all the reactants. This means that miRNA attenuates the noise the levels of target proteins.

4.3.3 In-silico experiment 3: Effect changing parameters on targets

External signals are often a source of changing some of the system parameters and rate constants. For instance, let be a system of two mRNA targets, and a signal affects the production rate of one mRNA targets. Then, we might expect that the miRNA are used differently by two targets, and in particular we might expect that the miRNA molecules are "used up" by the target that has a higher rate of production, and therefore higher level.

The aim of this study is then to analyse how the regulation by fixed initial number of molecules of miRNAs is affected by the levels of two mRNA targets, for different cases. More precisely, this experiment studies how the first target mRNA1 is consequently affected when the level of the second target mRNA2 is increased by an external signal. To increase the level of the second target, the production rate, q_2 (see the system of reactions 2.11, *chapter* 2), is increased. At the same time, different initial numbers of miRNAs -low, medium and high- are set up to give a global view of how the levels of the mRNA targets are affected by these two variables. After this, a third variable is introduced in the study, the miRNA-mediated fold-change in the target mRNA degradation rate d (see Eq. 2.6 in *chapter* 2), to analyse how it affects the changes experienced by

the system.

To carry on this simulation the model that is used for this scenario is the *two mRNA targets model*, the same model as in the two previous experiments. Therefore, the parameters and initial values will be considered the same as before until further notice. Here the Langevin equation or Tau-Leaping method are used to simulate the system.

Experimental Results

This study uses the simulation of Langevin Equation with a step size of $L = 1$ for three different miRNA levels and an increase on q_2, production rate (transcription rate) of the second target. So, the parameters of the model remain the same, except for an increase of q_2, where $q_2 = \{1, 10, 20, 40, 60, 80, 100\}$, and the three miRNA levels are 500, 1000 and 2000 (miRNA copies per cell), obtaining the results displayed in Figure 4.15.

In Figure 4.15 it is shown the effects that q_2 produces on the first target. In general, it can be seen that the increase of production of the second mRNA target, q_2, implies an increase on the first target (the lines on the plot have mainly a positive slope everywhere, and this happens for all three levels of miRNA). The effect is particularly pronounced when levels of miRNAs are low (for example the upper line in the plot that also manifests the larger slope) and it indicates that the first target has been more affected by the increase on q_2, than for higher miRNA levels.

At this stage of the experiment, another factor comes up as a point of interest in this study. It is the fold-change parameter of mRNA degradation denoted by d_i, for $i = \{1, 2\}$ (see Figure 2.5, Chapter 2). This parameter is the factor that accelerates the degradation of mRNA, or in other words, the parameter that participates in the rate constant of the decay of the mRNA|miRNA complexes as $(d_i\delta_i^*)$ for $i = \{1, 2\}$. Clearly, the higher d_i is the quicker mRNA|miRNA complexes decay, resulting in larger

target downregulation. For large d_i, the targets are strongly affected by miRNA, while weakly affected targets have small values ($d_i \leq 1/2$). The question to answer is how this parameter affects the extent of the target downregulation?

This question can be split up into two more steps: one that studies the changes of the system over d_1 while d_2 remains constant (at the value of 5), and the other, which changes d_2 and leaves d_1 at a constant value ($d_1 = 5$). We want to study whether indirect effect of one target caused by the changes in the other target depends on the strength of miRNA-mediated downregulation by each of them. For example, if a target that is regulated transcriptionally (target 2) is a weak miRNA target, then it is likely that the increase of it level on the other strong miRNA target will be small, as not much miRNA will be used up. Alternatively, effect of transcriptionally regulated strong target on a weaker miRNA target is more significant.

Figure 4.16 displays the first process with four subplots for the following values of $d_1 = \{3, 5, 10, 20\}$. The first plot is for $d_1 = 3$, the second (top on the right plot) is for $d_1 = 5$, the third (bottom on the left plot) is for $d_1 = 10$ and the fourth is for $d_1 = 15$. The same study but in the opposite way is displayed in Figure 4.17, analysing the changes on d_2 instead of d_1, and observing now which dependence d_2 has with the downregulation of the targets.

Thus, Figure 4.16 and 4.17 show that the miRNA-mediated fold-change in the target mRNA degradation rates d_i, i ={1,2}, is significant effect on downregulation of the targets. Principally the first figure (Figure 4.16) demonstrates that for a strong mRNA target (target1, high d_1) the increase in the level of the second mRNA target (increase of q_2) and by different miRNA level is much more downregulated than for a weak miRNA target (target1, small d_1).

Meanwhile, the second set of figures (Figure 4.17) demonstrates that the indirect

miRNA-caused changes in the level of the first target (mRNA1) by increasing the production rate of target2 (increase of q_2), depends on the level of downregulation of this target (d_2). This means that when the level of target2 are increased due to some external factors (as for example here that its transcription rate becomes higher) and if target2 is strong (d_2 higher than d_1), then it will require more miRNA to be used up for the second target than target1, meaning that it will be less miRNA left for target1, whose level will also become higher. If however target2 is weak (d_2 smaller than d_1), then increase in target2 level will have smaller influence on the level of target1.

Therefore it has been shown that the effect of one target on another target via miRNA regulation of both of them depends on the strength of miRNA downregulation of each target (d_1 and d_2 parameters).

Conclusions

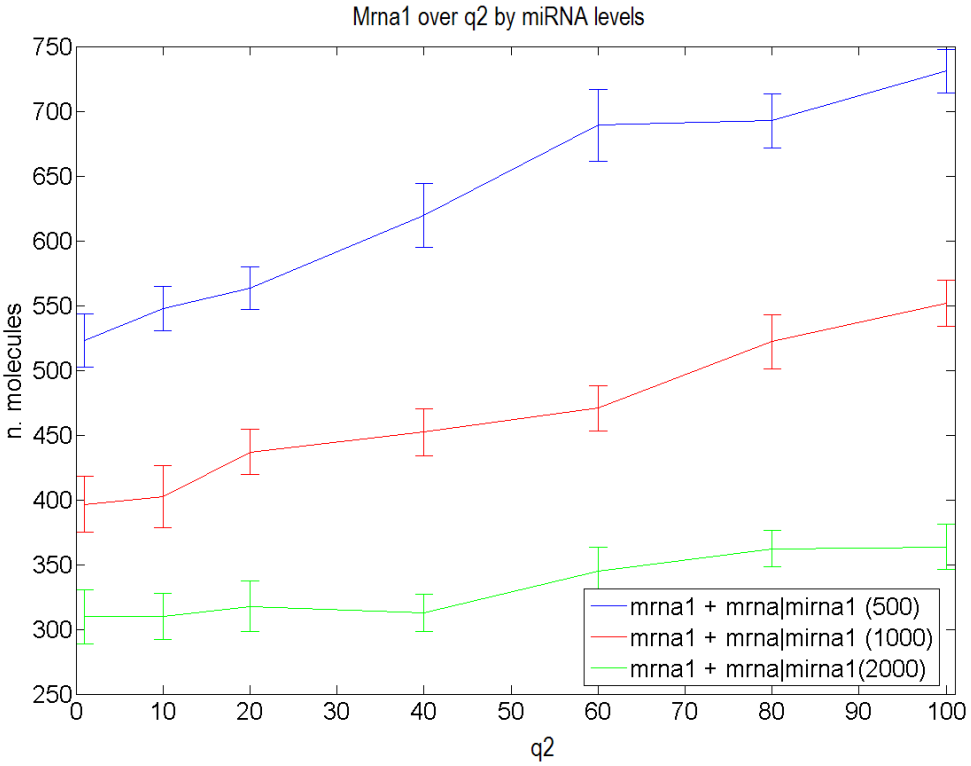
By simulating stochastic model, with two targets, it has been shown that the increase in the level of one of these targets indirectly affects the level of the other. Similar results have been obtained by (Khanin & Higham 2009) using ODE-type model. For instance, as it is shown in this experiment, the increase of the rate constant of one target production implies the increase of the level of the other target. This is caused by the indirect miRNA-mediated effect of the first target. This means that for a fixed level of miRNA, when the second target is increased, the first target is less downregulated as miRNA, that is common to both of them, is used up by the second target.

At high miRNA levels, the immediate effect of increase of one target by the increase of another one is less pronounced. This implies that for an unlimited number of copies of miRNAs per cell, this effect could be negligible. However, for a realistic situation wherein cell has only a limited highly controlled number of miRNA molecules,

this effect should be taken into account, in particular, while interpreting various experimental datasets, including microarrays that only measure expression levels of mRNAs.

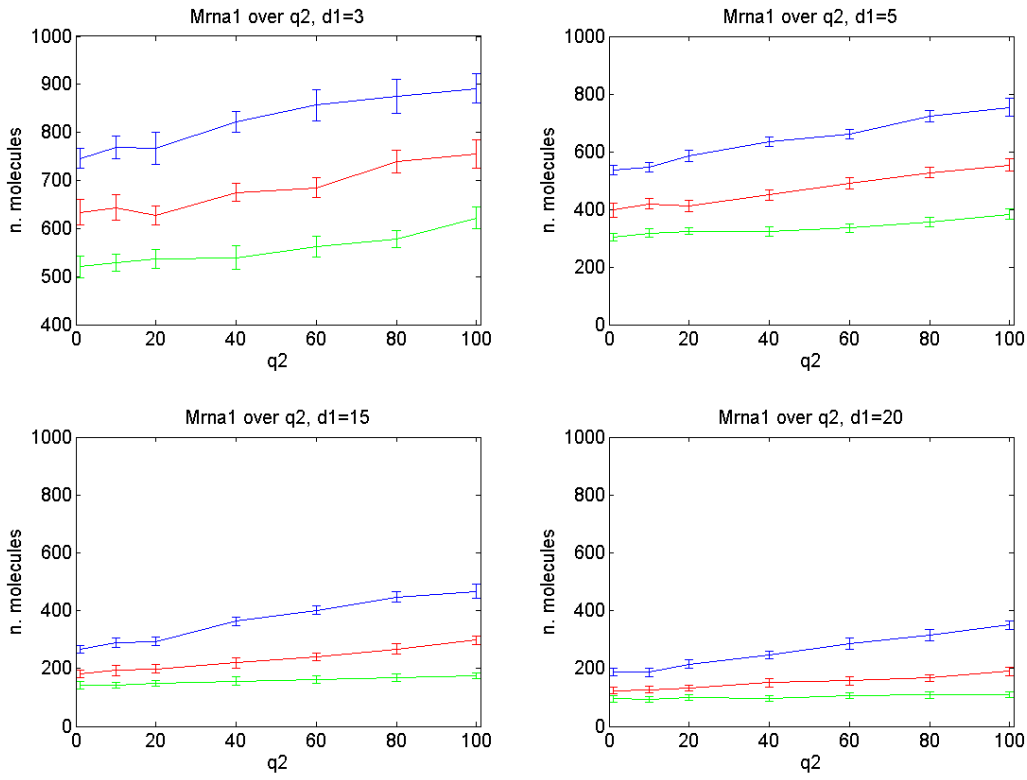
In a situation where the second target is increased (q_2) and the number of miRNA copies per cell is limited, the strong miRNA-regulated (higher d_1) target will be more downregulated. Also, if the fold-change of the second target (d_2) increases, for a low miRNA level the levels of mRNA1 increase as well, meanwhile at higher levels (i.e. bigger than 2,000) the levels of the first target decrease, being again downregulated.

Figure 4.15. Two targets of miRNA simulated for three different miRNA levels



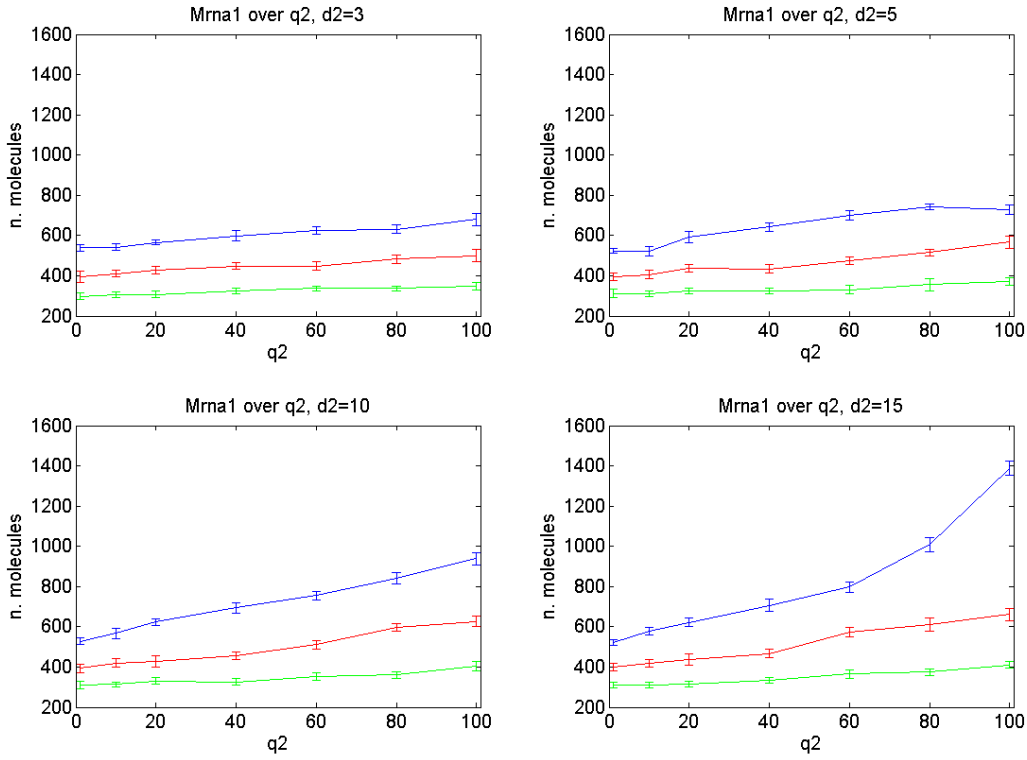
Errorbar plot of the effect of the transcription rate q_2 of mRNA2 for different miRNA levels on the *target 1*. The levels of miRNA 500, 1000 and 2000 are represented by blue (upper line), red (mid line) and green (lower line) respectively. Here $q = 0.9$ and other parameters as in Figure 4.13.

Figure 4.16. Two targets of miRNA simulated for different miRNA levels, over the increase of q_2, and depending on d_1



Effect of the fold-change rate d_1 of the decay of mRNA1, for different miRNA levels and different q_2. All parameters are as in Figure 4.15, except for q_2 and d_1.

Figure 4.17. Two targets of miRNA simulated for different miRNA levels, over the increase of q_2, and depending on d_2



Effect of the fold-change rate d_2 of the decay of mRNA2, for different miRNA levels and over a increase in q_2. All parameters are as for Figure 4.15, except for q_2 and d_2.

4.4 Analysis of the model with *N Targets*

Each miRNA regulates hundreds of targets. In addition, there are several hundred miRNAs that act cooperatively. To study complex behaviour of the cell, one needs to do large realistic simulations that involve many regulators (TF and miRNA) and their targets. Here the initial steps towards this type of analysis are described together with some ideas for further work on studying these interactions.

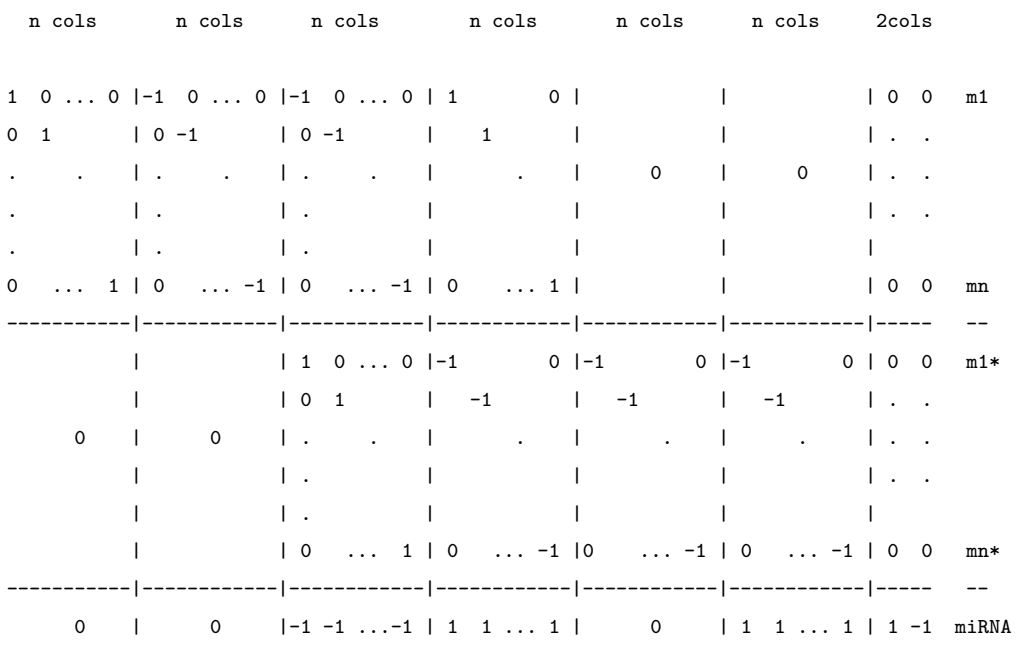
In this thesis we prepared the code for a model with N targets to be run by the stochastic simulation of Gillespie to be ready for studies related to this model. To be sure that the code was right, we took $N = 2$ in the N targets code and we compared the results to the two targets code.

4.4.1 In-silico experiment 1: Comparison between *2* targets Model and *N* targets

In order to set up the code of N targets and simulate its behaviour over time, one has to come up with a model distribution for system parameters, such as rate constants and initial levels. The rate constants will be sampled from value intervals taken from experimental observations. The initial values for mRNAs levels will be sampled from a probability distribution, specifically from the Pareto distribution. MiRNA is still being a variable introduced by the user.

One of the cumbersome steps of this analysis is to define algebraically all the reactions that can occur in a system of N targets. This is given by the stoichiometric matrix, ν_{ij}, where the rows represents the different reactants and the columns the reactions. The reactions expressed in this matrix will be called one to one, randomly, by Gillespie to produce the simulation, where each column is called the j^{th} reaction that fires each time. The stoichiometric matrix is included below and it demonstrates how a large system like this works, and how many reactions take place.

The stoichiometric matrix for a system with N targets looks as follows:



In this matrix each row represents a reactant that participates in the model of N targets. There are *2N + 1* reactants (rows): $m_1,\ldots,m_N$, $m_1^*,\ldots,m_N^*$ and miRNA. Here m_i are the mRNA targets and m_i^* are the mRNA-miRNA complexes, for all i=$\{1,\ldots,N\}$. Each column represents a different reaction. There are *6N + 2* columns, where the first six groups of N reactions describe the following: mRNA production, mRNA degradation, mRNA-miRNA complex formation, mRNA-miRNA complex dissociation, mRNA-miRNA complex degradation, mRNA-miRNA complex return to the pool, and the last two columns refer to the production of miRNA and its degradation.

In this study the main objective is to simulate the N targets model for $N = 2$ (as

described above), to make sure that it works properly. One way to do this is to simulate first the code of N targets, sampling all the initial mRNA levels and rate constants, and later apply the Gillespie simulation to study the evolution of the system over time. Sampling gives different values of levels and rate constants for each run. The next step is to compare the N $(= 2)$ targets with the two targets code: the two targets code can be run with the rate constants obtained by sampling in the N-code.

A simpler way to compare the equivalence of these two codes is to go the opposite way. This means that we start with the two targets code, for which initial values and rate constants have been introduced from the previous *in-silico experiment 2*, and these values are used in the code of N targets. Then, we run the simulation for the general code of N targets, setting $N = 2$. The system behaviour obtained is very similar to the results for two targets in *section 4.3.2*, because the constants and initial values are the same.

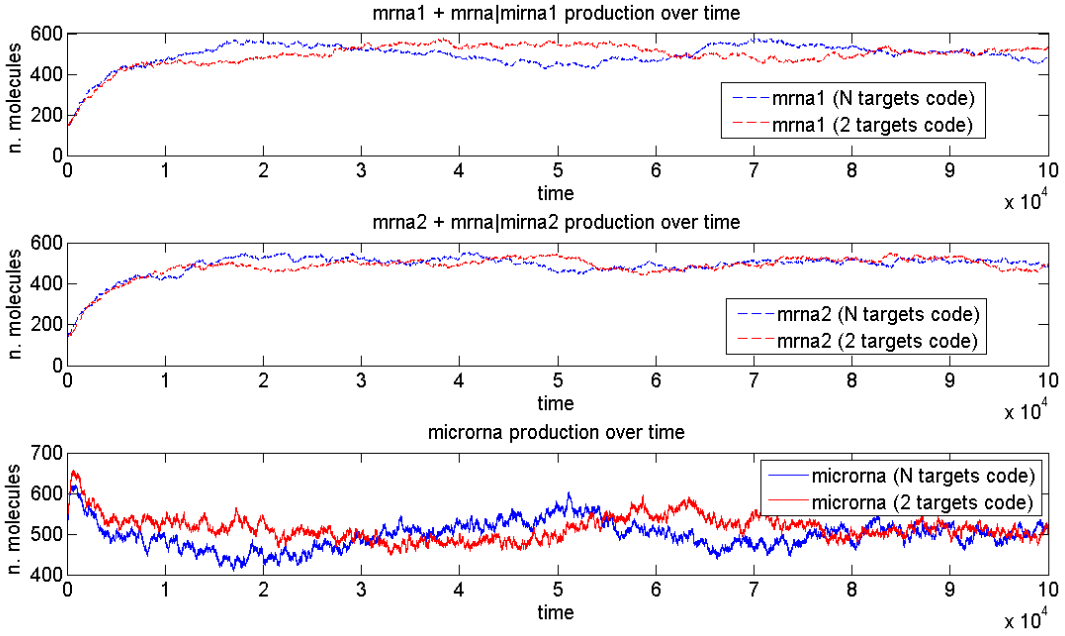
The results obtained with the N targets code are shown in Figure 4.18 and they are similar than in Figure 4.13, giving the concordance between the behaviour of the two systems. Thus, it is confirmed that the dynamic stochiometric matrix construction for N targets works correctly. The plots show similar results for both models

We conclude this analysis after this comparison between the two targets code and the N targets code with $N = 2$.

For further work, here there are two ideas about how to study these interactions:

- One way to do it could be taking the N targets, classifying them in three groups, for example by amount of molecules of mRNA (low, medium, high). The question is to study whether the extent of target downregulation is determined by the target levels or only by kinetic parameters.

Figure 4.18. N target model versus 2target model, setting N=2



Comparision of a N targets model (whith N=2) (blue line) and the two targets model itself (red line).The constants are: $q_1 = q_2 = 10$, $\delta_1 = \delta_2 = 0.01$, $\beta_1 = \beta_2 = 0.0001$, $\beta_1^- = \beta_2^- = 0.1$, $\delta_1^* = \delta_2^* = 0.05$, $p_m = 5$, q = 1.

- Another in-silico experiment is to study the N target interactions by fixing only one target and observing how differently it is affected by miRNA depending on the rest of the group.

Chapter 5

Conclusions and Discussion

This thesis deals with studying stochasticity in gene expression and gene regulation. Principally, the thesis concerns with the study of noise in protein production for different types of gene promoters and the effect that a new class of post-transcriptional gene regulators, microRNAs (miRNAs), have on the amount of noise in the protein output. Various *in-silico* experiments of gene regulation model were extensively simulated using three stochastic numerical methods.

By simulations of gene-regulatory circuits that involve miRNAs, we effectively confirm the hypothesis formulated at the beginning of this thesis, that miRNAs tune down gene expression and attenuate the variation or noise in the level of protein.

The different scenarios where these models have been involved and the statistical techniques worked through have been very suitable to determine this conclusion. The following sections describe the conclusions from each experiment more accurately.

5.1 Assessing the methodology

Three different stochastic numerical methods have been tested in this dissertation. These are Gillespie Algorithm, Langevin equation and Tau-leaping method. The Gillespie Algorithm is a computationally expensive method. Both chemical Langevin Equation and Tau-Leaping method are supposed to accelerate the Gillespie Algorithm. We found that the Chemical Langevin Equation accelerates it, while Tau-Leaping does not always speed things up. The latter depends on the number of molecules and values of rate-constants. Another observation is that considering the Tau-Leaping method with a small accuracy error (i.e. nearly zero) the approximation to the Stochastic Simulation Algorithm gets much better but, instead, the simulation is very expensive computationally. Our simulations confirm that both Chemical Langevin Equation and Tau-Leaping methods give good approximations to the results obtained by the Stochastic Simulation Algorithm (or Gillespie Algorithm) for gene regulation model.

The conclusions about the numerical methods are evidenced with the example of simple dimerization, where the results match for all three methods. A second example of a gene expression model of two mRNA targets, implying first-order and second order reactions, also verifies that simulations of Langevin Equation and Tau-Leaping not only accelerate the Gillespie Algorithm Simulation, but also the approximations are very close to the exact SSA procedure.

Other two analytical methods, the Probability Generating Function and the method of Gadgil *et al* (2005), have been examined. Both methods yielded the same theoretical results for the formulas of the noise of mRNA and protein, given by the Fano factor, for the Simple Model of gene expression. Moreover, we developed the formulas of noise for the reactants of a General Model of gene expression, with the analytical method, Gadgil *et al*. This representation of the theoretical results has been contrasted with the results of numerical simulation (see first example of analysis of noise, section 4.2.1)

giving a very good approximation between the two different types of methodology, analytical versus numerical.

5.2 Conclusions from one target model

5.2.1 Steady-state & noise by promoters

Our extensive simulations of post-transcriptional gene regulation by miRNAs reveal that miRNA contribution to attenuating the noise in protein output depends on the type of gene promoter. In other words, miRNA effect on reducing noise in protein levels depends on the rate-constants of promoter switching between active and inactive states. The so-called *stable* promoter that results in the highest among three promoter types protein output is the noisiest, while the *prokaryotic* promoter is the least noisy and produces medium levels of proteins. *Unstable* promoter produces smallest levels of protein and has medium noise. The general conclusion is that the presence of miRNA not only reduces the levels of target mRNA and protein, but also helps to tune down the noise in protein levels.

5.2.2 Transition Times

A number of studies involving regulation of a gene circuit by miRNA have been carried out in this thesis and interesting results obtained. In one of the studies, we looked at the system's response to a sudden change of miRNA regulator and compared it to the response due to change of transcription factor regulator. In the first case, the system is controlled on post-transcriptional level, and in the second case, similar changes in protein levels are caused on transcriptional level, so that transcription factor is a repressor. Two types of scenarios were considered: an external signal causes increase in the repressor levels causing the system to get repressed, while another scenario considers a signal that decreases the repressor causing the relief of the repression (de-repression). It emerges that the gene regulated at post-transcriptional level by miRNA shows a faster

response to a repressor signal than a system that is regulated by transcription factor. Furthermore, the transition times for a miRNA-regulatory circuit are less noisy than for a system regulated by transcription factor. In other words, proteins regulated by miRNA have a fast and accurate (less noisy) response to its repressor levels than those proteins with similar parameters that are regulated by transcription factors. We can speculate that essential proteins whose response to certain external signals require precise timing for being switched on and off are regulated on post-transcriptional level by miRNA that transfer these signals to these proteins. These results hold for prokaryotic promoter. Further simulations with different rate-constants, such as the fold-change of mRNA degradation and the half-saturation concentration of the protein translation step, are needed to check whether similar types of conclusions hold for at least some parameter ranges for the stable and unstable types of promoters.

5.2.3 Fluctuations in the signal

Another interesting observation from our *in-silico* experiments is that miRNAs may help filtering out noisy external signals. In particular, it has been shown in this thesis that systems regulated by a miRNA that is switched on/off by an external signal are less noisy on protein level than those systems that are regulated purely on transcriptional level. In other words, miRNA-regulated systems wherein miRNA is a sensor for external signal are less affected by signal fluctuations than systems regulated by transcription factors. We conclude that miRNA prevents the system to express protein when there is no real signal. Indeed, nearly half of human miRNAs are transcribed from protein-coding genes (intronic miRNAs), and some of these miRNAs downregulate their host genes by targeting their 3'UTRs. We propose that such construction has evolved for essential proteins to filter out external noise.

5.3 Conclusions using two targets models

Most simulations in this thesis deal with a single target of a miRNA. Further studies should involve considering multiple targets, and their interdependence. It should be noted than in more complex systems, such as in the circuits with two targets, the level of miRNA becomes a crucial factor as well as other parameters. Systems with low level of miRNA exhibit so-called target cross-talk, wherein changes in one target (on transcriptional level) are reflected in the changes of the other target(s) via indirect role of their common miRNA. Further work in this direction is needed to incorporate multiple targets with different topologies.

Appendices

Appendix A

Tau-Leaping Code for a dimer

```
% TAULEAPING_DIMER.M
%
% Simple implementation of the Tau-Leaping Method on the simplified protein
% monomer-dimer system.
%
% This version runs the model up to time tfinal
% for N times in order to evaluate the confidence interval
% for the endpoint 1st and 2nd moments.
%
% Adapted from DJH's SIAM Review code for the case of a
% simple dimerization model.
%
% Martina Marba (2008)


clear all
clc

%stoichiometric matrix
V = [1 -2 -1  0;  %monomer
     0  1  0 -1]; %dimer
```

```
%%% Parameters and Initial Conditions %%%
N = 1e4;       %runs
c(1) = 5;      % rate const for production of protein monomer
c(2) = 0.01;   % rate const for dimerization
c(3) = 0.1;    % rate const for decay of monomer
c(4) = 0.01;   % rate const for decay of dimer
tfinal = 20;

%%% This needed specific for Tau-leaping %%%
epsilon=0.04;     %Gillespie et al, 2007
d=size(V);
N_Species = d(1); %number of Species
M = d(2);         %number of Reactions
%%% End specific for Tau-Leaping %%%

for nn = 1:N
  X = zeros(2,1);
  b = zeros(M,N_Species);
  xi = zeros(N_Species,1);
  X(1) = 10;
  X(2) = 2;
  Xold = X;

  t = 0;
  count = 1;

  tvals = zeros(tfinal,1);
  begin=(length(X)+1);
  Xvals = zeros(begin,tfinal);
  tvals(1) = 0;
  Xvals(:,1) = [t X']';

  while t < tfinal
  %the propensity functions can't be negative
  a(1) = c(1);                %production of protein monomer
  a(2) = c(2)*X(1)*(X(1)-1)/2; %dimerization
       if X(1)<=1
           a(2)=0;
       end
  a(3) = c(3)*X(1);           %monomer decay
```

```
a(4) = c(4)*X(2);            %dimer decay
asum = sum(a);

%%% Matrix 'bij' (partial derivatives) by hand.
b(1,1)= 0;       %da1/dx1
b(2,1)= (c(2)*(X(1)-1)/2)+ c(2)*X(1)*(1/2);  %da2/dx1
b(3,1)= c(3);    %da3/dx1
b(4,1)= 0;       %da4/dx1
b(1,2)= 0;       %da1/dx2
b(2,2)= 0;       %da2/dx2
b(3,2)= 0;       %da3/dx2
b(4,2)= c(4);    %da4/dx2

% vector 'xi'.
for j=1:M
   xi(:,j) = a(j)*V(:,j);
end
xi=sum(xi,2);

vector = zeros(1,M);
for h=1:M
   vector(h)= epsilon*asum/abs((xi(1)*b(h,1)+xi(2)*b(h,2)));
end
index=find(vector==Inf);
vector(index)=[];



%Now we find 'tau'
tau = min(vector);

%CONDITION: To control that tau is not smaller than Gillespie's tau
if tau<=2/asum
   disp(['Is better apply Gillespie at this point. t= ',num2str(t)])
   %pause
end
%END CONDITION

k=zeros(M,1);
lambda=zeros(N_Species,1);
for j=1:M
    k(j)=poissrnd(a(j)*tau);
```

```
      lambda= lambda + k(j)*V(:,j);
  end

  % CONTROL that no reactant population will be driven negative
  if ((lambda(1)+X(1)) <0 || (lambda(2)+ X(2)) <0)
      tau=0;
      lambda=zeros(N_Species,1);
  end


  t = t + tau;
  Xold = X;
  X = X + lambda;


  count = count + 1;
  Xvals(:,count) = [t X']';

end

  Pfinal(nn) = Xold(1);   % state of P at time tfinal.
  P2final(nn) = Xold(2);  % state of P2 at time tfinal.

end

Pmean = mean(Pfinal);
Pstd = std(Pfinal);
Pconf = [Pmean-1.96*Pstd/sqrt(N),Pmean+1.96*Pstd/sqrt(N)]
Psqmean = mean(Pfinal.^2);
Psqstd = std(Pfinal.^2);
Psqconf = [Psqmean-1.96*Psqstd/sqrt(N),Psqmean+1.96*Psqstd/sqrt(N)]

P2mean = mean(P2final);
P2std = std(P2final);
P2conf = [P2mean-1.96*P2std/sqrt(N),P2mean+1.96*P2std/sqrt(N)]
P2sqmean = mean(P2final.^2);
P2sqstd = std(P2final.^2);
P2sqconf = [P2sqmean-1.96*P2sqstd/sqrt(N),P2sqmean+1.96*P2sqstd/sqrt(N)]
```

Appendix B

Code of the Switching Times of the signal (Fluctuations in the signal)

% timevectorcode.m (29.11.08)
%
% Piece of code for the time vector that denotes the random number of on-off switches % of a signal. This code creates first a random vector, that represents a fluctuation % of a signal (see the two first *if* conditions), and later it associates the % rate constants that the system takes depending on the signal being on or off. % In particularl here, the signal is deactivator.
%
% Martina Marba 2008

```
N_parts = ceil(rand*1000); %Number of times that signal switches on-off
tduring = rand*30;         %Durability of the signal.
L = tfinal/N_parts;        %Points of the vector where signal switches on
if (tduring > L)
    fprintf(1, 'PROBLEM: variable tduring > L, and this is not manageable');
```

```
end
tgaps= L-tduring;          %L=tduring + tgaps
if (tduring >= tgaps)
    while (tduring >= tgaps)
      tduring=rand*20;
    end
end

while t < tfinal
    vector=zeros(1,2*N_parts+1);
    vector(1) = 0;
    for i=2:2:2*N_parts
        vector(i)= ((i-2)/2)*L+ tduring;
        vector(i+1)= (i/2)*L;
    end
    for i=1:2:2*N_parts
        if (vector(i) <= t && t < vector(i+1))    %parts where signal is on
            c(3)=1e-3;  %kr
            c(7)=0;     %pm
        end
        if (vector(i+1) <= t && t <= vector(i+2)) %parts where signal off
            c(3)=0.1;
            if (mir==true)
              c(7)=0.5;
            end
        end
    end
end
```

Appendix C

Tau-Leaping Code for two targets model

```
% tauleaping.m (10.11.08)
%
% Simple implementation of the Tau-Leaping Method for two mrna targets.
% This function returns a matrix with the values of the reactants over
% time and the time vector. The inputs are V, X, c and tfinal, where
% V = stochiometric matrix, X = initial values of the reactants, c = vector
% with the constants and tfinal. (Equal inputs for all methods)
%
% NOTE: b(i,j) and the variable 'vector' different for each model (by hand)
%
% Martina Marba 2008


function Xvals=tauleaping1(V, X, c, tfinal)

%%% This needed specific for Tauleaping %%%
epsilon=0.04;     % Gillespie et al, 2007
d=size(V);
```

```
N_Species = d(1); % number of Species
M = d(2);         % number of Reactions
%%% End specific for Tau-Leaping%%%

b = zeros(M,N_Species);
xi = zeros(N_Species,1);
Xold = X;
t = 0;
count = 1;

tvals = zeros(tfinal,1);
begin=(length(X)+1);
Xvals = zeros(begin,tfinal);
tvals(1) = 0;
Xvals(:,1) = [t X']';



while t < tfinal
%the propensity functions can't be negative
    a(1) = c(1);            %production of mRNA1
    a(2) = c(2)*X(1);       %decay of mRNA1
    a(3) = c(3);            %production of mRNA2
    a(4) = c(4)*X(2);       %decay of mRNA2
    a(5) = c(5)*X(1)*X(5);  %formation of complex1 microRNA:mRNA1
    a(6)  = c(6)*X(2)*X(5); %formation of complex2 microRNA:mRNA1
    a(7) = c(7)*X(3);       %dissociation of complex1
    a(8) = c(8)*X(4);       %dissociation of complex 2 microRNA:mRNA2
    a(9) = c(9)*X(3);       %decay of complex1
    a(10) = c(10)*X(4);     %decay of complex2
    a(11) = c(11);          %microRNA production
    a(12) = c(12)*X(5);     %microRNA decay
    a(13) = c(13)*X(3);     %microRNA turnover from complex1 (and miRNA return to the pool)
    a(14) = c(14)*X(4);     %microRNA turnover from complex2 (and miRNA return to the pool)
    asum = sum(a);


  %%% Matrix 'bij' (partial derivatives) by hand.
  b=zeros(M,N_Species); %Only some are different to 0.
  b(2,1)= c(2);         %da2/dx1
  b(5,1)= c(2)*X(5);    %da5/dx1
  b(4,2)= c(4);         %da4/dx2
```

```
b(6,2)= c(6)*X(5);    %da6/dx2
b(7,3)= c(7);         %da7/dx3
b(9,3)= c(9);         %da9/dx3
b(13,3)= c(13);       %da13/dx3
b(8,4)= c(8);         %da8/dx4
b(10,4)= c(4);        %da10/dx4
b(14,4)= c(14);       %da14/dx4


%vector 'xi'.
for j=1:M
   xi(:,j) = a(j)*V(:,j);
end
xi=sum(xi,2);


vector = zeros(1,M);
for h=1:M
    vector(h)= epsilon*asum/abs((xi(1)*b(h,1)+xi(2)*b(h,2)+xi(3)*b(h,3)+xi(4)*b(h,4)));
end
index=find(vector==Inf);
vector(index)=[];


%Now we find 'tau'
tau = min(vector);

%CONDITION: To control that tau is not smaller than Gillespie's tau
if tau<=1/asum
   disp(['Is better apply Gillespie at this point. t= ',num2str(t)])
   % pause
end
%END CONDITION

k=zeros(M,1);
lambda=zeros(N_Species,1);
for j=1:M
    k(j)=poissrnd(a(j)*tau);
    lambda= lambda + k(j)*V(:,j);
end
```

```
  % CONTROL that no reactant population will be driven negative:
  if ((lambda(1)+X(1)) <0 || (lambda(2)+ X(2)) <0 || (lambda(3)+ X(3)) <0 || (lambda(4)+ X(4)) <0)
      tau=0;
      lambda=zeros(N_Species,1);
  end


  t = t + tau;
  Xold = X;
  X = X + lambda;


  count = count + 1;
  Xvals(:,count) = [t X']'; %Ara guardem el temps i les X junts, a la mateixa matrix

end

Xvals(:,count:end)= [];
return;
```

Appendix D

Langevin Code for two targets model

% langevin.m (27.06.08)

%

% Simple implementation of the code of D J Higham and R Khanin, 2007, to simulate the

% Chemical Langevin Equation for a simple two mRNA targets model regulated by miRNA.

% This version runs the model up to time tfinal and

% returns the values of all the reactants over time.

%

% Adapted from D J Higham and R Khanin (2007) code for

% the case of a simplified protein monomer-dimer system.

%

% Martina Marba, 2008.

```
function Xvals=langevin1(V, X, c, tfinal)

L= 5e2; %test a
```

```
%L= 5e3; %test b
tau=tfinal/L;  %stepsize

Xold = X
t = 0;

count = 1;
begin=(length(X)+1);
Xvals = zeros(begin,tfinal);
tvals(1)=0;
Xvals(:,1) = [t X']';



for k =1:L

    a(1) = c(1);            %production of mRNA1
    a(2) = c(2)*X(1);       %decay of mRNA1
    a(3) = c(3);            %production of mRNA2
    a(4) = c(4)*X(2);       %decay of mRNA2
    a(5) = c(5)*X(1)*X(5);  %formation of complex1 microRNA:mRNA1
    a(6)  = c(6)*X(2)*X(5); %formation of complex2 microRNA:mRNA1
    a(7) = c(7)*X(3);       %dissociation of complex1
    a(8) = c(8)*X(4);       %dissociation of complex 2 microRNA:mRNA2
    a(9) = c(9)*X(3);       %decay of complex1 and return of microRNA to the pool
    a(10) = c(10)*X(4);     %decay of complex2 and return of microRNA to the pool
    a(11) = c(11);          %microRNA production
    a(12) = c(12)*X(5);     %microRNA decay
    a(13) = c(13)*X(3);     %microRNA turnover from complex1
    a(14) = c(14)*X(4);     %microRNA turnover from complex2

    for m =1:length(a)
       d(m) = tau*a(m)+sqrt(abs(tau*a(m)))*randn;

    end

    X = X + d(1)*V(:,1) + d(2)*V(:,2) + d(3)*V(:,3) + d(4)*V(:,4) + d(5)*V(:,5) + d(6)*V(:,6)

    + d(7)*V(:,7) + d(8)*V(:,8) + d(9)*V(:,9) + d(10)*V(:,10)+ + d(11)*V(:,11) + d(12)*V(:,12)

    + d(13)*V(:,13) + d(14)*V(:,14);
    count = count + 1;
    t = t + tau;
```

```
    Xvals(:,count) = [t X']';
end

Xvals(:,count:end)= [];
return;
```

Bibliography

Ambros, V. (2004), 'The functions of animal microrna', *Nature* **431**, 350–355.

Blake, W. J., Kærn, M., Cantor, C. R. & Collins, J. J. (2003), 'Noise in eukaryotic gene expression', *Nature* **422 (6932)**, 633–637.

Bundschuh, R., Hayot, F. & Jayaprakash, C. (2003), 'Fluctuations and slow variables in genetic networks', *Biophysical Journal* **84**, 1606–1615.

Cao, Y., Gillespie, D. & Petzold, L. (2005), 'Avoiding negative populations in explicit poisson tau-leaping', *The Journal of Chemical Physics* **123**, 054104–054112.

Carthew, R. (2006), 'Gene regulation by microrna', *Curr Opin Genet Dev* **16**, 203–208.

Elowitz, M. B., Levine, A. J., Siggia, E. D. & Swain, P. S. (2002), 'Stochastic gene expression in a single cell', *Science* **297 (5584)**, 1183–1186.

Fraser, H., Hirsh, A., Giaever, G., Kumm, J. & Eisen, M. (2004), 'Noise minimization in eukaryotic gene expression', *PLoS Biol.* **2**(6), e137.

Gadgil, C., Lee, C.-H. & Othmer, H. G. (2005), 'A stochastic analysis of first-order reactions networks', *Bulletin of Mathematical Biology* **67**, 901–946.

Gillespie, D. (1976), 'General method for numerically simulating stochastic time evolution of coupled chemical-reactions', *Journal Comput. Phys.* **22**, 403–434.

Gillespie, D. (1992), 'A rigorous derivation of the chemical master equation', *Physica A* **188**, 404–425.

Gillespie, D. (2000), 'The chemical langevin equation', *The Journal of Chemical Physics* **113**(1), 297–306.

Gillespie, D. (2007), 'Stochastic simulation of chemical kinetics', *Annual Review of Physical Chemistry* **58**, 35–55.

Gillespie, D. T. (2001), 'Approximate accelerated stochastic simulation of chemically reacting systems', *The Journal of Chemical Physics* **115**(4), 1716–1733.

Higham, D. J. (2007), 'Modeling and simulating chemical reactions', *SIAM Rev.* **50**(2), 347–368.

Higham, D. J. & Khanin, R. (2008), 'Chemical master versus chemical langevin for first-order reaction networks', *The Open Applied Mathematics Journal* **2**, 59–79.

Hobert, O. (2008), 'Gene regulation by transcription factors and microrna', *Science* **319**(5871), 1785–1786.

Hornstein, E. & Shomron, N. (2006), 'Canalization of development by microrna', *Nature Genetics* **38 Suppl 1**, S20–S24.

Kærn, M., Elston, T. C., Blake, W. J. & Collins, J. J. (2005), 'Stochasticity in gene expression: from theories to phenotypes.', *Nat Rev Genet* **6**(6), 451–464.

Khanin, R. & Higham, D. J. (2008), 'Chemical master equation and langevin regimes for a gene transcription model', *Theoretical Computer Science*.

Khanin, R. & Higham, D. J. (2009), 'A multi-step model for microrna-mediated gene regulation', *Submitted*.

Khanin, R. & Vinciotti, V. (2008), 'Computational modeling of post-transcriptional gene regulation by microrna', *Journal of Computational Biology* **15**(3), 305–316.

Levine, E., Jacob, E. B. & Levine, H. (2007), 'Target-specific and global effectors in gene regulation by microrna', *Biophysical Journal* **93**(11), L52–L54.

Lewis, B., Burge, C. & Bartel, D. (2005), 'Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets', *Cell* **120**, 15–20.

Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D. & van Oudenaarden, A. (2002), 'Regulation of noise in the expression of a single gene', *Nature genetics* **31 (1)**, 69–73.

Rajewsky, N. (2006), 'Microrna target predictions in animals', *Nat Rev Genet* **38 Suppl 1**, S8–S13.

Raser, J. M. & O'Shea, E. K. (2004), 'Control of Stochasticity in Eukaryotic Gene Expression', *Science* **304**, 1811–1815.

Rodriguez, A., Griffiths-Jones, S., Ashurst, J. & Bradley, A. (2004), 'Identification of mammalian microrna host genes and transcription units', *Genome research* **14**(10), 1902–1910.

Shimoni, Y., Friedlander, G., Hetzroni, G., Niv, G., Altuvia, S., Biham, O. & Margalit, H. (2007), 'Regulation of gene expression by small non-coding rnas: a quantitative view', *Molecular Systems Biology* **3**, 138.

Standart, N. & Jackson, R. J. (2007), 'Micrornas repress translation of m7gpppg-capped target mrnas *in vitro* by inhibiting initiation and promoting deadenylation', *Genes Dev* **21**, 1975–1982.

Stark, A., Brennecke, J., Bushati, N., Russell, R. & Cohen, S. (2005), 'Animal micrornas confer robustness to gene expression and have a significant impact on 3'utr evolution', *Cell* **123**, 1133–1146.

Swain, P. S., Elowitz, M. B. & Siggia, E. D. (2002), 'Intrinsic and extrinsic contributions to stochasticity in gene expression', *Proc. Natl. Acad. Sci. USA* **99 (20)**, 12795–12800.

Takasu, F. (2005), 'Lecture 6: Immigration-emigration models 4', Lecture Notes, Dept. Information and Computer Sciences, Nara Women's University. takasu@ics.nara-wu.ac.jp.

Thattai, M. & van Oudenaarden, A. (2001), 'Intrinsic noise in gene regulatory networks', *Proc. Natl. Acad. Sci. USA* **98 (15)**, 8614–8619.

Tsang, J., Zhu, J. & van Oudenaarden, A. (2007), 'Microrna-mediated feedback and feedforward loops are recurrent network motifs in mammals', *Molecular Cell* **26**(5), 753–67.

Van Kampen, N. G. (2007), *Stochastic Processes in Physics and Chemistry, Third Edition (North-Holland Personal Library)*, North Holland.